



UNIVERSIDAD DE LAS CIENCIAS INFORMÁTICAS

FACULTAD 2 “TELECOMUNICACIONES Y SEGURIDAD INFORMÁTICA”

**TRABAJO DE DIPLOMA PARA OPTAR POR EL TÍTULO DE
INGENIERO EN CIENCIAS INFORMÁTICAS**



**Título: Propuesta de integración de los subsistemas del Sistema Informático de
Gestión de Auditoría y Control (SIGAC).**

Autores:

Anay Díaz Estrada.

Andrés Bernardo González Morales.

Tutor:

Ing. Yasser Azan Basallo.

Co-Tutora:

Ing. Yaily Rodríguez Rodríguez.

Ciudad de La Habana, Mayo del 2009.

“Año del 51 Aniversario del Triunfo de la Revolución Cubana”

Declaración de Autoría

Declaramos que Anay Díaz Estrada y Andrés B. González Morales somos los únicos autores de este trabajo y autorizamos a la Universidad de las Ciencias Informáticas (UCI) y a la Facultad (2) para que hagan el uso que estimen pertinente con este trabajo.

Para que así conste firmamos la presente a los ____ días del mes de _____ del 2008.

Anay Díaz Estrada

Andrés B. González Morales

Ing. Yasser Azan Basallo.

Ing. Yaily Rodríguez Rodríguez



“Hombre es algo más que ser torpemente vivo: es entender una misión, ennoblecerla y cumplirla.”

José Martí

AGRADECIMIENTOS

DEDICATORIA

RESUMEN

La integración de los subsistemas que componen la solución de software del Sistema Informático para la Gestión de Información del Ministerio de Auditoría y Control (en lo adelante SIGAC) tiene gran importancia para que este funcione como un todo, debido a que sus módulos de manera independiente no podrán satisfacer todas las necesidades para las cuales fueron creados, debido a la dependencia que existe entre sí.

Es por esto que surge la necesidad, como parte del SIGAC para el Ministerio de Auditoría y Control (MAC), de una propuesta que tiene como objetivo concreto proveer a la solución de software implementada la integración de sus distintos subsistemas garantizando la interoperabilidad entre los mismos. Para esto se define una arquitectura de Integración de Aplicaciones Empresariales o EAI que además servirá de guía para la integración de futuros módulos añadidos al SIGAC.

En el presente extenso se plasman los resultados del estudio realizado para el desarrollo de dicha solución, se explican los conceptos relacionados con el mismo, se hace un análisis de la propuesta del sistema, finalmente se dejan algunas recomendaciones para el mejoramiento futuro de la propuesta a desarrollar.

PALABRAS CLAVES

Integración, EAI, arquitectura.

ÍNDICE DE CONTENIDOS

AGRADECIMIENTOS	II
DEDICATORIA	III
RESUMEN	IV
INTRODUCCIÓN.....	1
CAPÍTULO #1: FUNDAMENTACIÓN TEÓRICA	4
1.1 INTRODUCCIÓN.....	4
1.2 TÉCNICAS DE INTEGRACIÓN DE DATOS	4
1.2.1 Replicación de Datos.....	5
1.2.2 Extracción, Transformación y Carga de Datos (ETL).....	5
1.2.3 Integración de Información Empresarial (EII)	5
1.2.4 Integración de Aplicaciones Empresariales (EAI)	6
1.3 ARQUITECTURA EAI.....	7
1.4 PROTOCOLO DE COMUNICACIÓN	9
1.4.1 HTTP.....	10
1.5 SERVICIOS WEB	11
1.6 METODOLOGÍA DE DESARROLLO.....	15
1.6.1 TBI (Total Business Integration).....	15
1.7 HERRAMIENTAS Y LENGUAJES	16
1.7.1 Lenguaje de Modelado.....	16
1.7.2 Herramienta CASE.....	17
1.7.3 Lenguajes de Programación.....	18
1.8 CONCLUSIONES.....	19
CAPÍTULO #2: DEFINICION DEL PROBLEMA	20
2.1 INTRODUCCIÓN.....	20
2.2 INTEGRACIÓN DE SISTEMAS.....	20
2.4 SUBSISTEMAS QUE SE RELACIONAN.....	21
2.4.1 Módulo Planificación.....	21
2.4.2 Módulo Acciones de Control	22
2.5 DEFINICIÓN DE PROCESOS DE NEGOCIO	24
2.5.1 Proceso: Crear Orden de Trabajo.....	27
2.5.2 Proceso: Obtener Control de Cumplimiento	29
2.6 PROCESO DE SOPORTE.....	31
2.6.1 Guardar Configuración de Alfresco.....	31
2.6 RELACIÓN DE REQUERIMIENTOS.....	32
2.6.1 Requisitos Funcionales.....	33
2.6.2 Requisitos no Funcionales	35
2.7 CONCLUSIONES.....	37
CAPÍTULO #3: SOLUCIÓN PROPUESTA	38
3.1 INTRODUCCIÓN.....	38
3.2 ARQUITECTURA EAI EN EL SISTEMA SIGA C	38
3.3 ORGANIZACIÓN GENERAL	39

3.4 DESCRIPCIÓN DE LAS INTERFACES	40
3.4.1 Módulo Integración.....	42
3.5 PATRONES EN LA PROPUESTA.....	45
3.5.1 Patrones de Integración	45
3.5.2 Patrones Arquitectónicos.....	45
3.5.3 Patrones Generales de Asignación de Responsabilidades.....	47
3.5.4 Patrones de Diseño.....	48
3.6 VISTA FÍSICA DE INTEGRACIÓN.....	49
3.7 DESPLIEGUE DEL SUBSISTEMA.....	50
3.8 CONCLUSIONES.....	52
CAPÍTULO #4: EVALUACION DE LA PROPUESTA.....	53
4.1 INTRODUCCIÓN.....	53
4.2 EVALUACIÓN.....	53
4.3 MÉTODO DELPHI.....	54
4.3.1 Proceso de Selección de expertos.....	54
4.3.2 Objetivo a evaluar por los especialistas:	55
4.3.3 Guía para la validación de la propuesta.....	55
4.4 CONCLUSIONES.....	61
CONCLUSIONES	62
RECOMENDACIONES	63
BIBLIOGRAFÍA	64
GLOSARIO DE TÉRMINOS	68
ANEXOS	72

INDICE DE FIGURAS

FIGURA 1. ARQUITECTURA ACCIDENTAL.....	7
FIGURA 2. ARQUITECTURA HUB-AND-SPOKE ADOPTADA POR LOS SISTEMAS EAI	8
FIGURA 3. NIVELES DEL MODELO OSI	9
FIGURA 4. PROTOCOLO HTTP.....	10
FIGURA 5. GRUPO DE PROTOCOLOS BASES DE LOS SERVICIOS WEB.....	12
FIGURA 6. FASES Y SUB-FASES DE LA METODOLOGÍA TOTAL BUSINESS INTEGRATION.....	16
FIGURA 7. LAS 4 GRANDES ÁREAS DEL GESTOR DE CONTENIDOS EMPRESARIAL ALFRESCO.....	24
FIGURA 8. INTERFAZ NO FUNCIONAL “ADICIONAR ACCIÓN DE CONTROL” DEL MÓDULO PLANIFICACIÓN.....	25
FIGURA 9. PROCESO CREAR ORDEN DE TRABAJO.....	28
FIGURA 10. PROCESO: OBTENER CONTROL DE CUMPLIMIENTO.....	30
FIGURA 11. PROCESO DE SOPORTE GUARDAR CONFIGURACIÓN DE ALFRESCO.....	32
FIGURA 12. ARQUITECTURA EAI EN EL SISTEMA SIGAC.....	39
FIGURA 13. SUBSISTEMAS DE LA SOLUCIÓN DE INTEGRACIÓN DEL SISTEMA SIGAC.....	40
FIGURA 14. SERVICIOS WEB QUE BRINDA ALFRESCO.....	41

FIGURA 15. VISTA LÓGICA DEL PAQUETE INTEGRACIÓN.....	42
FIGURA 16. PROTOTIPO NO FUNCIONAL DE INTERFAZ CONFIGURAR ALFRESCO.....	44
FIGURA 17. ESTRUCTURA DE MVC.....	46
FIGURA 18. SUBSISTEMAS INVOLUCRADOS.....	49
FIGURA 19. COMPONENTES DEL SUBSISTEMA INTEGRACIÓN.....	50
FIGURA 20. DESPLIEGUE DEL SUBSISTEMA.....	51
FIGURA 21. REPRESENTACIÓN DEL VALOR DE LOS CRITERIOS EN PORCIENTO.....	57

INDICE DE TABLAS

TABLA 1. RESPUESTAS DEL MÓDULO INTEGRACIÓN.....	27
TABLA 2. MODIFICACIÓN DE DATOS EN EL PROCESO CREAR ORDEN DE TRABAJO.....	29
TABLA 3. MODIFICACIÓN DE DATOS EN EL PROCESO OBTENER CONTROL DE CUMPLIMIENTO.....	31
TABLA 4. EVALUACIÓN DE LOS EXPERTOS.....	57
TABLA 5. VALORES OBTENIDOS.....	59
TABLA 6. CALIFICACIÓN PROMEDIO PARA CADA CRITERIO.....	60

INTRODUCCIÓN

La incorporación de las Tecnologías de la Información y la Comunicación (TIC) en el ámbito empresarial puede resultar un elemento clave para mejorar la competitividad, impulsar el crecimiento económico y lograr una mayor agilidad en los procesos. Por tanto, la competitividad de Cuba y su papel en el orden mundial dependen, en buena medida, de la presencia de la UCI en la sociedad de la información.

Cuba no está ajena a la necesidad de ingresar definitivamente en el mundo del software y para ello se ha trazado la estrategia de aplicar paulatinamente las tecnologías a las distintas esferas. Como parte de la incorporación a este avance surgen las relaciones de trabajo y colaboración entre el Ministerio de Auditoría y Control (MAC) y la Universidad de las Ciencias Informáticas (UCI), institución que nació con el compromiso de informatizar la sociedad cubana.

El MAC se crea, como “un Organismo de la Administración Central del Estado encargado de dirigir, ejecutar y controlar la aplicación de la política del Estado y del Gobierno en materia de Auditoría Gubernamental, Fiscalización y control Gubernamental; así como para regular, organizar, dirigir y controlar, metodológicamente, el Sistema Nacional de Auditoría.” (1), el 25 de abril del 2001, por el Decreto Ley No. 219 del Consejo de Estado.

Actualmente, para efectuar el trabajo en el Ministerio de Auditoría y Control solo existen algunas herramientas de ayuda, las cuales no resultan de mucha utilidad, ya que los procesos para llevar un control de la información se realizan con lentitud y resultan complejos, afectando el correcto funcionamiento de la institución. Con el objetivo de solucionar dichos problemas se crea en la universidad el proyecto Informatización del MAC (IMAC), compuesto por dos módulos: Planificación y Acciones de Control. Este proyecto, en su interés de informatizar al ministerio que se viene mencionando, se propone como parte de la solución del proyecto ERP-CUBA¹, con el cual se integrará, por lo que debe regirse por las mismas normas tecnológicas que el mismo (lenguaje de programación, marco de trabajo, etc.).

¹ Proyecto que se propone el desarrollo de un sistema ERP, o Sistema Integral de Gestión para la Empresa, el cual será implantado en todo el país.

El módulo Acciones de Control se desarrolla sobre el gestor de contenido empresarial Alfresco, la integración del mismo con Planificación o posteriores módulos que se añadan se hace difícil, esto ocurre debido a la heterogeneidad de lenguajes y herramientas en que se implementan. Por esta razón resulta necesario el desarrollo de una solución que logre la vinculación entre los módulos de SIGAC con Alfresco, permitiendo la centralización de algunos procedimientos que garantizarían el buen funcionamiento del mismo.

Por lo antes expuesto se ha identificado como **Problema Científico** a resolver:

¿Cómo lograr el intercambio de información entre los subsistemas Acciones de Control y Planificación del sistema SIGAC?

Como **Objeto de Estudio** para este trabajo se tiene: técnica para la integración de aplicaciones y como **Campo de Acción** proceso de integración de los módulos Planificación y Acciones de Control que componen el sistema informático SIGAC.

La presente investigación tiene como **Objetivo General**: Definir una propuesta de solución que permita integrar los subsistemas: Planificación y Acciones de Control, de forma que se garantice la interoperabilidad entre los mismos en el sistema SIGAC para el MAC.

Objetivos específicos:

- ❖ Fundamentar teóricamente la propuesta de integración de los subsistemas Planificación y Acciones de Control del sistema SIGAC.
- ❖ Presentar la propuesta de solución mediante el diseño de una de arquitectura de integración.
- ❖ Validar la propuesta presentada.

Para cumplir con lo antes descrito en esta investigación se han trazado las siguientes **Tareas de la Investigación**:

- ❖ Realizar un estudio de las técnicas de integración existentes.
- ❖ Realizar un estudio sobre los aspectos relacionados con los diferentes subsistemas que componen la solución de software.
- ❖ Presentar el lenguaje y herramientas a utilizar para el diseño e implementación de la solución.
- ❖ Proponer la solución mediante la arquitectura a seguir para la integración de los subsistemas Planificación y Acciones de Control.

- ❖ Evaluar la solución de integración propuesta.

El presente extenso, está estructurado en 4 capítulos, a continuación se muestra una breve descripción de cada uno de ellos:

Capítulo 1 “Fundamentación teórica”, en este capítulo se realiza un estudio sobre el tema integración de aplicaciones y se proponen un conjunto de herramientas y lenguaje a utilizar en el desarrollo de la integración de SIGAC.

Capítulo 2 “Definición del problema”, es donde se plantea la necesidad del módulo Integración como subsistema, se describen los procesos a ejecutarse mediante la integración y se exponen los requisitos, tanto funcionales como no funcionales del subsistema, está enfocado a definir todo el proceso de negocio.

Capítulo 3 “Solución Propuesta”, describe cómo desarrollar los procesos de Integración, definiéndose la arquitectura del sistema, está enfocado a definir la solución que se propone.

Capítulo 4 “Evaluación de la Propuesta”, es donde se evalúa la solución propuesta en el capítulo anterior.

CAPÍTULO #1: FUNDAMENTACIÓN TEÓRICA

1.1 Introducción

En este capítulo se realiza un breve estudio sobre las principales técnicas de integración, enfatizando en la técnica acorde al caso de estudio. Se mencionan un conjunto de herramientas y lenguajes a utilizar en el desarrollo de la integración del Sistema de Gestión para el Ministerio de Auditoría y Control (SIGAC).

1.2 Técnicas de Integración de datos

El esquema tradicional cliente/servidor permite que diferentes módulos de aplicación se comuniquen directamente sin una capa intermediaria. El problema se presenta en sistemas complejos con componentes de diversos proveedores donde resulta poco flexible e inoperante la comunicación. (2)

La Visión de Integración es la entrega de información:

- ❖ Precisa.
- ❖ Consistente.
- ❖ Oportuna.
- ❖ Coherente

La integración se puede enfocar de varias formas diferentes dependiendo de la idea de “Integración” que se tenga. Fundamentalmente existen cuatro:

- ❖ Replicación de Datos.
- ❖ Extracción, Transformación y Carga de Datos o ETL².
- ❖ Integración de Información Empresarial o EII³.
- ❖ Integración de Aplicaciones Empresariales o EAI⁴.

² Del inglés Extract, Transform and Load.

³ Del inglés Enterprise Information Integration.

⁴ Del inglés Enterprise Application Integration.

1.2.1 Replicación de Datos

La Replicación de Bases de Datos es una técnica de integración que se basa en la creación y mantenimiento de múltiples “copias” de una misma base de datos. En la mayoría de las implementaciones de Replicación, un servidor mantiene la copia primaria de la base de datos y servidores adicionales mantienen las copias esclavas de la misma. (3)

- ❖ Lo escrito en la Base de Datos se envía al servidor primario y luego es replicado por las copias esclavas.
- ❖ Las lecturas de las Bases de Datos están divididas entre todos los servidores, lo cual trae una gran ventaja debido al uso compartido de carga.

La transferencia se realiza de Base de Datos a Base de Datos (este es el estilo más antiguo de integración), lo cual es una desventaja para la resolución del problema a resolver dado la heterogeneidad de las fuentes.

1.2.2 Extracción, Transformación y Carga de Datos (ETL)

La técnica Extracción Transformación y Carga de Datos (Extract, Transform and Load), como su nombre lo indica extrae información de un sistema fuente, transforma esos datos para satisfacer los requerimientos del negocio y carga el resultado en el sistema destino. Tanto la fuente como el destino son generalmente Bases de Datos y archivos.

La transformación puede implicar la reestructuración y reconciliación del registro de datos, limpieza del contenido de datos (es decir, revisados por si existen discrepancias y eliminación de datos obviamente falsos), y/o agregación del contenido de datos. Esto sucede mediante una serie de procedimientos especiales que permiten obtener un formato unificado común y mejorado. Sólo después de la revisión y unificación de los datos estos son cargados. Esta técnica se encarga de la integración de datos, no de aplicaciones, y obtiene los datos directamente de las Bases de Datos. De acuerdo a las características de esta técnica se concluyó que no es la más adecuada para la solución que se necesita. (4)

1.2.3 Integración de Información Empresarial (EII)

La Integración de Información Empresarial o EII es otra de las técnicas de integración. Es un mecanismo de transformación y acceso a datos transparente y optimizado para suministrar una única interfaz a lo

largo de los datos de las organizaciones. Dicha interfaz permite acceder a los datos y el resultado de este método es un Sistema de Información Heterogéneo Distribuido, Virtualmente Integrado. Este tipo de solución consiste en crear un intermediario que contenga los directorios de la Base de Datos y que a su vez sirva de canal de consulta y representación de la información recuperada. La información es capturada en tiempo real lo que implica que las fuentes de datos tengan una estructura tecnológica sólida y bien establecida. (5)

EII protege a las aplicaciones de la complejidad de recuperar datos de múltiples localizaciones, donde los datos pueden diferir en semántica y formato, y emplear diferentes interfaces de datos.

Teniendo en cuenta estos aspectos, para la integración de Datos a Tiempo Real la técnica EII constituye una buena alternativa, sin embargo no es factible para la integración de aplicaciones.

1.2.4 Integración de Aplicaciones Empresariales (EAI)

EAI⁵ es el proceso de integrar múltiples aplicaciones desarrolladas independientemente, que utilizan tecnología incompatible y que son gestionadas de forma independiente, permitiendo que se comuniquen e intercambien transacciones de negocio, mensajes, y datos entre sí. Uno de los principales objetivos de EAI es proporcionar acceso transparente a la amplia gama de aplicaciones que existen en una organización. Las características más importantes de esta tecnología es que se utiliza para la integración de Aplicaciones – a – Aplicaciones y *“proporciona un enfoque de integración orientado a proceso basado en mensajes XML.”* (6) Es generalmente utilizada para el procesamiento de transacciones de negocio operacional en tiempo real.

Beneficios:

- ❖ Integración no invasiva.
- ❖ El contenido de los datos (el mensaje) es irrelevante.
- ❖ Puede generalmente acceder a muchos puntos finales.
- ❖ El contenido del mensaje podría ser XML, Servicios Web, datos planos, etc.
- ❖ Los puntos finales pueden ser colas, protocolos, Servicios Web, Bases de Datos, adaptadores de aplicaciones, entre otros.

⁵ Del inglés Enterprise Application Integration.

El sistema SIGAC necesita un intercambio de información en tiempo real entre los subsistemas (Planificación y Acciones de Control), por lo que se plantea EAI como solución propuesta para lograr esta integración.

1.3 Arquitectura EAI

La solución más común que se adoptó inicialmente para abordar la integración de la funcionalidad de distintos sistemas fue el establecimiento de conexiones punto a punto entre los sistemas que tenían que ser integrados, realizando modificaciones en los mismos para posibilitar su comunicación. A esto se le llama arquitectura accidental, caracterizada por un conjunto de integraciones “artesanales” que no responde a un plan bien definido, y que introduce sucesivos cambios en los sistemas integrados, haciendo peligrar su integridad y dificultando su mantenimiento. Además, este tipo de integración hace muy difícil la tarea de conocer las dependencias entre sistemas, monitorizar la comunicación entre los mismos y, sobre todo, de abordar nuevas necesidades de integración ágilmente y con garantías. (7)

La Figura1 muestra una arquitectura accidental típica, donde se representa la comunicación entre diferentes sistemas:

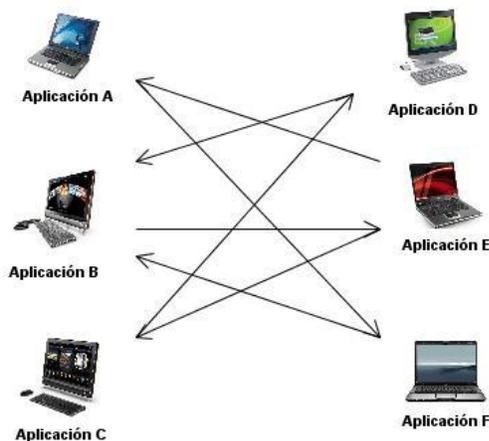


Figura 1. Arquitectura Accidental.

A mediados de la década de los 90 comenzó la consolidación de una categoría de software orientado a la integración de aplicaciones: los comúnmente denominados EAI brokers o brokers de integración, que intentan disminuir los problemas de la arquitectura accidental. Este tipo de software nace con el propósito

de integrar sistemas aislados de una forma controlada y de facilitar los proyectos de conexión de sistemas heterogéneos.

Mientras que las estrategias iniciales de integración se basaban en la integración punto a punto de los sistemas en cuestión, los brokers EAI suelen usar una arquitectura hub-and-spoke en la que las comunicaciones y transferencias de datos pasan a través de un hub central, donde un servidor de integración maneja las comunicaciones y realiza las transformaciones de datos. Este tipo de arquitectura es flexible, promueve la reusabilidad de interfaces (los “spokes”) reduciendo el número de interfaces necesarias (significativamente si se compara con una arquitectura de punto a punto), es simple de diseñar y mantener. (8)

La Figura 2 muestra la arquitectura hub-and-spoke adoptada por los brokers EAI:

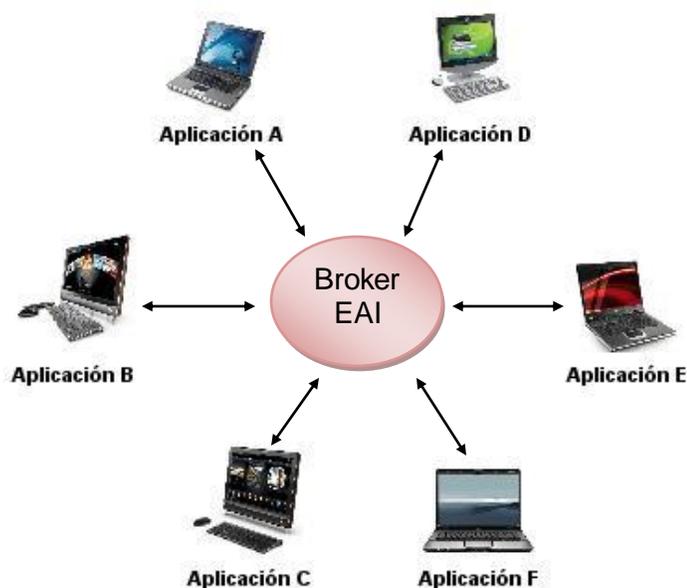


Figura 2. Arquitectura Hub-and-Spoke adoptada por los sistemas EAI.

El software categorizado como broker para la integración de aplicaciones suele proporcionar un conjunto de características que incluye: conectividad con distintas aplicaciones y tecnologías a través de adaptadores, motores de transformación de mensajes para que estos se ajusten al formato que espera el sistema destino, e integración de sistemas en tiempo real.

1.4 Protocolo de comunicación

Se entiende por protocolo a un método establecido para intercambiar datos en Internet, es además, un método por el cual dos ordenadores acuerdan comunicarse, una especificación que describe cómo los ordenadores se comunican el uno al otro en una red, son reglas de comunicación que permiten el flujo de información entre computadoras distintas que manejan lenguajes distintos.

Los protocolos se pueden dividir en varias categorías, una de las clasificaciones más estudiadas es la OSI⁶, según dicha clasificación, la comunicación de varios dispositivos se puede estudiar dividiéndola en 7 niveles, que son expuestos desde su nivel más alto hasta el más bajo. Estos 7 niveles se pueden subdividir en dos categorías, las capas superiores y las capas inferiores. Las 4 capas superiores trabajan con problemas particulares a las aplicaciones, y las 3 capas inferiores se encargan de los problemas pertinentes al transporte de los datos (Figura 3).

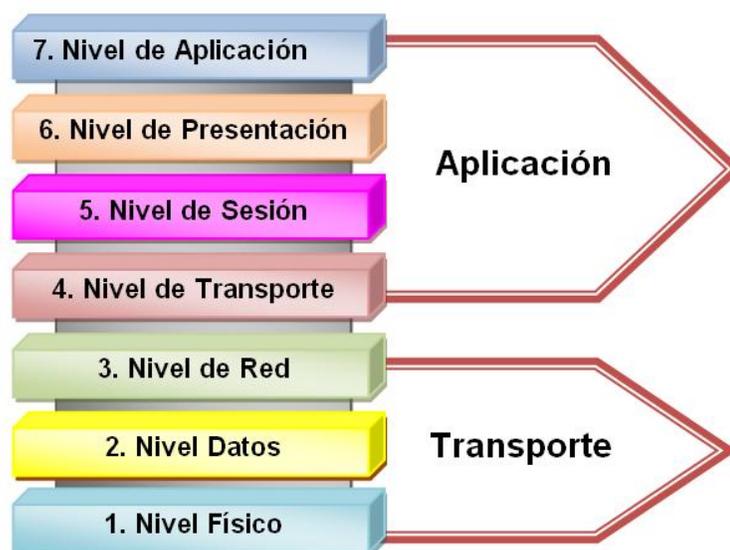


Figura 3. Niveles del Modelo OSI.

⁶ El modelo de referencia de Interconexión de Sistemas Abiertos (OSI, Open System Interconnection) modelo de red descriptivo creado por ISO (Organización Internacional para la Estandarización).

Los protocolos de cada capa tienen una interfaz bien definida. Una capa generalmente se comunica con la capa inmediata inferior, la inmediata superior, y la capa del mismo nivel en otros computadores de la red. Esta división de los protocolos ofrece abstracción en la comunicación.

Existen y pueden existir tantos protocolos como aplicaciones existan, es importante aclarar que el usuario normalmente no interactúa directamente con el nivel de aplicación sino con programas que son los que a su vez interactúan con el nivel de aplicación pero ocultando su complejidad. El usuario solamente conoce que el ordenador o dispositivo debe soportar los protocolos adecuados si se quiere obtener comunicación con otros ordenadores, pero la manera en que lo realiza resulta transparente para el mismo.

1.4.1 HTTP

Protocolo de Transferencia de Hipertexto, del inglés HyperText Transfer Protocol, es usado en cada transacción de la Web (WWW), define la sintaxis y la semántica que utilizan los elementos software de la arquitectura web (clientes, servidores, proxies) para comunicarse. Es un protocolo orientado a transacciones y sigue el esquema petición-respuesta entre un cliente y un servidor (Figura 4).

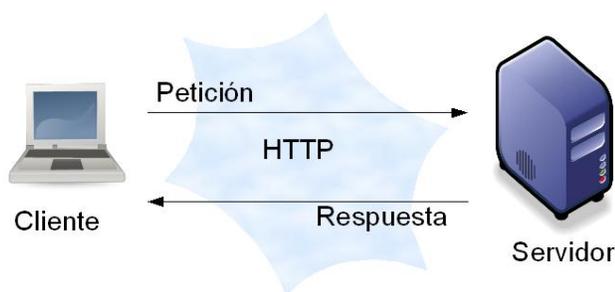


Figura 4. Protocolo HTTP.

La información transmitida mediante este protocolo se le llama recurso y se le identifica mediante un url. Los recursos pueden ser: archivos, el resultado de la ejecución de un programa, una consulta a una base de datos o la traducción automática de un documento, entre otros.

Entre las características que brinda está la utilización de servicios web que puede formar parte de una solución de intercambio de información a la hora de realizar integración entre aplicaciones.

1.5 Servicios Web

En la medida que surgen y perfeccionan las herramientas de desarrollo, los clientes se enfrentan con la difícil tarea de integrar una amplia variedad de aplicaciones distribuidas con diferentes entornos de ejecución y desarrollo. Debido a la importancia que representa en la actualidad este aspecto se han desarrollado varios “enfoques y/o tecnologías” para facilitar la integración, gracias a los estándares abiertos y al énfasis en la comunicación y colaboración entre personas y aplicaciones, y a sus múltiples usos, los Servicios Web son muy utilizados. (9) Estos solucionan los siguientes problemas:

- ❖ independencia de la Plataforma Hardware.
- ❖ independencia del Sistema Operativo.
- ❖ independencia del Lenguaje de Programación.

Los Servicio Web facilitan el intercambio de datos entre aplicaciones o componentes de aplicaciones de forma estándar a través de protocolos comunes (como http) y de manera independiente al lenguaje de programación, plataforma de implantación, formato de presentación o sistema operativo.

Un Servicio Web es un contenedor que encapsula funciones específicas y permite que estas puedan ser utilizadas en otros servidores. Los mismos se actualizan de forma transparente para el programador y los usuarios, mediante ellos se pueden desarrollar funciones complejas. Por otra parte la carga de uso del CPU que supone la ejecución de un algoritmo determinado que en algunos casos puede resultar de gran complejidad desaparece, pues la carga se ejecuta sobre el servidor del servicio web. (10)

En el intercambio de servicios existen dos elementos de colaboración indispensables:

- ❖ **Consumidor de servicio:** Es una aplicación, un módulo de software u otro servicio que requiere un servicio. Es la entidad de software que realiza una petición al proveedor de servicios. Tradicionalmente llamado “cliente”. Inicia la búsqueda en el registro, enlaza con el servicio a través del transporte y ejecuta la función del servicio de acuerdo con las reglas establecidas.
- ❖ **Proveedor de servicios:** La entidad de software que implementa la especificación del servicio. El proveedor de servicios es una entidad a la que se puede acceder a través de la red y que acepta y ejecuta peticiones de los consumidores. Publica las interfaces de los servicios en el registro de servicios para que los consumidores puedan descubrirlos y puedan acceder a ellos.

Algunas ventajas que presentan los Servicios Web son:

- ❖ Son programables.
- ❖ Están basados en XML⁷, que es un lenguaje abierto.
- ❖ Son auto-descriptivos.
- ❖ Pueden buscar registros de otros Servicios Web.

La interoperabilidad se consigue mediante la adopción de estándares abiertos. Las organizaciones OASIS⁸ y W3C⁹ son los comités responsables de la arquitectura y reglamentación de los servicios Web. Para mejorar la interoperabilidad entre distintas implementaciones de servicios Web se ha creado el organismo WS-I¹⁰, encargado de desarrollar diversos perfiles para definir de manera más exhaustiva estos estándares. (10) A los Servicios Web hay una serie de protocolos asociados:



Figura 5. Grupo de Protocolos Bases de los Servicios Web.

- ❖ **XML (Extensible Markup Language):** XML es un metalenguaje extensible de etiquetas que permite definir la gramática de lenguajes específicos, se propone como un estándar para el intercambio de información estructurada entre diferentes plataformas. Es un lenguaje con una importante función en el proceso de intercambio, estructuración y envío de datos en la Web.

⁷ Del inglés Extensible Markup Language.

⁸ Del Inglés Organization for the Advancement of Structured Information Standards.

⁹ Del Inglés World Wide Web Consortium.

¹⁰ Del Inglés Web Services Interoperability Organization.

Describe los datos de tal manera que es posible estructurarlos utilizando para ello etiquetas. Las características especiales son la independencia de datos, o de la separación de los contenidos de su presentación. Es una tecnología sencilla que tiene a su alrededor otras que la complementan y la hacen mucho más grande y con unas posibilidades mucho mayores. Tiene un papel muy importante en la actualidad ya que permite la compatibilidad entre sistemas para compartir la información de una manera segura, fiable y fácil. (11)

Entre las principales ventajas del uso de XML se tienen:

- Es extensible. Después de diseñado y puesto en producción, es posible extender XML con la adición de nuevas etiquetas, de modo que se pueda continuar utilizando sin complicación alguna.
- El analizador es un componente estándar. No es necesario crear un analizador específico para cada versión de lenguaje XML. Esto posibilita el empleo de cualquiera de los analizadores disponibles. De esta manera se evitan errores y se acelera el desarrollo de aplicaciones.
- Mejora la compatibilidad entre aplicaciones. Si un tercero decide usar un documento creado en XML, es sencillo entender su estructura y procesarla.

Como se muestra, la tecnología XML posee muchos aspectos que benefician su incorporación en una solución, sobre todo si se trata de la comunicación entre aplicaciones heterogéneas.

- ❖ **SOAP (Simple Object Access Protocol):** Protocolo Simple de Acceso a Objetos se trata de un protocolo basado en XML, que permite la interacción entre varios dispositivos y que tiene la capacidad de transmitir información compleja, ofreciendo funciones muy útiles a usuarios del medio Web. Los datos pueden ser transmitidos a través de diferentes protocolos como HTTP, SMTP, etc. SOAP especifica el formato de los mensajes. (12)

Algunas de las Ventajas de SOAP son:

- No está asociado con ningún lenguaje: No especifica una API¹¹, por lo que la implementación de esta se deja al lenguaje de programación y la plataforma.

¹¹ Interfaz de programación de aplicaciones es el conjunto de funciones y procedimientos (o métodos, si se refiere a programación orientada a objetos) que ofrece cierta biblioteca para ser utilizado por otro software como una capa de abstracción.

- No se encuentra fuertemente asociado a ningún protocolo de transporte: Su especificación no describe cómo se deberían asociar los mensajes de SOAP con HTTP. Dichos mensajes no son más que un documento XML, por lo que puede transportarse utilizando cualquier protocolo capaz de transmitir texto.
- Aprovecha los estándares existentes en la industria: Utiliza los estándares existentes para satisfacer sus necesidades.
- Permite la interoperabilidad entre múltiples entornos: Debido a que aprovecha los estándares existentes en la industria, las aplicaciones que se ejecuten en plataformas con dichos estándares pueden comunicarse mediante mensaje SOAP con aplicaciones que se ejecuten en otras plataformas.

❖ **WSDL (Web Services Description Language):** Se puede definir un archivo WSDL como un documento XML que describe un conjunto de mensajes SOAP y la forma en la que éstos se intercambian. WSDL es XML (es decir, se puede leer y modificar en gran parte de los casos) y es generado y utilizado por software. (9)

Especifica qué debe contener un mensaje de solicitud y cómo se verá el mensaje de respuesta en una notación clara, notación que se basa en el estándar XML, además define el lugar en el que está disponible el servicio y qué protocolo de comunicaciones se utiliza para hablar al servicio.

❖ **UDDI (Universal Description, Discovery and Integration):** Es donde se registran los Servicios Web para que los futuros usuarios los encuentren fácilmente, hace posible que empresas puedan tanto publicar como encontrar Servicios Web. Una vez descritos los negocios y los tipos de servicios que proporcionan se pueden registrar y publicar en un Registro UDDI. Tales negocios publicados pueden ser buscados, consultados o "descubiertos" por otros negocios utilizando mensajes con SOAP. (11)

Permite fácilmente buscar una empresa que ofrece los servicios que necesita, leer acerca del servicio ofrecido y ponerse en contacto con una persona para solicitar más información.

1.6 Metodología de Desarrollo.

La metodología EAI es diferente a los tradicionales Ciclos de Vida de Desarrollo de Software (SDLC), sin embargo, prevalece el enfoque del uso de técnicas que funcionan bien en la solución de problemas simples y entonces pueden ser aplicadas a problemas más complejos.

1.6.1 TBI (Total Business Integration)

Integración Total del Negocio o TBI¹² es propuesta actualmente como la base para las mejores prácticas en la integración por el Consorcio de la Industria EAI (EAI Industry Consortium o EAIC). El desarrollo de la metodología TBI es conducido por EAIC Methodology Committee (Comité de Metodología del Consorcio de la Industria EAI), y por EAIC Best Practices Committee (Comité de Mejores Prácticas del Consorcio de la Industria EAI), basándose en estándares de la industria y las mejores prácticas, como UML, y las mejores prácticas de Integradores de Sistemas. (13)

Esta metodología fue diseñada para los proyectos de integración Application-to-Application (A2A) y Business-to-Business (B2B). Es una metodología orientada a procesos del negocio, diseñada para proyectos de integración de datos, aplicaciones y procesos que utilizan sistemas heterogéneos para la gestión del negocio de la empresa. Suministra las bases para maximizar las oportunidades de reutilización, suministrando un enfoque de diseño común, plantillas reutilizables y procesos que pueden ser utilizados por otros proyectos dentro de la empresa. Cubre el ciclo completo y es independiente de las aplicaciones origen y las tecnologías, describe 4 fases (Figura 5), cada una con un grupo de subfases. (13)

¹² Del inglés Total Business Integration.



Figura 6. Fases y Sub-fases de la metodología Total Business Integration.

Beneficios de la Metodología TBI

- ❖ Crea un enfoque estandarizado para la integración.
- ❖ Promueve el re-uso y la influencia de los servicios para la integración, que reducen el costo total e incrementan la velocidad de desarrollo.
- ❖ Permite que se requieran cambios mínimos cuando el proceso de la empresa o la arquitectura de la aplicación subyacente cambien.

Debido al déficit de documentación con respecto a esta metodología se decide utilizar como modelo teórico con el cual se hace converger la solución presentada, aplicando los procesos que esta define para obtener a la postre un resultado integrado a nivel de empresa.

1.7 Herramientas y Lenguajes

1.7.1 Lenguaje de Modelado

Lenguaje Unificado de Modelado (UML, por sus siglas en inglés, Unified Modeling Language) Es un lenguaje gráfico para visualizar, especificar, construir y documentar un sistema de software, muy conocido y utilizado en la actualidad puesto que sirve para el modelado completo de sistemas complejos, tanto en el

diseño de los sistemas software como para la arquitectura hardware donde se ejecuten. Se puede aplicar en una gran variedad de formas para dar soporte a una metodología de desarrollo de software, pero no especifica en sí mismo qué metodología o proceso usar. (14)

Otro objetivo de este modelado visual es que sea independiente del lenguaje de implementación, de tal forma que los diseños realizados usando UML se puedan implementar en cualquier lenguaje que soporte las posibilidades de UML (principalmente lenguajes orientados a objetos).

UML aporta, además, las siguientes ventajas: (14)

- ❖ Permite un modelado con gran rigor en la especificación.
- ❖ Posibilita realizar una verificación y validación del modelo realizado.
- ❖ Se pueden automatizar determinados procesos y permite generar código a partir de los modelos y a la inversa (a partir del código fuente generar los modelos). Esto permite que el modelo y el código estén actualizados, con lo que siempre se puede mantener la visión en el diseño, de más alto nivel, de la estructura de un proyecto.

Este es el lenguaje de modelado que propone la metodología Total Business Integration.

1.7.2 Herramienta CASE

La realización de un nuevo software requiere que los procesos sean organizados y completados en forma correcta y eficiente. Las Herramientas CASE (Computer Aided Software Engineering/ Ingeniería de Software Asistida por Ordenador) fueron desarrolladas para automatizar esos procesos y aumentar la productividad en el desarrollo de software reduciendo el coste de los mismos en términos de tiempo y de dinero. (15)

Visual Paradigm for UML es una herramienta UML profesional que soporta el ciclo de vida completo de desarrollo de software, es independiente de la plataforma y está dotada de una buena cantidad de módulos para facilitar el trabajo durante la confección de un software. Posee, entre otras, las siguientes características que garantizan la calidad del producto final: (15)

- ❖ **Es profesional:** Brinda la posibilidad de crear un conjunto bastante amplio de artefactos utilizados con mucha frecuencia durante la confección de un Software.

- ❖ **Es amigable:** Puede ser utilizado en varios idiomas, sus componentes se encuentran relacionados, por lo que se hace muy fácil la creación de cualquier tipo de diagrama.
- ❖ **Brinda un número considerable de estereotipos a utilizar:** Esto que permite un mayor entendimiento de los diagramas.
- ❖ **Integración con distintos Ambientes de Desarrollo Integrados (IDE):** Se integra fácilmente con varios IDEs, entre ellos el de Visual Studio y el Eclipse.
- ❖ **Generación de documentación:** Brinda la posibilidad de documentar todo el trabajo sin necesidad de utilizar herramientas externas.
- ❖ **Disponibilidad en múltiples plataformas:** Microsoft Windows (98, 2000, XP, o Vista), Linux, Mac OS X, Solaris o Java.

Se utiliza Visual Paradigm en su versión 6.1, la cual cumple con el Standard UML 2.0, lo que permite principalmente, en comparación con las versiones anteriores, un modelado del proceso de negocio o BPMN¹³ (Notación de Modelado del Proceso de Negocio).

1.7.3 Lenguajes de Programación

JavaScript: Es un lenguaje de programación utilizado para crear pequeños programas encargados de realizar determinadas acciones dinámicas dentro del ámbito de una página Web. Se trata de un lenguaje de programación del lado del cliente que permite la programación de pequeños scripts, pero también de programas más grandes, orientados a objetos, con funciones y estructuras de datos complejas. (16)

Por otra parte pone a disposición del programador todos los elementos que forman la página Web, para que éste pueda acceder a ellos y modificarlos dinámicamente. Se utilizará este lenguaje para la implementación de algunos aspectos dinámicos como por ejemplo la validación de formularios y creación de ventanas emergentes con el propósito de mostrar algún tipo de información. (16)

PHP: Este es un lenguaje de programación de páginas dinámicas del lado del servidor, multiplataforma y totalmente gratuito, perteneciente a la familia de software libre, característica muy conveniente hoy en día, pues el software libre es aquel que, una vez obtenido, puede ser usado, copiado, estudiado, modificado y redistribuido libremente. (17)

¹³ Del inglés Business Process Modeling Notation.

Además el software libre suele estar disponible gratuitamente en Intranet, o a precio del coste de la distribución a través de otros medios; sin embargo no es obligatorio que sea así y, aunque conserve su carácter de libre, puede ser vendido comercialmente. PHP es un lenguaje multiplataforma, pues puede ejecutarse en entornos Unix, Linux y Windows, y es independiente del navegador que utilices, pues el código PHP se ejecuta en el servidor y al navegador solo devuelve texto. (17)

Entre sus principales ventajas están:

- ❖ Es un lenguaje multiplataforma.
- ❖ Capacidad de conexión con la mayoría de los manejadores de base de datos que se utilizan en la actualidad, destaca su conectividad con MySQL y PostgreSQL.
- ❖ Capacidad de expandir su potencial utilizando la enorme cantidad de módulos (llamados ext's o extensiones).
- ❖ Facilidad de aprendizaje y uso: Su sintaxis se basa en otros lenguajes de programación, principalmente en C y Perl.
- ❖ Posee una amplia documentación en su página oficial, entre la cual se destaca que todas las funciones del sistema están explicadas y ejemplificadas en un único archivo de ayuda.
- ❖ Es libre, por lo que se presenta como una alternativa de fácil acceso para todos.
- ❖ Permite las técnicas de Programación Orientada a Objetos.
- ❖ Biblioteca nativa de funciones sumamente amplia e incluida.
- ❖ No requiere definición de tipos de variables.
- ❖ Tiene manejo de excepciones.

1.8 Conclusiones

Durante el desarrollo del capítulo se dedicó un espacio a tratar brevemente de manera introductoria algunos temas sobre integración, servicios web y sus ventajas para obtener resultados satisfactorios en la integración de aplicaciones, se analizó la metodología a seguir, el lenguaje de modelado y de programación a tener en cuenta para plantear la solución a proponer.

CAPÍTULO #2: DEFINICION DEL PROBLEMA

2.1 Introducción

En este capítulo se desarrolla la primera fase del proceso de integración según plantea la metodología Total Business Integration (TBI) aunque, será sólo como modelo teórico, aplicando los procesos que esta define y haciéndola converger con la solución presentada.

En este capítulo se determina la necesidad de Integración como subsistema, quedando reflejado su objetivo, se definen y realiza el análisis de los procesos y se especifican los requerimientos funcionales y no funcionales.

2.2 Integración de Sistemas

El grado de conocimiento, automatización y mejora continua de los procesos en una entidad es una variable clave para la mejora de su eficiencia. Puesto que la realización de cualquier proceso en una entidad involucra típicamente el uso de varios sistemas informáticos, el uso de arquitecturas y tecnologías que permitan una integración ágil de sistemas para definir nuevos procesos o actualizar procesos existentes, y que a su vez permitan automatizar hasta donde sea posible dichos procesos, es altamente relevante para lograr el objetivo de mejorar el funcionamiento de la entidad. Sin embargo, existen ciertas dificultades que deben ser superadas para racionalizar y mejorar la gestión de los procesos de negocio, entre los que se destacan:

- ❖ **Sistemas heterogéneos deben intervenir en un mismo proceso.** Estos sistemas no están suficientemente integrados, lo que no hace posible la comunicación entre los mismos. La falta de integración introduce ineficiencias, como por ejemplo la necesidad de reintroducir datos manualmente de un sistema a otro, o la necesidad de tareas manuales que, de otro modo, podrían ser automatizadas.
- ❖ **La gestión del proceso es (parcialmente) manual;** los actores deben conocer el proceso y, manualmente, utilizar los sistemas necesarios para llevar a cabo las distintas actividades del

proceso. En último término, el control del proceso recae en los actores que intervienen en el mismo, lo que aumenta la posibilidad de errores y dificulta su gestión.

- ❖ **La comunicación entre los niveles de negocio y de tecnología de la entidad presenta ciertas dificultades** en lo que respecta a la comunicación mutua de requerimientos de negocio y disponibilidad de tecnología. Esto dificulta la puesta en marcha de nuevos procesos y la mejora de procesos existentes que involucren sistemas de la entidad.

El problema de la integración de aplicaciones consiste en hacer colaborar entre sí a aplicaciones distribuidas, heterogéneas y posiblemente autónomas.

La integración de aplicaciones para el sistema SIGAC tiene gran importancia, sin esta los módulos Planificación y Acciones de Control toman la forma de islas de información. Esto ocasiona que el valor de los sistemas individuales no sea aprovechado al máximo debido a su mismo aislamiento.

2.4 Subsistemas que se relacionan

Para el intercambio de información entre los módulos existen 2 procesos fundamentales, para definir estos procesos es necesario primeramente conocer de una forma clara y sencilla los módulos antes mencionados, teniendo en cuenta sus funcionalidades y las tecnologías que utilizan.

2.4.1 Módulo Planificación

El subsistema Planificación es una aplicación web que tiene como objetivo general automatizar el proceso de planificación y control de las *acciones de control* del Sistema Nacional de Auditorías. Este proceso responde a la dirección de Planificación, Análisis y Control (DPAC) del MAC, la cual es la encargada de realizar una planificación anual de todas las auditorías a partir de los resultados del año anterior y teniendo en cuenta las directivas trazadas para el año siguiente.

Entre las principales funcionalidades se encuentra:

- ❖ Adicionar acciones de control.
- ❖ Modificar acciones de control.
- ❖ Eliminar acciones de control.
- ❖ Buscar acciones de control.

- ❖ Mostrar detalles de las acciones de control.
- ❖ Cargar plan anual.
- ❖ Mostrar propuesta de plan.
- ❖ *Aprobar propuesta de plan.*
- ❖ Eliminar propuesta de plan.
- ❖ *Efectuar los Controles de Cumplimientos.*

Las funcionalidades sobre las cuales actúa la integración en este módulo son:

- ❖ *Aprobar propuesta de plan:* Una vez aprobado al plan, transparentemente al usuario, se envían los datos del mismo a Acciones de Control.
- ❖ *Efectuar los Controles de Cumplimientos:* Para llevar a cabo los Controles de Cumplimiento, necesitan, los datos que se recogen en los mismos, de Acciones de Control.

El subsistema Planificación está desarrollado con PHP en su versión 5.0 que es un lenguaje de programación de páginas dinámicas del lado del servidor, multiplataforma y gratuito; para el almacenamiento de datos utiliza el gestor de BD, PostgreSQL y el servidor Web, Apache.

2.4.2 Módulo Acciones de Control

El subsistema de Acciones de Control es una aplicación que tiene como objetivo automatizar el proceso de ejecución de acciones de control del sistema nacional de auditorías. Este proceso corresponde a las direcciones de:

- ❖ DACE (Dirección de Auditoría y Comprobaciones Especiales)
- ❖ DCG (Dirección de Control Gubernamental)
- ❖ DAG (Dirección de Auditoría Gubernamental)

Estas direcciones son las encargadas de la realización de las acciones de control (auditorías, comprobaciones especiales, controles, recontroles e inspecciones) a partir de los objetivos trazados para cada una de ellas.

Entre las principales funcionalidades que brinda se encuentran:

- ❖ Permite que cada documento de una o varias Acciones de Control sigan su flujo de trabajo correspondiente y cambien de estado (revisión, errores, aprobado) según el mismo vaya avanzando por los espacios de los usuarios.
- ❖ Permite obtener versiones de los documentos a los cuales se le hayan realizado algún cambio sin perder las anteriores.
- ❖ Permite asignarle tarea a otros usuarios a partir de un documento agregándole fecha de inicio y fin de la actividad y estos usuarios a su vez pueden cambiar el estado de su tarea según el cumplimiento que esta tenga, además de poder notificar una vez que se haya culminado.
- ❖ Permite la búsqueda de documentos de forma rápida a través de los metadatos o elementos únicos del documento.

Estos documentos pueden ser actas, informes, etc., un ejemplo de los mismos son las Órdenes de Trabajo.

El subsistema está organizado mediante espacios de trabajo, permitiendo que cada usuario sólo pueda trabajar en el espacio que le corresponde de acuerdo a la función que realiza, o además ser invitado en otro espacio, por el usuario que pertenezca al mismo, siempre y cuando este tenga permiso para hacerlo.

El subsistema Acciones de Control está desarrollado sobre el Sistema de Gestión Empresarial de Contenido Documental, Alfresco. Para el almacenamiento de la información utiliza el gestor de Base de Datos, PostgreSQL.

2.4.2.1 Alfresco

Es una solución Open Source para la gestión empresarial de contenido documental. Este Gestor de Contenidos Empresarial ofrece 4 grandes áreas (Ver Figura 6) de servicios o funcionalidades (18):



Figura 7. Las 4 grandes áreas del Gestor de Contenidos Empresarial Alfresco.

Todas las funcionalidades que tiene Alfresco facilitan al módulo Acciones de Control el desarrollo de las tareas que las direcciones (DACE, DAG y DCG) del ministerio necesitan.

Este gestor cuenta con una API de PHP que favorece la integración con otras aplicaciones, que utilicen esta tecnología, ventaja que se aprovecha para lograr la comunicación que se necesita entre los módulos del sistema SIGAC.

2.5 Definición de procesos de Negocio

Conociendo las principales funcionalidades de los subsistemas a integrar se pueden definir los procesos de integración, ya que esto permite identificar que información se necesita intercambiar y en que momento.

Para la confección del Plan Anual es necesario adicionar todas las Acciones de Control que tendría el mismo. En el proceso de creación de una acción de control se registran los siguientes datos de las mismas (ver Figura 7):

- ❖ Tipo de Acción de Control.
- ❖ Fecha de Inicio Planificada.
- ❖ Fecha de Fin Planificada.
- ❖ Datos de la Estructura a Comprobar:

- Nombre.
 - Afectación Económica.
 - UCAI¹⁴ a la que Pertenece.
 - Código REEUP.
 - Tipo de Estructura.
 - Particularidades.
- ❖ Datos de la estructura que ejecuta:
- Nombre.
 - Tipo de Estructura.



Figura 8. Interfaz No Funcional “Adicionar Acción de Control” del módulo Planificación.

¹⁴ Unidad Central de Auditores Internos.

La dirección de Planificación, Análisis y Control (DPAC) del MAC una vez aprobado el Plan Anual necesita que el mismo sea del conocimiento de las direcciones encargadas de su ejecución (DACE -Dirección de Auditoría y Comprobaciones Especiales-, DCG -Dirección de Control Gubernamental-, DAG-Dirección de Auditoría Gubernamental-) lo más pronto posible. Con los datos obtenidos de este Plan estas direcciones (con el módulo Acciones de Control) conforman las denominadas *Órdenes de Trabajo*, estas son planillas que utilizan los auditores durante todo el proceso de ejecución de una acción de control y donde se registra todo lo referente a las mismas (ver *Anexo #1*). Por cada acción de control del plan se crea una Orden de Trabajo.

Por otra parte la dirección de Planificación, Análisis y Control genera una serie de reportes, cada 15 días, como parte del control de cumplimiento que necesitan llevar a cabo. La información para estos reportes es generada en las *Órdenes de Trabajo* (en el Módulo Acciones de Control).

Estos procesos de intercambio entre Planificación y Alfresco, son de suma importancia. Sin el primero, el jefe de la unidad organizativa del Ministerio, quien ejecutará el trabajo, tendría que solicitar vía correo u otro mecanismo la información necesaria y luego introducirla a su sistema, mientras que sin el segundo los principales perjudicados serían los especialistas. Esto podría provocar para el trabajo general del ministerio:

- ❖ Demora en el comienzo de la ejecución de las acciones de control.
- ❖ No obtención de la información en tiempo real (información errónea en los Controles de Cumplimiento).
- ❖ Mayor trabajo y demora por parte de los especialistas y/o jefe de la unidad organizativa del Ministerio.
- ❖ Posible error y pérdida de información.

En general, retardaría todo el proceso que se desarrolla en la dirección de Planificación, Análisis y Control, tanto en la ejecución de las tareas como en la rectificación de la información introducida en el sistema. Todo esto, mirando desde el punto de vista de los usuarios finales (personal del ministerio de Auditoría y Control), pero para el proyecto Informatización del Ministerio de Auditoría y Control (IMAC), quienes son los encargados de desarrollar el sistema SIGAC, sería proveer una solución de software incompleta, dos sistemas completamente independientes, necesiándose uno del otro.

Para la ejecución de estos procesos de negocio se requiere configurar al acceso a Alfresco, por lo que se necesita tener registrados la dirección URL y el usuario y contraseña de administración, del mismo, por lo que esto generaría una nueva funcionalidad.

A continuación se representa la correspondencia entre las tareas empleadas por los usuarios con las respuestas que debe proporcionar el módulo integración.

Tarea Demandada por el Usuario	Respuesta del Sistema
Configurar Alfresco.	Obtener los datos especificados, validar los mismos (Sintácticamente y comprobando la conexión al servidor Alfresco) y almacenarlos.
Aprobar Plan Anual.	Crear Órdenes de Trabajo y almacenar (en Alfresco).
Solicitar Registros de Cumplimiento.	Proporcionar los contenidos solicitados (que se encuentran en Alfresco).

Tabla 1. Respuestas del Módulo Integración.

A continuación se muestra más detalladamente los procesos a llevarse a cabo entre dichos subsistemas.

2.5.1 Proceso: Crear Orden de Trabajo

En Acciones de Control se manipulan las **Órdenes de Trabajo**, que utilizan los auditores durante todo el proceso de ejecución de una acción de control y donde el jefe de grupo, o persona designada cumple la información requerida en las mismas, para esto necesita una serie de datos que son generados inicialmente en Planificación. Estos datos no son todos los expuestos anteriormente, necesarios para Adicionar una Acción de Control, sino un grupo de los mismos.

El proceso **Crear Orden de Trabajo** se puede dividir en una serie de actividades secuenciales (Ver Figura 8).

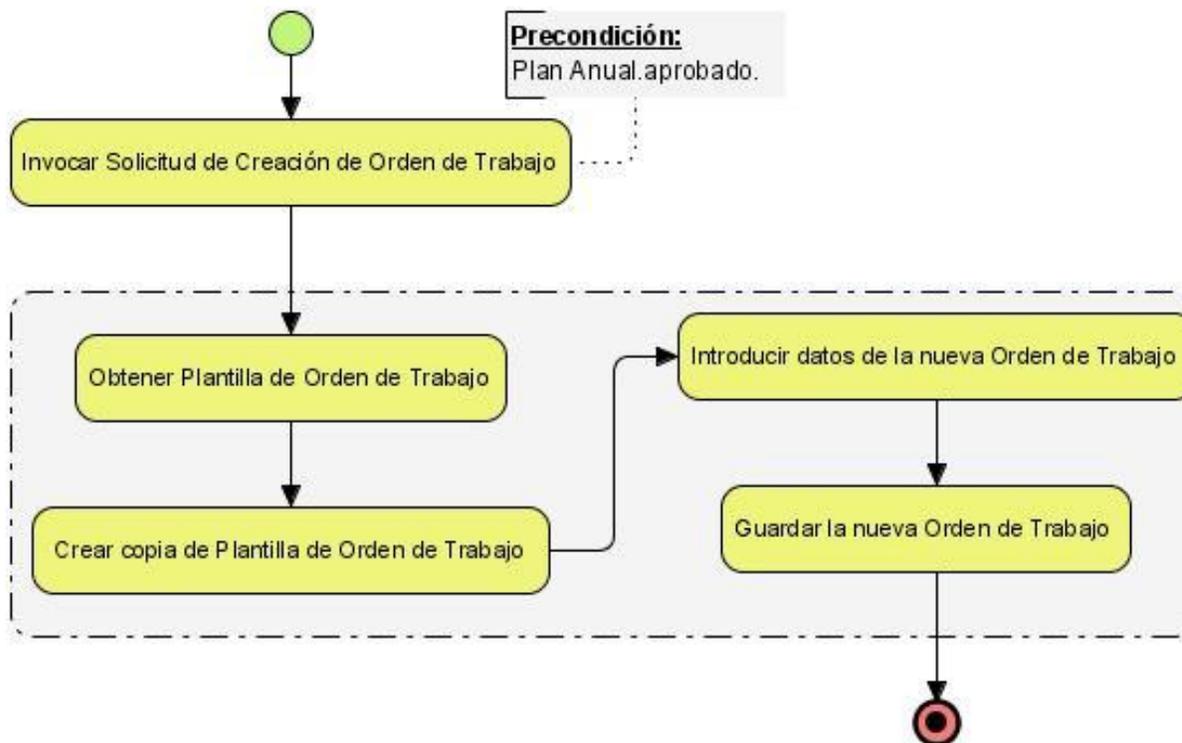


Figura 9. Proceso Crear Orden de Trabajo.

Modificación de datos en el Proceso Crear Orden de Trabajo	
Pre-condiciones:	Es necesario que el Plan de Trabajo esté aprobado.
Datos de Entrada:	Los siguientes datos de las Acciones de Control Planificadas: <ul style="list-style-type: none"> • Tipo de acción de control. • Nombre completo de la UCAI sobre la que se realiza la acción de control. • Nombre completo de la estructura sobre la que se realiza la acción de control. • Tipo de estructura sobre la que se realiza la acción de control. • Código REEUP de la estructura sobre la que se realiza la acción de control.

	<ul style="list-style-type: none"> • Dirección de la estructura sobre la que se realiza la acción de control. • Municipio de la estructura sobre la que se realiza la acción de control. • Provincia de la estructura sobre la que se realiza la acción de control. • Fecha de inicio planificada. • Fecha de fin planificada.
Datos de Salida:	Nueva Orden de Trabajo (documento Word)
Sistemas involucrados:	<ul style="list-style-type: none"> • Planificación (Sistema que solicita el envío) • Acciones de Control (Sistema que recibe)

Tabla 2. Modificación de datos en el Proceso Crear Orden de Trabajo.

Este proceso se inicia cuando el especialista de la dirección de Planificación, Análisis y Control aprueba el Plan Anual, también puede ser la Vice-Ministra de la DPAC quien es la máxima responsable de la Dirección de Planificación y encargada de Gestionar el Plan Anual de Auditorías. Inmediatamente aprobado este plan el módulo Planificación debe invocar la solicitud de creación de las Ordenes de Trabajo al módulo Integración, el cual es el encargado de crear la copia de la Plantilla Orden de Trabajo, luego insertar los datos pertinentes a esta y finalmente guardar la nueva Orden de Trabajo en una carpeta de Alfresco.

2.5.2 Proceso: Obtener Control de Cumplimiento

Para Planificación poder mantener correctamente actualizados los reportes que necesita generar, debe obtener los datos incluidos en las Órdenes de Trabajo (ver Anexo #1), de las acciones de control ejecutadas. Como este proceso se lleva a cabo en el módulo Acciones de Control es necesario obtener los datos desde ahí en el momento que se necesite. Este segundo módulo, luego de ejecutada una acción de control, le asigna el estado *en cierre*, una vez en el mismo, puede pasar a formar parte de los Controles de Cumplimiento de Planificación (reportes que genera este módulo).

La secuencia de este proceso es representado a continuación:

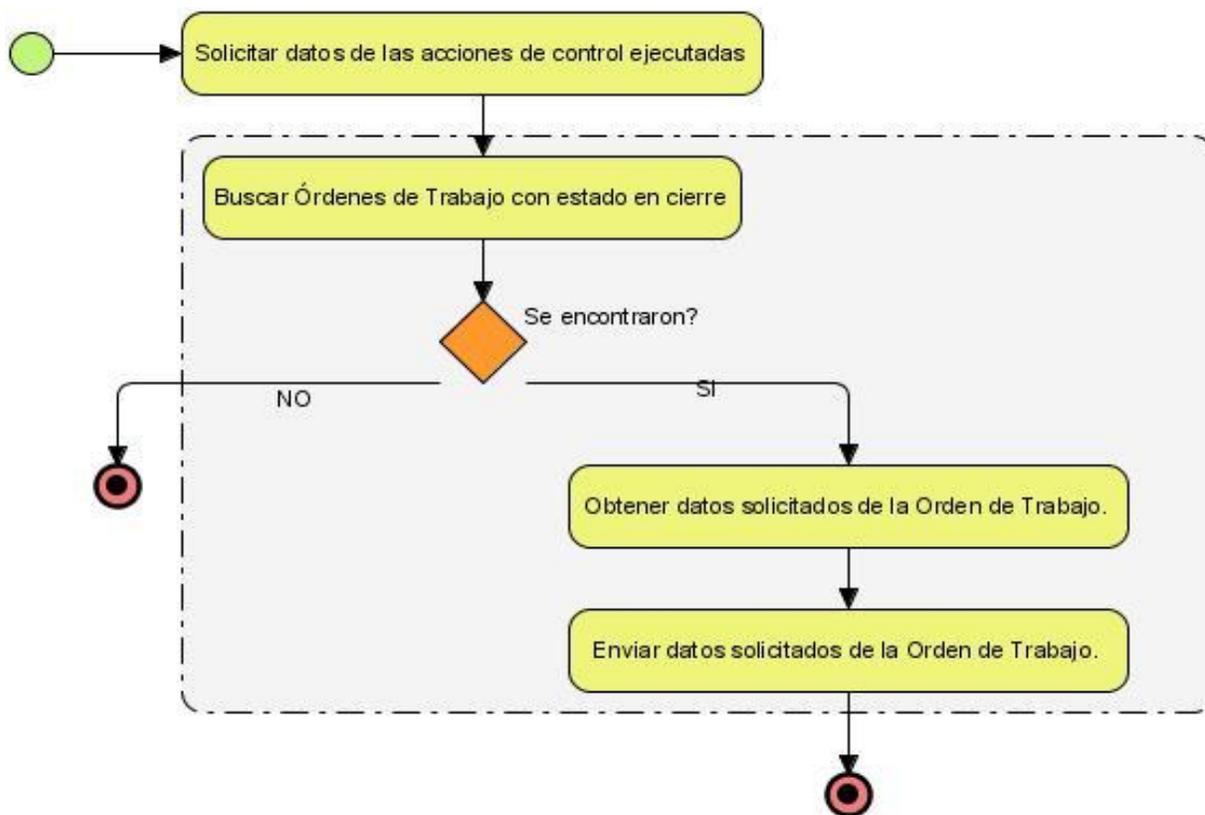


Figura 10. Proceso: Obtener Control de Cumplimiento.

Modificación de datos en el Proceso Obtener Control de Cumplimiento	
Datos de Entrada:	Metadatos de la Orden de Trabajo con estado <i>en cierre</i> (documento Word)
Datos de Salida:	Los siguientes datos de las Acciones de Control ejecutadas: <ul style="list-style-type: none"> • Fecha de inicio real. • Fecha de fin real. • Presunto Hecho Delictivo. (PHD) • Presunto Hecho de Corrupción. (PHC) • Calificación

	<ul style="list-style-type: none">• Resumen
Sistemas involucrados:	<ul style="list-style-type: none">• Planificación (Sistema que solicita la recepción)• Acciones de Control (Sistema que brinda los datos)

Tabla 3. Modificación de datos en el Proceso Obtener Control de Cumplimiento.

Este proceso lo inicia el especialista cuando necesita ver un reporte, también puede ser solicitado por la Vice-Ministra de la DPAC. A este evento el sistema Planificación solicita una serie de datos de cada una de las acciones nuevas ejecutadas al módulo Integración, el cual se encarga de buscar las nuevas Órdenes de Trabajo que se encuentren en estado en cierre, obtener los datos solicitados de las mismas y luego enviárselos a Planificación.

2.6 Proceso de Soporte

Para que se efectúen los procesos del negocio es necesario configurar la conexión al servidor de Alfresco, Como el sistema SIGAC aún no cuenta con un módulo Administración que se encargue de esta tarea, se suma a este módulo, una funcionalidad más que a continuación se explica.

2.6.1 Guardar Configuración de Alfresco

Para conectarse a Alfresco es necesario conocer su dirección URL, ya que a través de la misma se identifica de manera unívoca la dirección a la cual se va a acceder. También es necesario conocer usuario y contraseña de administración de Alfresco para obtener los datos necesarios de los espacios de trabajo de los diferentes usuarios.

Este proceso lo inicializa el administrador de sistema, este es el primer paso que debe tener en cuenta, una vez instalado el sistema SIGAC, de lo contrario no es posible la comunicación con Acciones de Control. (Ver Figura 10)

Para esto, una vez especificados URL, usuario y contraseña, por el administrador, es necesario primeramente validar la dirección, esta validación debe ser primeramente sintáctica y luego una comprobación para ver si es la dirección correcta, o sea, si con esta, efectivamente se puede conectar con

el otro módulo. Para evitar que el administrador introduzca mal la contraseña se le pedirá comprobar la misma.

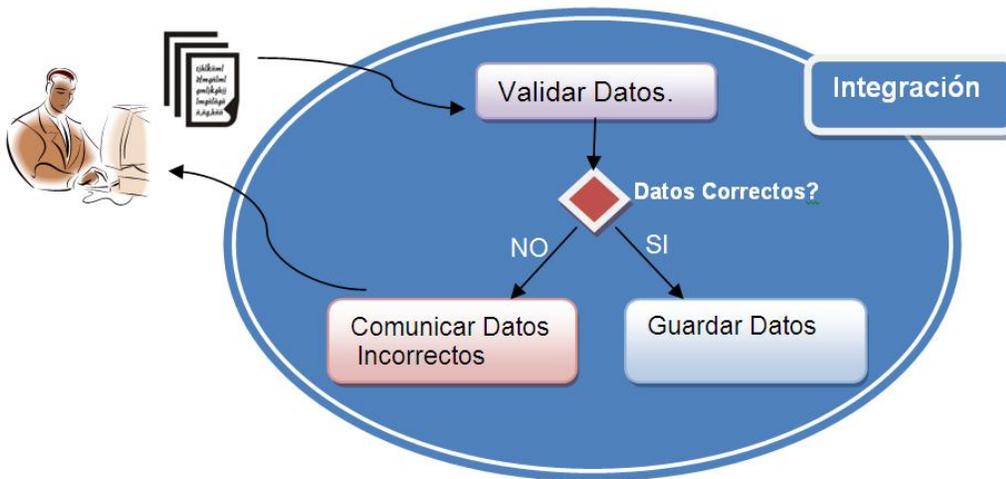


Figura 11. Proceso de Soporte Guardar Configuración de Alfresco.

Este proceso resulta de relevante importancia, sin encontrarse almacenados estos datos de configuración es imposible que se efectúen los procesos de negocio.

2.6 Relación de Requerimientos

La arquitectura de un sistema está guiada, en gran medida, por los requerimientos (funcionales y no-funcionales) que debe cubrir el sistema y normalmente se toma el subconjunto arquitectónicamente más importante de dichos requerimientos para definirla. Según Booch, Rumbaugh y Jacobson los requerimientos de un sistema se definen como:

- ❖ Condición o capacidad que necesita un usuario para resolver un problema o lograr un objetivo.
- ❖ Condición o capacidad que tiene que ser alcanzada o poseída por un sistema o componente de un sistema para satisfacer un contrato, estándar, u otro documento impuesto formalmente.
- ❖ Define que es lo que el sistema debe hacer, para lo cual se identifican las funcionalidades requeridas y las restricciones que se imponen.

- ❖ Es una característica que un sistema debe tener para cubrir alguna de las necesidades que lo motivan.

A continuación se muestran los requisitos o capacidades que debe tener el sistema para satisfacer las necesidades de integración, tanto los Funcionales como los No Funcionales.

2.6.1 Requisitos Funcionales

A continuación se muestran los requerimientos funcionales, los cuales indican lo que debe hacer el sistema. Los requerimientos funcionales identificados para la Integración fueron los siguientes:

RF 1. Enviar a Acciones de Control los datos de cada acción de control del Plan Anual.

1.1 Enviar los siguientes datos de una acción de control.

1.1.1-Tipo de acción de control.

1.1.1.1- Comprobaciones Especiales

1.1.1.2- Visitas de Supervisión

1.1.1.3- Inspecciones Gubernamentales

1.1.1.4- Auditorías

1.1.1.4.1- Auditoría de Gestión.

1.1.1.4.2- Auditoría Especial.

1.1.1.4.3- Auditoría de Seguimiento.

1.1.1.4.4- Auditoría de Cumplimiento.

1.1.1.4.5- Auditoría de Tecnología de la Información.

1.1.2- Nombre completo de la UCAI sobre la que se realiza la acción de control.

1.1.3- Nombre completo de la estructura sobre la que se realiza la acción de control.

1.1.4- Tipo de estructura sobre la que se realiza la acción de control.

1.1.4.1- Ministerio.

1.1.4.2- Grupo Empresarial.

1.1.4.3- Sucursal.

1.1.4.4- Unión.

1.1.4.5- Unidad básica

1.1.4.6- Establecimiento.

1.1.4.7- Empresa.

1.1.5- Código REEUP de la estructura sobre la que se realiza la acción de control.

1.1.6- Dirección de la estructura sobre la que se realiza la acción de control.

1.1.7- Municipio de la estructura sobre la que se realiza la acción de control.

1.1.8- Provincia de la estructura sobre la que se realiza la acción de control.

1.1.9- Fechas de Planificación.

1.1.9.1- Fecha de inicio planificada.

1.1.9.2- Fecha de fin planificada.

RF 2. Enviar a Planificación los datos de cada acción de control en estado *en cierre*.

2.1 Enviar los siguientes datos de una acción de control.

2.1.1- Controles de cumplimiento.

2.1.1.1- Fecha de inicio real.

2.1.1.2- Fecha de fin real.

2.1.1.3- Presunto Hecho Delictivo. (PHD)

2.1.1.4- Presunto Hecho de Corrupción. (PHC)

2.1.1.5- Calificación

2.1.1.6- Resumen

RF 3. Guardar Configuración de Alfresco.

3.1 Los datos necesarios para configurar Alfresco que se muestran son.

3.1.1- Dirección URL de Alfresco.

3.1.2- Usuario de Administración de Alfresco.

3.1.3- Contraseña de Administración de Alfresco.

3.1.4- Confirmar Contraseña de Administración de Alfresco.

2.6.2 Requisitos no Funcionales

A continuación se presentan los requisitos no funcionales, estas propiedades constituyen las características que hacen al producto atractivo, usable, rápido o confiable. En muchos casos dichos requisitos son fundamentales en el éxito del producto. Los requisitos no funcionales identificados para el desarrollo del Módulo Integración del SIGAC son:

❖ **Apariencia o interfaz externa.**

- La interfaz debe ser lo más sencilla y agradable posible, que favorezca el estado de ánimo del cliente y que combine correctamente los colores, tipo de letra y tamaño y que los iconos estén en correspondencia con lo que representan.

❖ **Usabilidad.**

- Debe ser de fácil y rápido manejo para los usuarios.

❖ **Rendimiento.**

- El módulo debe estar concebido para el consumo mínimo de recursos.
- La latencia del sistema no debe ser mayor de 15 segundos.

❖ **Soporte.**

• **Software.**

- Para el servidor de aplicaciones se requiere que esté instalado PHP en su versión 5.2.5 o superior.
- Se utilizará un servidor con sistema operativo Ubuntu 7.10 o superior.
- Servidor Web Apache en su versión 2.0.50 o superior con los módulos ssl y rewrite habilitados.
- En las computadoras de los clientes se garantizará versiones de Windows 2000 o superior, así como Linux y sus correspondientes distribuciones.
- Por parte del cliente se requiere un navegador capaz de interpretar JavaScript, por ejemplo: Internet Explorer 5.5 o superior y Mozilla Firefox 2.0.0.1 o superior.

• **Hardware:**

- Para las estaciones de trabajo:
 - Se requiere tengan tarjeta de red.
 - Se requiere tengan al menos 256 MB de memoria RAM.
 - Procesador de 800 MHz como mínimo.
- Para los servidores:
 - Se requiere tarjeta de red.
 - Se requiere tenga a menos 512MB de RAM.
 - Se requiere al menos 40GB de disco duro.
 - Procesador 2.0 GHz como mínimo.

❖ **Portabilidad, escalabilidad, reusabilidad.**

- El sistema debe ser multiplataforma.

❖ **Seguridad.**

- Sólo el administrador del sistema tendrá derecho a acceder a la interfaz Configurar Alfresco
- Los datos de configuración de Alfresco deben estar protegidos.

❖ **Legales.**

- El sistema se basará en el manual de normas y principios establecidos por el MAC.
- La mayoría de las herramientas de desarrollo deberán ser libres y del resto, las licencias deberán estar avaladas.

- El sistema tendrá en cuenta lo establecido por el “*Reglamento de las funciones de las Direcciones y Delegaciones del Ministerio de Auditoría y Control*”, en todo lo referido al desarrollo del sistema.

❖ **Confiabilidad y fiabilidad.**

- Debe garantizarse la disponibilidad del sistema en toda la jornada laboral (8 horas de trabajo diarias).

2.7 Conclusiones.

En el desarrollo del capítulo se mostraron las principales funcionalidades de los módulos Planificación y Acciones de Control, en el caso de este último se abordaron algunos temas sobre el Gestor de Contenidos Alfresco, en el cual se desarrolla este. Se definen los procesos de negocio (Proceso de Envío y Proceso de Recepción de Datos), así como el proceso de soporte a los mismos (Guardar Configuración de Alfresco), el cual permite tener almacenados la dirección url, usuario y contraseña de administración del Alfresco, proceso de vital importancia para lograr la comunicación entre ambos sistemas, además quedan establecidos los requerimientos funcionales y no funcionales del sistema. De esta manera, con un mayor nivel de entendimiento, será más fácil comprender la propuesta definida en el próximo capítulo de la investigación.

CAPÍTULO #3: SOLUCIÓN PROPUESTA

3.1 Introducción

En este capítulo se describe cómo desarrollar los procesos de Integración, definiéndose las diferentes clases y componentes que se deben incluir como parte de la solución propuesta, se proponen algunos patrones arquitectónicos, de diseño y patrones GoF. Este capítulo está enfocado a brindar la solución de Integración.

3.2 Arquitectura EAI en el sistema SIGAC

Si la integración se aplica sin seguir un enfoque estructurado de EAI, las conexiones punto a punto crecen al interior de la organización resultando en una gran cantidad que es difícil de mantener, además, de esta forma la responsabilidad para definir el receptor del mensaje recae en quien lo envía.

Aplicar desde este momento una solución EAI contribuiría, a facilitar el proceso de integración posterior de otros módulos, permitiendo que se desplace la responsabilidad para definir el receptor del mensaje en esta capa, solo aquí se tendría que modificar la lógica para dirigir los mensajes.

El gestor de contenidos Alfresco podría ser de interés para el intercambio de información hacia otros módulos (pensados los del proyecto ERP-Cuba), por lo que es factible la implementación de un módulo que se encargue de la integración, sin modificar el negocio de los restantes y permitiendo la reutilización de la información.

Los mecanismos de comunicación Máquina – Máquina han recorrido un extenso y escabroso camino hasta la actualidad, desde los antiguos mecanismos de envío/recepción sincrónicos hasta los servicios Web, pasando por la tecnología COM¹⁵, CORBA¹⁶ y otras que han sido relegadas debido a la fuerte interoperabilidad de los Servicios Web. Alfresco provee una serie de Servicios Web, ventaja a fructificar para lograr el acceso al mismo.

¹⁵ Del Inglés Common Object Model

¹⁶ Del Inglés Common Object Request Broker Architecture

El gestor de contenidos Alfresco, provee toda una librería de extensiones orientadas a la integración con otras soluciones basada en esta tecnología, permitiendo utilizar sus funcionalidades evitándose la necesidad de programar todo desde el inicio. Esta API de PHP facilita la comunicación con este proveedor de servicios, ventaja que ha de ser usada por el proyecto para llevar a feliz término la comunicación entre estos sistemas.

El módulo Integración funciona como puente entre los demás módulos, gestionando la relación entre los mismos:

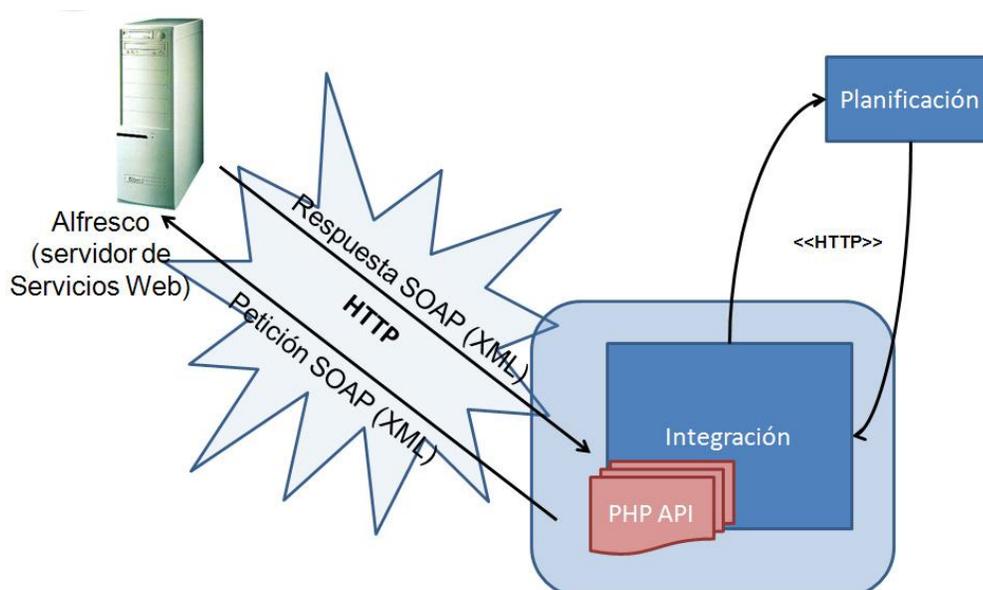


Figura 12. Arquitectura EAI en el sistema SIGAC.

3.3 Organización general

Anteriormente se mencionaba que el sistema SIGAC se desarrolla sobre el marco de trabajo del proyecto ERP-Cuba, el módulo Integración, como parte de la solución del sistema SIGAC debe estar desarrollado sobre este mismo marco de trabajo, y de esta forma la integración con todo el sistema estaría resuelta.

Este marco de trabajo para el desarrollo con php se basa en Zend framework y para el trabajo con las vistas ExtJS. Con el objetivo de darle solución a funcionalidades propias del proceso de negocio, el marco

de trabajo de ERP tiene desarrollado un grupo de componentes extendidos del ZendFramework (ZendExt).

3.4 Descripción de las interfaces

No se puede ver al módulo Integración de forma aislada al sistema en general, por lo que la relación de este con los demás del sistema y con Alfresco, se muestra como sigue:

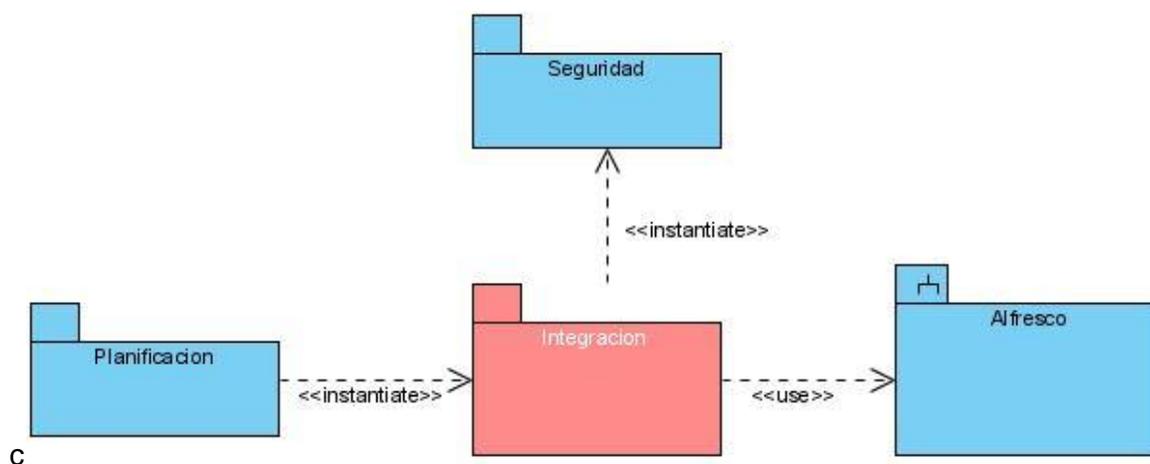


Figura 13. Subsistemas de la Solución de Integración del sistema SIGAC.

El paquete Planificación corresponde al módulo del mismo nombre, Alfresco al subsistema Acciones de Control, el paquete Seguridad corresponde al módulo que brinda la seguridad a todo el sistema SIGAC, exceptuando a Acciones de Control y el paquete Integración al módulo objeto de este trabajo de tesis.

Acciones de Control, tiene su propia seguridad al implementarse sobre el CMS¹⁷ Alfresco. Este gestor de contenidos tiene desarrollado todo un mecanismo de definición de permisos de acceso, gestión de usuarios, grupos y roles y seguridad de documentos.

¹⁷ Del inglés Content Management System.

Paquete Alfresco:

A continuación se muestran los Servicios Web que brinda Alfresco, los cuales son llamados directamente desde la API de PHP.

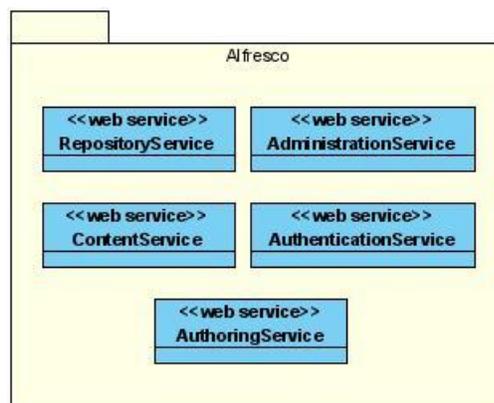


Figura 14. Servicios Web que brinda Alfresco.

Paquete Seguridad

El paquete Seguridad representa al Sistema de Gestión Integral de Seguridad (SIGIS), desarrollado por el centro UCID (Unidad de Compatibilización, Integración y Desarrollo de Software para la Defensa), el cual se incluye dentro de la solución del SIGAC. Este sistema es el encargado de gestionar la seguridad de los módulos Planificación e Integración, así como los que se integren posteriormente a esta solución, los cuales serán suscritos al mismo.

SIGIS se encarga de controlar las funcionalidades que brindan los módulos suscritos y de cada una de ellas las acciones que las soportan. Igualmente controla los roles y usuarios con permisos para acceder a cada una de ellas, así como los servicios consumidos y ofertados a otros subsistemas. De esta forma permite o deniega el acceso tan profundamente como se requiera y define un sistema de administración centralizado de seguridad que puede ser configurado para una política restrictiva tan severa como se desee.

Cada usuario tiene su perfil y podrá configurarlo según entienda. El sistema permite la administración de conexiones para que cada sistema pueda configurar a qué servidores o bases de datos conectarse, el usuario también tiene su usuario de base de datos para conectarse y realizar las acciones que tenga

asignadas. De igual forma son permitidos los accesos provistos por bases de datos centrales de “estructura y composición” que relacionan información relativa a personas y entidades con acceso a los sistemas que se controlan en el SIGIS.

En fin, SIGIS permite la gestión de la seguridad de sistemas de forma centralizada. Posibilitando la autenticación, autorización, administración de perfiles, administración de conexiones y la auditoría.

3.4.1 Módulo Integración

En la Figura 14 se muestra una vista lógica de la solución presentada para el paquete Integración, se presentan las clases que la conforman y sus relaciones, seguido una breve descripción de cada uno de los elementos. En la solución se incluye la presencia de la API de PHP de Alfresco la cual incluye una serie de clases y métodos que serán utilizados para el intercambio con Acciones de Control. Estos métodos son los encargados de hacer las llamadas a los Servicios Web de Alfresco, abstrayendo a los desarrolladores de Integración de esto.

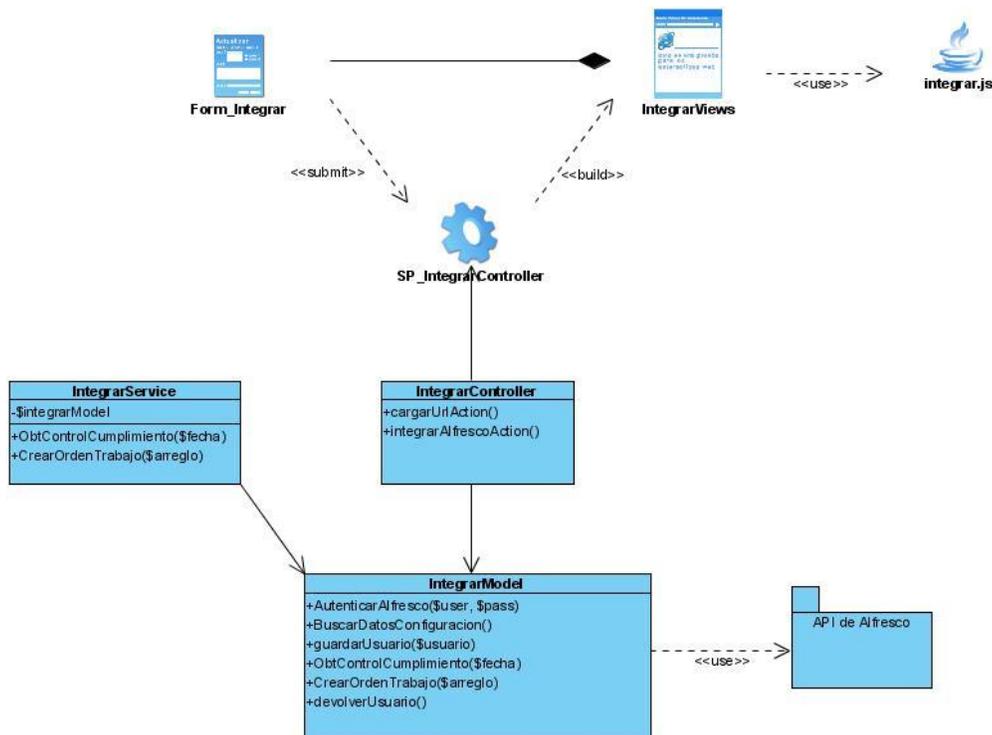


Figura 15. Vista lógica del paquete Integración.

A continuación se muestra una breve descripción de cada uno de estos elementos:

IntegrarViews: Es la interfaz que se muestra al administrador del sistema cuando va a configurar Alfresco.

Form_Integrar: Es el formulario que permite introducir los datos de configuración de Alfresco.

Integrar.js: Es el código JavaScripts utilizado para la interfaz de usuario.

IntegrarController: Permite el control de la entrada de información desde la interfaz, logrando obtener la configuración de Alfresco para guardar la misma en un fichero.

IntegrarModel: Esta clase es la que se encarga de llevar la lógica del negocio de Integración.

IntegrarService: Es quien permite la comunicación del módulo Planificación con Integración, esta clase sirve como portada para Planificación hacer las peticiones a Integración.

Para los dos procesos de negocio, Creación de Orden de Trabajo y Obtención de Control de Cumplimiento, desde Planificación, se invocan a los contratos `CrearOrdenTrabajo` y `ObtControlCumplimiento` respectivamente, mediante un llamado a la clase `IntegrarService`.

- ❖ `CrearOrdenTrabajo` se encarga de hacer las copias de la plantilla Orden de Trabajo, escribir en las mismas la información enviada desde Planificación y guardarlas en un espacio de trabajo en Alfresco.
- ❖ `ObtControlCumplimiento` se ocupa de acceder a las Órdenes de Trabajo, buscar las que estén en estado *en cierre* que no se hayan extraído y obtener los datos necesarios para devolvérselos a Planificación.

La introducción de la configuración de Alfresco mediante un formulario en la página cliente, `IntegrarViews`, se logra con la acción del contrato `integrarAlfrescoAction` de la clase `IntegrarController`.

- ❖ `integrarAlfrescoAction` es el contrato encargado de guardar los datos de configuración y realizar las validaciones pertinentes.

Una vez efectuadas las validaciones necesarias de los datos de configuración, se instancia al servicio que brinda Seguridad, insertarUsuarioSIGAC(\$usuario), quien se encarga de guardar esta configuración con las mismas políticas de seguridad aplicadas a todo el sistema. Para obtener estos datos, cuando sea necesario acceder a Alfresco, igualmente se invoca a otro servicio de Seguridad, devolverDatosSIGAC(\$usuario).

El prototipo no funcional de interfaz propuesta para el proceso Guardar Configuración de Alfresco es el que se muestra en la Figura 16 siguiendo el estándar definido para el sistema SIGAC.

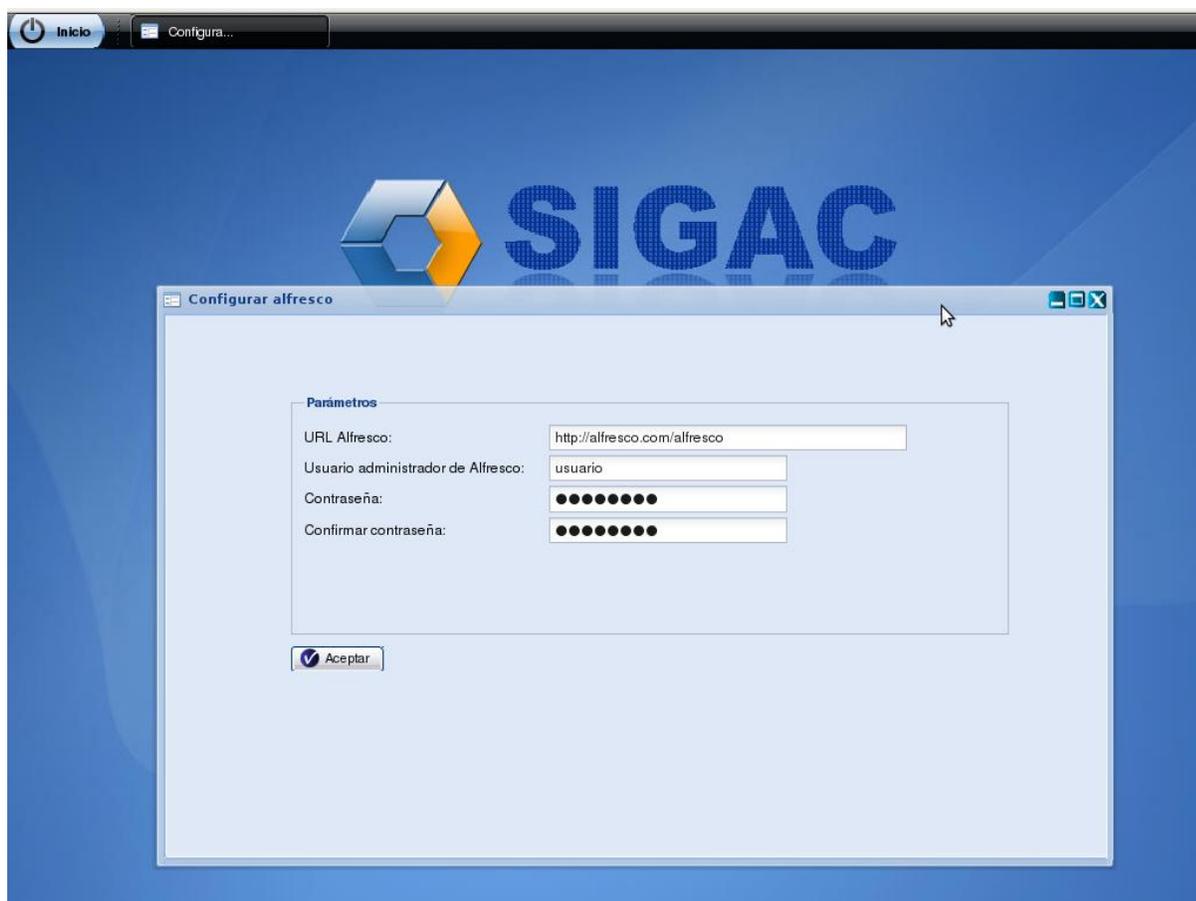


Figura 16. Prototipo no funcional de interfaz Configurar Alfresco.

3.5 Patrones en la Propuesta

Un patrón es una solución probada a un problema recurrente en un contexto específico. Su uso es más frecuente ya que estos permiten crear software altamente flexible, disminuyen el tiempo invertido en la generación de código y fomentan en gran medida la reutilización del mismo aumentando la depuración del software. Actualmente existen infinidad de patrones encaminados a resolver disímiles situaciones, en el presente se propone el uso de varios de ellos, los cuales serán explicados para facilitar su entendimiento.

3.5.1 Patrones de Integración

Los patrones de integración (o EAI) definen diseños comunes en el desarrollo de funcionalidades relacionadas con la integración de aplicaciones, especifican una manera de realizar ciertas tareas y ayudan a conocer con un lenguaje común determinadas cosas que desarrollamos habitualmente.

Los sistemas de EAI principalmente implementan dos patrones:

- ❖ **Mediación:** El sistema EAI actúa como el canal que vincula ambas aplicaciones o sistemas (Planificación y Acciones de Control) utilizando técnicas de encaminamiento. Cuando en una aplicación (Planificación) ocurre un evento que puede ser interesante a alguna otra aplicación (Acciones de Control) esta notifica al módulo Integración del sistema EAI, entonces el sistema EAI propaga esos cambios a la otra aplicación.
- ❖ **Federación:** En este caso, el sistema EAI actúa como un consolidador de información entre varias aplicaciones. Todos los accesos entre cualquiera de las aplicaciones son gestionados por el sistema EAI, y éste está configurado para exponer sólo la información relevante conectándose a Alfresco (aplicación externa), y efectuando todas las interacciones con Planificación (aplicación interna) sin intervención del agente externo.

Ambos patrones son usados en conjunto, sirviendo Integración como mediación entre las aplicaciones y al mismo tiempo sirve requerimientos para los módulos a integrar.

3.5.2 Patrones Arquitectónicos

El diseño del módulo Integración se basa en el estilo arquitectónico Modelo Vista Controlador (MVC) ya que el framework que se utiliza (Zend) está basado en el mismo. Este estilo está cimentado en el patrón

de arquitectura de software del mismo nombre, el cual permite separar la complejidad de un sistema en tres componentes elementales que tienen una función específica:

Modelo: Encapsula los datos de la aplicación y la lógica de negocio.

Vista: Extrae los datos desde el Modelo y los presenta al usuario.

Controlador: Es el que se encarga de dirigir el flujo de la aplicación, recibe las llamadas HTTP por parte del usuario y se encarga de traducirlas para el Modelo y la Vista.

De esta manera es posible separar el negocio de la vista comunicando ambas capas a través del controlador, facilitando la extensibilidad y el desacoplamiento. Estas vistas se ven representadas a través de las clases IntegrarViews que representa la vista, IntegrarController e IntegrarModel que representan las clases controladora y modelo respectivamente.

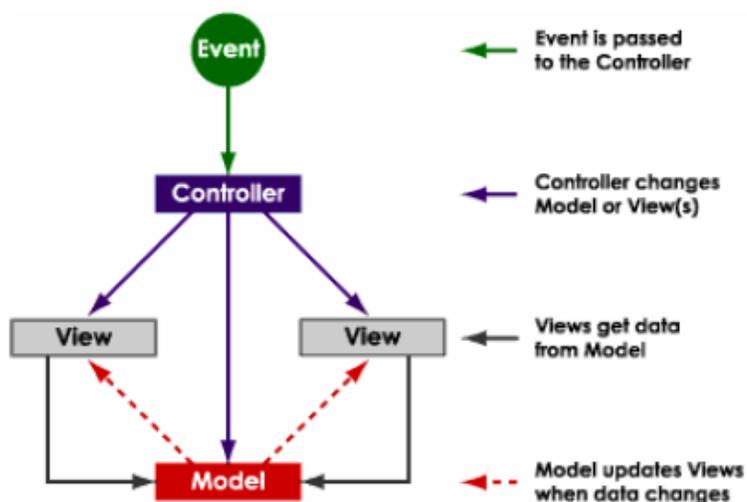


Figura 17. Estructura de MVC.

Además se deberán aplicar otros patrones arquitectónicos entre los que se pueden mencionar:

- ❖ **Inyección de Dependencias:** (Dependency Injection, DI) consiste en inyectar objetos a la clase en lugar de ser la propia clase quien cree el objeto, resuelve las dependencias de cada clase generando los objetos cuando se arranca la aplicación y luego inyectarlos en los demás objetos, por ejemplo, cuando el módulo Planificación accede a Integración, se hace una llamada al objeto

integrator quien se encarga de llamar al servicio implementado en Integración, igualmente sucede para guardar la configuración de Alfresco, Integración utiliza esta solución para acceder utilizando este mismo objeto a los servicios brindados por Seguridad (guardarConf y obtenerConf).

3.5.3 Patrones Generales de Asignación de Responsabilidades

GRAPS (General Responsibility Assignment Software Patterns, Patrones Generales de Software de Asignación de Responsabilidades)

- **Experto:** Consiste en asignar una responsabilidad al experto en información, es decir, la clase que tiene la información necesaria para cumplir con la responsabilidad, en este caso la clase IntegrarController la cual le asigna la responsabilidad a la clase IntegrarModel de la lógica de integración.
- **Creador:** Permite identificar quién debe ser el responsable de la creación o instanciación de nuevos objetos o clases, esto se pone en práctica en la clase IntegrarController la cual es la responsable de la instanciación o creación de la clase del modelo (IntegrarModel) que sería la clase con responsabilidades. La creación de instancias es una de las actividades más comunes en un sistema orientado a objetos.
- **Controlador:** Sirve como intermediario entre una determinada interfaz y el algoritmo que la implementa, IntegrarController encargada de llevar el control y seguimiento de la lógica del negocio ya que recibe los datos del usuario y los envía a las distintas clases según el método llamado.
- **Bajo Acoplamiento:** Consiste en crear pocas dependencias entre las clases con el objetivo que un cambio en alguna de ellas no afecte al resto o dicha afectación tenga una mínima repercusión en el resto de las clases potenciando la reutilización y disminuyendo la dependencia, esto se ve claramente cuando aplicamos el estilo MVC, de esta manera se garantiza una dependencia mínima entre las capas vista y modelo, en este caso, IntegrarViews e IntegrarModel las cuales se relacionan directamente con la clase IntegrarController.
- **Alta Cohesión:** Nos dice que la información que almacena una clase debe de ser coherente y estar en la mayor medida de lo posible relacionada con la clase de modo que cada elemento de nuestro diseño debe realizar una labor única dentro del sistema. Una

clase con mucha cohesión es útil porque es bastante fácil darle mantenimiento, entenderla y un aumento en la capacidad de reutilización. Por lo antes planteado recomendamos la aplicación del patrón desde los inicios del desarrollo.

3.5.4 Patrones de Diseño

Los patrones de Diseño o patrones GoF como también se les conoce están subdivididos en dos categorías:

- Creacionales
- Estructurales

Algunos de los patrones recomendados en la propuesta se muestran a continuación:

- ❖ **Creacionales:** patrones que se centran en la forma de crear las clases y sus instancias, así como el uso que recibirán una vez creadas.
 - **Patrón Singleton (Instancia única):** Muy importante uso de este patrón será en la aplicación de IoC. El frameworks implementa la clase IoC, la cual implementa el aspecto ISingleton, permitiendo una instanciación única.
- ❖ **Estructurales:** patrones que se centran en la composición y estructura de las clases y en la herencia entre ellas.
 - **Adapter (Adaptador o wrapper):** Realiza la función de adaptar una determinada interfaz para que pueda ser utilizada por una clase que de otro modo no podría accederla, la implementación de este patrón es esencial dentro del módulo integración, donde la API de PHP funciona como un adaptador entre este módulo y Alfresco, ya que sus interfaces son diferentes.
 - **Facade (Fachada):** Garantiza la existencia de una única instancia para una clase y la creación de un mecanismo de acceso global a dicha instancia. Permite que cierto tipo de dato esté disponible para todos los demás objetos de la aplicación, esto se pone en práctica en la clase IntegrarService la cual proporciona una fachada al módulo planificación encapsulando la lógica del módulo Integración que está implementada en la clase IntegrarModel.

3.6 Vista Física de Integración

Los elementos del modelo de diseño se implementan en términos de componentes, como ficheros de código fuente, ejecutables, etc. El modelo de implementación describe también cómo se organizan los componentes de acuerdo con los mecanismos de estructuración y modularización disponibles en el entorno de implementación y en el lenguaje o lenguajes de programación utilizados, y como dependen los componentes unos de otros.

Un componente es el empaquetamiento físico de los elementos de un modelo, como son las clases en el modelo de diseño (The Unified Modeling Language for Object-Oriented Development, Documentation set). Algunos estereotipos estándar de componentes son los siguientes:

- ❖ <<executable>> Es un programa que puede ser ejecutado en un nodo.
- ❖ <<file>> Es un fichero que contiene código fuente o datos.
- ❖ <<library>> Es una librería estática o dinámica.
- ❖ <<table>> Es una tabla de la base de datos.
- ❖ <<document>> Es un documento.

En la Figura 17 se muestran los subsistemas relacionados, donde se presentan el conjunto de configuraciones y contratos que definen la interacción entre los subsistemas a través de las interfaces (servicios).

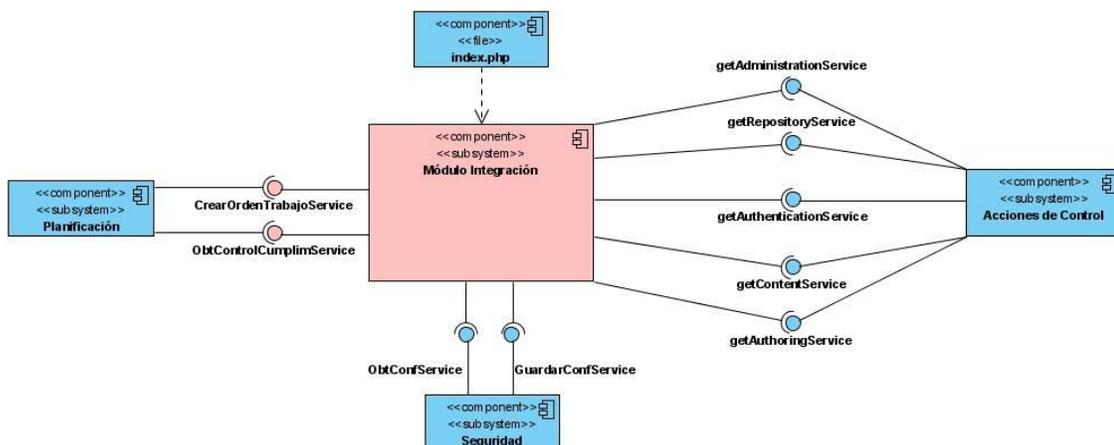


Figura 18. Subsistemas involucrados.

Esto provee una visión al equipo de desarrolladores del subsistema a implementar. A continuación se muestran los componentes del subsistema Integración.

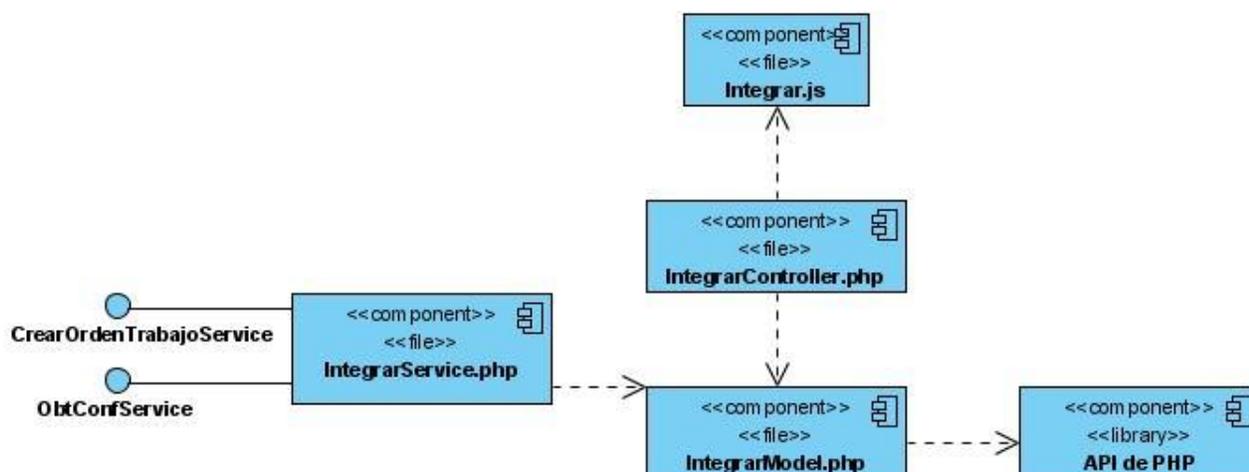


Figura 19. Componentes del subsistema Integración.

3.7 Despliegue del subsistema

Se propone una distribución física del sistema a través de nodos, es decir, se modela la configuración en funcionamiento del sistema (software y hardware) y las relaciones entre sus componentes.

Por el momento, en que solamente se incluye la integración entre los módulos Planificación y Acciones de Control el módulo Integración se encontrará en el mismo servidor de aplicaciones que utiliza Planificación y Seguridad, en un posterior despliegue donde se incluyan otros subsistemas a integrarse, en dependencia de la cantidad y complejidad de integración de los mismos se podría pensar en dividir estos módulos en diferentes servidores de aplicación, cuya distribución depende del caso que se presente, por el momento puede ser un único servidor de Aplicaciones, esto teniendo en cuenta que no son tantas las PCs Clientes conectadas al mismo.

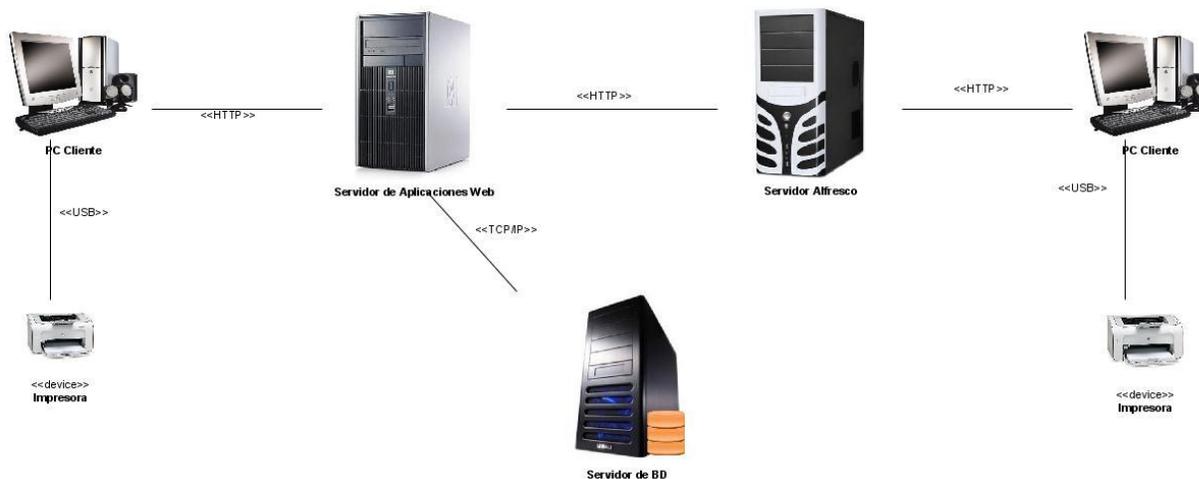


Figura 20. Despliegue del Subsistema.

Descripción de los nodos:

- ❖ **Impresora:** Satisfacen las necesidades de los clientes de impresión de reportes generados por la aplicación. Las mismas estarán conectadas por USB.
- ❖ **PC Cliente:** Representa a los usuarios finales del sistema los cuales se conectan al servidor web mediante HTTP, utilizando un navegador web. Estas requieren tener instalado lo siguiente:
 - Versiones de Windows 2000 o superior, así como Linux y sus correspondientes distribuciones.
 - Navegador como Internet Explorer 5.5 o superior, Mozilla Firefox 2.0.0.1 o superior o cualquier otro capaz de interpretar JavaScript.
- ❖ **Servidor de Aplicaciones Web:** Constituye uno de los nodos fundamentales, en él se encuentran los subsistemas, Seguridad, Planificación e Integración y el servidor web Apache.
 - Instalado en sistema operativo Linux.
 - Todos los nodos clientes deben poder acceder al mismo.
- ❖ **Servidor de Base Datos PostgreSQL:** Representa el servidor de Bases de Datos. En el se encuentra el gestor PostgreSQL.
 - Sistema operativo Linux y sus correspondientes distribuciones.

- ❖ **Servidor Alfresco:** En este nodo se guardan las imágenes y documentos generados por la aplicación, tales como informes y expedientes de auditorías. Será donde se ejecutará todo el módulo de Acciones de Control.

3.8 Conclusiones.

Durante el capítulo se desarrollaron temas específicos, esenciales a tener en cuenta para la futura implementación, así como las diferentes vistas mediante diagramas que facilitan la comprensión dando una visión gráfica de la solución a la problemática presentada. Este capítulo cubre las necesidades de información más importantes para los interesados en el tema.

CAPÍTULO #4: EVALUACION DE LA PROPUESTA

4.1 Introducción

Se entiende por evaluación al estudio de factibilidad que pretende detectar y evaluar riesgos y problemas, para luego buscar recomendaciones para solucionarlos y así evitar que el proyecto fracase en posteriores etapas.

En este capítulo se hace una evaluación mediante el uso de los criterios de un panel de expertos y el empleo de técnicas propuestas por el método Delphi con el objetivo de obtener una garantía sobre la calidad de la propuesta realizada. Se hace necesario evaluar el trabajo para contar con opiniones especializadas en el tema antes de continuar con la implementación del mismo dentro del proyecto SIGAC.

4.2 Evaluación

Las decisiones arquitectónicas determinan los atributos de calidad del sistema, por lo que es posible evaluar las decisiones arquitectónicas con respecto a su impacto sobre dichos atributos.

El objetivo en la evaluación de una arquitectura es saber si la misma puede habilitar los requerimiento de calidad y restricciones para asegurar que el sistema ha construir cumple con las necesidades de los stakeholders.

Las características principales que trae consigo una evaluación de la arquitectura son:

- ❖ *Es uno de los principales puntos de evaluación dentro del proyecto*, ya que errores en ella pueden traer que el proyecto fracase, lo que implicaría inconformidades tanto para el equipo de desarrollo como para el cliente que espera su producto, además de la pérdida de tiempo, esfuerzo y recursos económicos que no fructifican en una solución favorable.

- ❖ Puede ser realizada por gente Interna o Externa al proyecto, aunque lo más interesante es que sea realizada por personal Externo (Mentores o Arquitectos del Área), ya que estos llegan con una visión más amplia del tema a evaluar, a diferencia de los arquitectos internos los cuales ya vertieron su conocimiento en la solución dada hasta ese momento, resultándoles más difícil encontrar errores o mejoras a la misma.

La evaluación de la arquitectura se recomienda en dos momentos:

- ❖ Durante la propuesta:
- ❖ Durante el desarrollo:

Debido a que la propuesta de integración aún no ha sido implementada se pondrá en práctica un método de evaluación *durante la propuesta*.

4.3 Método Delphi

El método Delphi fue desarrollado por la Corporación Rand, es una metodología de investigación multidisciplinaria para la realización de pronósticos y predicciones a corto o largo plazo en varias ramas de las ciencias y la técnica, es considerada una de las técnicas subjetivas de pronosticación más confiables, para el desarrollo de este método se necesita la opinión de un grupo de Expertos los cuales son consultados individualmente mediante cuestionarios para que sus planteamientos no influyan en los ajenos.

4.3.1 Proceso de Selección de expertos.

El grupo de expertos se conformó con especialistas que poseen un amplio conocimiento y experiencia en el desarrollo de aplicaciones de Software, medimos algunos parámetros obligatorios a cumplir por los mismos, estos son:

- ❖ Graduado de nivel superior.
- ❖ Mantenga una vida activa en el desarrollo de productos informáticos.
- ❖ Conocimientos sobre arquitectura de Software.
- ❖ Un año de experiencia laboral como mínimo.

Escoger correctamente los expertos que conformarán el equipo propicia obtener resultados con calidad y una opinión grupal con alto nivel de consenso.

Un experto debe ser una persona experimentada en el tema que se trata, capaz de ofrecer valoraciones objetivas y hacer recomendaciones al respecto, es decir, no se resuelve nada en utilizar la opinión de un especialista en Metodología de la Investigación cuando el problema a tratar es puramente de arquitectura.

Por tal razón esta selección se realizó atendiendo fundamentalmente a la experiencia y los resultados obtenidos por cada encuestado, entre ellos encontramos líderes de proyectos, profesores de Ingeniería de Software y arquitectos de proyectos, garantizando de esta forma que las opiniones brindadas sean confiables y válidas para el objetivo propuesto.

Cantidad de Expertos seleccionados

No existe una norma que rija la cantidad de expertos que deben conformar un equipo para que este se considere válido, sin embargo, se plantea que a partir de 7 expertos el error disminuye exponencialmente y a partir de 30 la disminución del error continua disminuyendo pero de una manera poco significativa por lo que no compensa el incremento de costo y esfuerzo, por tal motivo se considera que el rango óptimo sería entre 7 y 30 expertos. En este caso contaremos con las valoraciones de 7 expertos los cuales se pueden ver sus datos en el anexo.

4.3.2 Objetivo a evaluar por los especialistas:

Valorar la propuesta de integración de los subsistemas del Sistema Informático de Gestión de Auditoría y Control (SIGAC), como parte del proyecto para la Informatización de Ministerio de Auditoría y Control.

4.3.3 Guía para la validación de la propuesta.

Una vez conformado el equipo de Expertos se elabora el cuestionario con los parámetros a medir teniendo en cuenta las características y necesidades de la propuesta de arquitectura para lograr la integración de los módulos Planificación y Acciones de Control.

El cuestionario se elaboró de manera que las respuestas fueran categorizadas con una puntuación entre 1 y 5 puntos.

- ❖ No adecuado → 1
- ❖ Poco Adecuado → 2

- ❖ Adecuado → 3
- ❖ Bastante Adecuado → 4
- ❖ Muy Adecuado → 5

A continuación se muestran los criterios de evaluación que fueron utilizados en el desarrollo de la encuesta, los cuales se agruparon por categorías:

- ❖ Criterios del método científico
 - Nivel de Calidad de la Investigación
 - Aportes Científicos Novedosos
- ❖ Criterios de Implantación
 - Necesidad de Uso de la propuesta
 - Satisfacción de las necesidades de integración
- ❖ Criterios de generalización
 - Nivel de comprensión
 - Facilidades de uso
 - Nivel de adaptación a otras aplicaciones.
- ❖ Criterios de impacto
 - Contribución al proceso de desarrollo de SW.
 - Posibilidades de aplicación.

En la Tabla 4 se muestran los valores otorgados por cada experto (E) a cada criterio (CRIT) respectivamente, una vez recogida las opiniones de los expertos se comienza el análisis de la información.

Criterios/Expertos	E1	E2	E3	E4	E5	E6	E7	Ep
CRIT 1	4	4	4	4	4	4	4	4
CRIT 2	3	4	3	3	4	4	4	3,5714286
CRIT 3	5	5	5	5	5	5	5	5
CRIT 4	4	5	5	4	5	5	5	4,7142857
CRIT 5	4	4	5	4	5	4	5	4,4285714
CRIT 6	4	4	5	4	5	4	5	4,4285714
CRIT 7	4	4	3	4	4	4	4	3,8571429
CRIT 8	4	5	4	5	5	4	5	4,5714286
CRIT 9	4	5	4	3	5	4	4	4,1428571

Tabla 4. Evaluación de los Expertos.

En la Figura 21 se representan los criterios evaluados así como el valor obtenido por cada uno de ellos expresado en porcentaje.

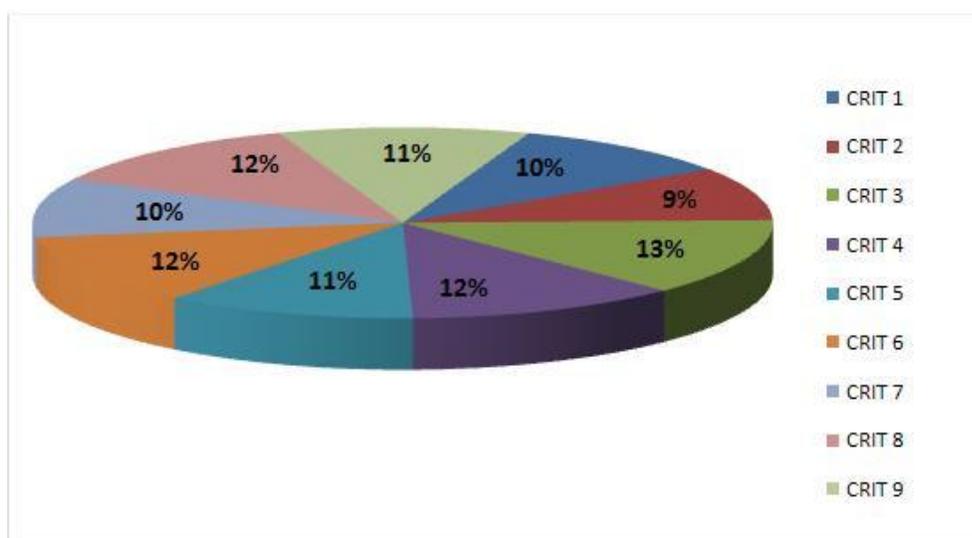


Figura 21. Representación del valor de los criterios en porcentaje.

Para verificar la consistencia en el trabajo de los expertos utilizamos el coeficiente de concordancia de Kendall y el estadígrafo Chi cuadrado (X^2).

4.3.3.1 Coeficiente de concordancia de Kendall y el estadígrafo Chi cuadrado (X^2).

1. Sea C el número de criterios que van a evaluarse (9) y E el número de expertos que realizan la evaluación (7).

Para cada criterio se determina:

ΣE : Sumatoria del peso dado por cada experto.

E_p : Puntuación promedio del peso dado por cada experto.

$M\Sigma E$: media de los ΣE .

ΔC : Diferencia entre ΣE y $M\Sigma E$.

2. Se determina la desviación de la media, que posteriormente se eleva al cuadrado para obtener la dispersión (S) por la expresión:

$$S = \Sigma (\Sigma E - \Sigma (\Sigma E / C))^2$$

3. Una vez conocida la dispersión se puede calcular el coeficiente de concordancia de Kendall (W):

$$W = 12 * S / E^2 (C^3 - C)$$

4. El coeficiente de concordancia de Kendall permite calcular el Chi cuadrado real:

$$X^2 = E (C-1) W$$

Los valores obtenidos de los cálculos anteriores se muestran en la Tabla 5.

Criterios/Expertos	E1	E2	E3	E4	E5	E6	E7	Σ E
CRIT 1	4	4	4	4	4	4	4	28
CRIT 2	3	4	3	3	4	4	4	25
CRIT 3	5	5	5	5	5	5	5	35
CRIT 4	4	5	5	4	5	5	5	33
CRIT 5	4	4	5	4	5	4	5	31
CRIT 6	4	4	5	4	5	4	5	31
CRIT 7	4	4	3	4	4	4	4	27
CRIT 8	4	5	4	5	5	4	5	32
CRIT 9	4	5	4	3	5	4	4	29
M Σ E	29,8137							
S	1800							
W	0,6122							
X ²	34,2832							
X ² (α; c-1)	8,351							

Tabla 5. Valores obtenidos.

5. Determinado el coeficiente de Kendall, es necesario realizar la prueba de hipótesis de que los expertos no tienen comunidad de preferencia. Con este criterio se intenta verificar la hipótesis fundamental:

H₀: No existe concordancia entre los criterios emitidos por los expertos.

Contra la hipótesis alternativa

H₁: Hay una concordancia no casual entre los criterios emitidos por los expertos.

El cálculo del coeficiente de concordancia de Kendall resultó ser de 0.6122, se presupone que si $W > 0.5$ existe convergencia de criterios entre los expertos, por tanto, como queda demostrado, $0.6122 > 0.5$, en resumen, los criterios alcanzados convergen, en caso de no existir convergencia se hace necesario repetir el trabajo de expertos. Para validar este resultado estadísticamente, se utilizó el estadígrafo Chi cuadrado (X^2).

6. El Chi cuadrado calculado se compara con el obtenido de las tablas estadísticas incluidas en el Anexo 3 y se considera que es válida la Hipótesis H₀ si se cumple:

$$X^2_{\text{real}} < X^2_{(\alpha, c-1)}$$

Por tanto, se rechaza la hipótesis H₀ demostrando que:

$$X^2_{\text{real}} > X^2_{(\alpha, c-1)}$$

34.2832 > 8.351

Quedando demostrada la hipótesis H_1 que plantea la existencia de concordancia en el trabajo de expertos.

- Una vez demostrada la consistencia del trabajo realizado por los expertos se puede definir el peso relativo de cada criterio (P) apoyándonos en la siguiente fórmula:

$$P_i = E_j / \sum_1^{10} E$$

Conociendo el peso de cada criterio y la calificación dada por los evaluadores se puede construir la siguiente tabla donde se muestra la calificación de cada criterio para obtener el valor de $P \times c$, donde (c), es el criterio promedio concebido por los expertos.

Criterios	Calificación Promedio (c)					P	P x c
	1	2	3	4	5		
CRIT 1				X		0,1033	0,4132
CRIT 2				X		0,0922	0,3688
CRIT 3					X	0,1291	0,6455
CRIT 4					X	0,1218	0,609
CRIT 5				X		0,1144	0,4576
CRIT 6				X		0,1144	0,4576
CRIT 7				X		0,0997	0,3988
CRIT 8					X	0,1181	0,5905
CRIT 9				X		0,107	0,428

Tabla 6. Calificación promedio para cada criterio.

- Se calcula el Índice de aceptación del proyecto (IA).

$$IA = \Sigma (P \times c) / 5$$

$$IA = 4.369 / 5$$

$$IA = 0.8738$$

- Y por último se determina la probabilidad de éxito de la propuesta

Rangos predefinidos de Índice de Aceptación

$IA > 0,7$ Existe alta probabilidad de éxito.

$0,7 > IA > 0,5$ Existe probabilidad media de éxito.

$0,5 > IA > 0,3$ Probabilidad de éxito baja.

$0,3 > IA$ Fracaso seguro.

Por lo que se puede concluir que la propuesta tiene una alta probabilidad de éxito.

4.4 Conclusiones

La evaluación de la propuesta de integración de los subsistemas del Sistema Informático de Gestión de Auditoría y Control (SIGAC) por el criterio de expertos valida y garantiza que dicha propuesta ya esta lista para pasar a una etapa superior concluyendo con resultados satisfactorios, por tanto, se puede asegurar que la propuesta cumple con los objetivos del negocio y es técnicamente factible. La misma fue evaluada por expertos seleccionados basándose en su preparación y experiencia acumulada en el desarrollo de productos informáticos.

CONCLUSIONES

Para llevar a cabo este trabajo se realizó un estudio teórico profundo con el objetivo de obtener información y alguna vía de solución propuesta por otras aplicaciones que se asemejarán a nuestra problemática, varias fueron las ideas encontradas pero ninguna resolvía en totalidad nuestra necesidad, de ahí surge la iniciativa de comenzar a trabajar utilizando la API de PHP que brinda el gestor documental Alfresco para vincularse con el lenguaje PHP, basándose esta en el uso de Servicios Web.

Esta investigación sirve como punto de partida para la construcción de otros sistemas con características similares, en la actualidad existen muchos proyectos en la universidad trabajando con el Marco de Trabajo del ERP y con el mismo gestor documental: Alfresco por lo que es muy probable que este módulo de integración le pueda resultar de utilidad.

Se documentó todo el diseño del componente de integración de la arquitectura del proyecto. Finalmente esta propuesta fue evaluada mediante el criterio de diversos expertos, obteniéndose resultados satisfactorios. Con el desarrollo de este trabajo de diploma se le da cumplimiento a todas las Tareas de la Investigación, dando respuesta a las necesidades de interoperabilidad entre los módulos Planificación y Acciones de Control requeridas para la feliz puesta en marcha del sistema SIGAC en el Ministerio de Auditoría y Control.

RECOMENDACIONES

Con el objetivo de ampliar y mejorar la documentación de este trabajo referente al proceso de integración se proponen las siguientes recomendaciones:

- ❖ Integrar el módulo de integración con los otros módulos Planificación y Acciones de Control.
- ❖ Realizar las pruebas de Integración para garantizar una mayor confiabilidad en el módulo.
- ❖ Incorporar en la documentación los diferentes patrones que se vayan utilizando a medida que se desarrolle el sistema.
- ❖ Utilizar esta documentación como medio de consulta y la reutilización de código para otras aplicaciones con fines similares.
- ❖ Estudiar la posibilidad de optimizar el código en aras de lograr una mayor eficiencia.

BIBLIOGRAFÍA

1. **Consejo de estado de la República de Cuba.** Ministerio de Auditoría y Control. [En línea] [Citado el: 15 de enero de 2009.] <http://www.minauditoria.cu/documentos/legales/DL-219.pdf>.
2. **Mendoza, Luis E.** Estudio de Tecnologías Middleware para Sistemas Peer-to-Peer. Caracas : s.n.
3. *Data Integration: Using ETL, EAI, and EII Tools to Create an Integrated Enterprise.* **White, Colin.**
4. *The Data Warehouse ETL Toolkit.* **Kimball, Ralph.** s.l. : Wiley.
5. **Morgenthal, JP.** *Enterprise Information Integration: A Pragmatic Approach.* s.l. : Bk&CD Rom edition., 2006.
6. **denodo technologies.** denodo technologies. [En línea] 1999. [Citado el: 28 de marzo de 2009.] <http://www.denodo.com>.
7. *Gestión de procesos e integración de sistemas: EAI, BPM, SOA y ESB.* **Lara, Rubén.**
8. **Banares, José Angel y J.Álvarez, Pedro.** Conceptos y Arquitecturas de Servicios Web. *Grupo de Sistemas de Información Avanzados.* [En línea] [Citado el: 6 de marzo de 2009.] <http://iaaa.cps.unizar.es/docencia/SW.html>.
9. **Benedí, Jennifer Pérez.** *Web Services.* Departamento de Sistemas Informáticos y Computación, Universidad Politécnica de Valencia. Valencia, España : s.n.
10. **Toledano., Moisés Daniel Díaz.** Web Services. Introducción y Escenarios para su Uso. *MOISESDANIEL.COM.* [En línea] [Citado el: 10 de marzo de 2009.] <http://www.moisesdaniel.com/es/wri/wsepsu.pdf>.
11. **Universidad Católica Andrés Bello.** XML y Web Services. [En línea] http://www.ucab.edu.ve/tl_files/IngenieriaInformatica/Electivas/XMLyWebServices.pdf.

12. **C., Benjamín González.** DesarrolloWeb.com. [En línea] [Citado el: 27 de marzo de 2009.] <http://www.desarrolloweb.com/articulos/1557.php>.
13. **Consortio de la Industria EAI.** *EAI Total Business Integration Methodology*.
14. **Larman, Craig.** *UML y Patrones*. s.l. : mexicana, 1999.
15. **Visual Paradigm International.** Visual Paradigm for UML. [En línea] <http://www.visual-paradigm.com/product/vpum/>.
16. **Maestros del Web.** ¿Qué es Javascript? [En línea] <http://www.maestrosdelweb.com/editorial/%C2%BFque-es-javascript/>.
17. —. ¿Qué es el PHP? [En línea] <http://www.maestrosdelweb.com/editorial/phpintro/>.
18. **Alfresco.** Alfresco.com. [En línea] <http://www.alfresco.com>.
19. **Brown, Laura.** Developing UML Architecture for EAI Products. [En línea] [Citado el: 2 de abril de 2009.] <http://www.systeminnovations.net/UMLforEAI.htm>.
20. **Mota, Sanjay Kumar K.** RoseIndia.net. [En línea] <http://www.roseindia.net/eai/EAIElephantanditscomponents.shtml>.
21. *EAI Total Business Integration Methodology.* **EAI Industry Consortium.** 2004.
22. *Total Business Integration Methodology Overview.* **Field, Stephen.** Canada : s.n., 2004.
23. **Schmidt, John G.** *EAI Methodology: The Theory of Application Integration*.
24. **World Wide Web Consortium.** W3C. [En línea] World Wide Web Consortium. [Citado el: 29 de marzo de 2009.] <http://www.w3c.es>.
25. **Schmidt, Jhon.** *EAI Methodology Call To Action*. s.l. : EAI Industry Consortium.
26. **Salvachúa, Joaquín.** Aplicaciones y Servicios Web (Web Services). *Internet de Nueva Generación*. [En línea] [Citado el: 28 de febrero de 2009.] <http://internetng.dit.upm.es/ponencias-jing/2004/Web2.pdf>.

27. **Risi, Walter.** SG. *Software GURU*. [En línea] 20 de septiembre de 2007. [Citado el: 8 de abril de 2009.] <http://www.sg.com.mx/content/view/422>.
28. **ISP Glossary.** ISP Glossary. [En línea] internet.com, The network for Technology Professionals. [Citado el: 12 de marzo de 2009.] <http://isp.webopedia.com>.
29. **North, Ken.** SQL Submit. *SQL Submit*. [En línea] [Citado el: 28 de marzo de 2009.] <http://www.sqlsubmit.com>.
30. **White, Colin.** TDWI. *Business Intelligence and Data Warehousing*. [En línea] The Data Warehousing Institute, 21 de mayo de 2006. [Citado el: 28 de marzo de 2009.] <http://www.tdwi.org>.
31. *Replication vs ETL vs EAI vs Ell.* **Tandrón, Lic. Iván M. Cárdenas.** La Habana. Universidad de las Ciencias Informáticas. : s.n., 2008.
32. **Coello González, Sayda y Hernández León, Rolando Alfredo.** *El Paradigma Cuantitativo de la Investigación Científica*. Habana : Universitaria, 2002. ISBN:959-16-0343-6.
33. **Definición.de.** Definición.de. [En línea] [Citado el: 15 de enero de 2009.] <http://definicion.de/integracion>.
34. *Revista de Software Libre ATIX.* **Arias, Yuniel Eliades Proenza.** 9 de marzo de 2009.
35. *El Lenguaje Unificado de Modelado.* **Rumbaugh, James, Jacobson, Ivar y Booch, Grady.** s.l. : Addison-Wesley.
36. **Booch, Grady.** *The Unified Modeling Language Use Guide*. s.l. : Addison Wesley, 1999.
37. **Syntel.** *Elements of a Successful Enterprise Application Integration*.
38. **SAP.** SAP. [En línea] [Citado el: 4 de abril de 2009.] http://www.sap.com/spain/company/press/sapclub/pdf/n_11/Tendencias.pdf.
39. **Proyecto ERP-CUBA.** *Especificación de la colaboración entre las clases que participan en la integración entre los subsistemas*. La Habana : s.n., 2009.

40. **Alfresco Developers.** Alfresco PHP API. [En línea] 10 de marzo de 2009.
41. *Web Services. Introducción y Escenarios para su uso.* **Toledano, Moisés Daniel Diaz.**
42. **González, Benjamín.** SOAP (Simple Object Access Protocol). [En línea] <http://www.desarrolloweb.com/articulos/1557.php>.
43. **Tec de Monterrey, Campus Monterrey.** Web Services. *HomePages.* [En línea] Tec de Monterrey, Campus Monterrey. [Citado el: 8 de marzo de 2009.] <http://homepages.mty.itesm.mx/al450951/>.
44. **SearchSOA.com.** SearchSOA.com. [En línea] 29 de enero de 2009. [Citado el: 10 de abril de 2009.] http://searchsoa.techtarget.com/tip/0,289483,sid26_gci1346341_mem1,00.html.
45. **Navarro, Orelvys Armas.** Aplicación de un procedimiento para el cálculo y evaluación de costos de calidad en la Química Ligera (ELQUIM). [En línea] <http://www.monografias.com/trabajos30/calculo-costos/calculo-costos.shtml>.

GLOSARIO DE TÉRMINOS

-A-

Interfaz de Programación de Aplicaciones (API): Del inglés **A**pplication **P**rogramming **I**nterface. Es el conjunto de funciones y procedimientos (o métodos, si se refiere a programación orientada a objetos) que ofrece cierta biblioteca para ser utilizado por otro software como una capa de abstracción. Una API representa una interfaz de comunicación entre componentes software. Se trata del conjunto de llamadas a ciertas bibliotecas que ofrecen acceso a ciertos servicios desde los procesos y representa un método para conseguir abstracción en la programación. De esta forma, los programadores se benefician de las ventajas de la API haciendo uso de su funcionalidad, evitándose el trabajo de programar todo desde el principio.

-C-

CMS (del inglés Content Management System): Un sistema de gestión de contenidos es un programa que permite crear una estructura de soporte (framework) para la creación y administración de contenidos, principalmente en páginas web, por parte de los participantes. Consiste en una interfaz que controla una o varias bases de datos donde se aloja el contenido del sitio. El sistema permite manejar de manera independiente el contenido y el diseño. Así, es posible manejar el contenido y darle en cualquier momento un diseño distinto al sitio sin tener que darle formato al contenido de nuevo, además de permitir la fácil y controlada publicación en el sitio a varios editores.

-E-

Estándar abierto: Es una especificación disponible públicamente para lograr una tarea específica. La especificación debe haber sido desarrollada en proceso abierto a toda la industria y también debe garantizar que cualquiera la pueda usar sin necesidad de pagar regalías o rendir condiciones a ningún otro. Al permitir a todos el obtener e implementar el estándar, pueden incrementar y permitir la compatibilidad e interoperabilidad entre distintos componentes de hardware y software, ya que cualquiera

con el conocimiento técnico necesario y recursos puede construir productos que trabajen con los de otros vendedores, los cuales comparten en su diseño base el estándar. Los estándares abiertos tienden a generar un mercado libre y muy dinámico, porque al no haber restricciones en su uso lo común es que sobre unos estándares abiertos se edifiquen otros y así sucesivamente. Es el caso de los estándares más comunes de Internet y por tanto de la Internet misma.

-I-

Integración: “La palabra **integración** tiene su origen en el concepto latino *integratio– onis*. Se trata de la **acción y efecto de integrar o integrarse** (constituir un todo, completar un todo con las partes que faltaban o hacer que alguien o algo pase a formar parte de un todo).”¹⁸

Interoperatividad: Es la condición mediante la cual sistemas heterogéneos pueden intercambiar procesos o datos. Es una propiedad que puede predicarse de sistemas de naturaleza muy diferente, como pueden ser los sistemas informáticos (en cuyo caso se le suele denotar también como neutralidad tecnológica), o los ferroviarios. La interoperatividad de la Web es una condición necesaria para que los usuarios (humanos o mecánicos) tengan un acceso completo a la información disponible.

-L-

Lucene: API para recuperación de información de código abierto, originalmente implementada en Java por Doug Cutting. Está apoyado por el Apache Software Foundation y se distribuye bajo la Apache Software License. Lucene tiene versiones para otros lenguajes incluyendo Delphi, Perl, C#, C++, Python, Ruby y PHP.

-M-

Metodología de desarrollo: Se entiende por metodología de desarrollo una colección de documentación formal referente a los procesos, las políticas y los procedimientos que intervienen en el desarrollo del software. En Inglés Software Development Methodology (SDM) o System Development Life Cycle (SDLC).

¹⁸**Definición.de.** Definición.de. [En línea] [Citado el: 15 de enero de 2009.] <http://definicion.de/integracion>.

La finalidad de una metodología de desarrollo es garantizar la eficacia y la eficiencia en el proceso de generación de software.

-O-

OASIS: Organization for the Advancement of Structured Information Standards. Es un consorcio internacional sin fines de lucro que orienta el desarrollo, la convergencia y la adopción de los estándares de comercio electrónico y servicios web.

-P-

Proceso: Del latín *processus*. Es un conjunto de actividades o eventos que se realizan o suceden (alternativa o simultáneamente) con un fin determinado.

Proceso de Negocio: Conjunto de tareas relacionadas lógicamente llevadas a cabo para lograr un resultado de negocio definido. Cada proceso de negocio tiene sus entradas, funciones y salidas. Es una colección de actividades estructurales relacionadas que producen un valor para la organización, sus inversores o sus clientes. Es, por ejemplo, el proceso a través del que una organización ofrece sus servicios a sus clientes.

-U-

UCAI:

-W-

World Wide Web Consortium (W3C): Es un consorcio internacional donde las organizaciones miembro, personal a tiempo completo y el público en general, trabajan conjuntamente para desarrollar estándares Web. La misión del W3C es guiar la Web hacia su máximo potencial a través del desarrollo de protocolos y pautas que aseguren el crecimiento futuro de la Web.

Web Services Interoperability Organization (WS-I): Su objetivo es fomentar y promover la Interoperabilidad de Servicios Web (Web Services Interoperability - WS-I) sobre cualquier plataforma,

sobre aplicaciones, y sobre lenguajes de programación. Su intención es ser un integrador de estándares para ayudar al avance de los servicios web de una manera estructurada y coherente.

ANEXOS

Anexo #1: Orden de Trabajo.

Proforma #1

ORDEN DE TRABAJO

Tipo de acción de control:
Orden:

Fecha de emisión:

DATOS DE LA ENTIDAD A AUDITAR (1)

Entidad:

Dirección:

Municipio:

Provincia:

Subordinada a:

Opera en moneda Librementemente Convertible: Sí____ No____ No procede: ____

Otros datos:

Tipo de Entidad:

Tipo de Subordinación:

TAREA A REALIZAR (2)

Planificada: Sí____ No____

Objetivos:-

Programas a aplicar:

Audidores- Días previstos para el trabajo:

PERSONAL DESIGNADO (3)

Director (a)

DEFICIENCIAS, E INSUFICIENCIAS O FALLOS TÉCNICOS, APORTES Y PRESUNTOS HECHOS DELICTIVOS (6)

Tipo de Acción de Control:

Temas	Deficiencias Cant.	Deficiencias Importe (MP con un decimal)
Activos Fijos	_____	(1) _____ (1)
Inversiones	_____	(2) _____ (2)
Inventarios	_____	(3) _____ (3)
Combustible	_____	(4) _____ (4)
Operaciones no comprendidas dentro del objeto social que afecten los resultados	_____	(5) _____ (5)
Efectivo		(6) _____ (6)
Nómina	_____	(7) _____ (7)
Recursos humanos	_____	(8) _____ (8)
Situación Financiera	_____	(9) _____ (9)
Otros	_____	(10) _____ (10)
	Total (1/10)	Total (1/10)

Proforma No. 01

Orden:

Fecha de emisión:

APORTES, CONTRIBUCIONES E IMPUESTOS NO EJECUTADOS
(MP con un decimal)

Importe

Retribuciones no reclamadas en tiempo	_____ (1)
Amortización de activos fijos	_____ (2)
Amortización de cargos diferidos	_____ (3)
Recursos obtenidos de las inversiones	_____ (4)
Saldo Libre de Ganancias	_____ (5)
Pago por los Fondos	_____ (6)
Impuesto de Circulación y otros	_____ (7)
Donaciones	_____ (8)
Ingresos indebidos no aportados al Presupuesto	_____ (9)
Otros conceptos	_____ (10)
Total (1/10)	_____
Claves (): Para uso de la secretaria	

PRESUNTOS HECHOS DELICTIVOS
(MP con un decimal)

IMPORTE

	_____ (1)
Inventario	_____ (2)
Activos Fijos	_____ (3)
Inversiones	_____ (4)
Combustible	_____ (5)
Efectivos	_____ (6)
Desvío de recursos	_____ (7)
Nóminas	_____ (8)
Dietas	_____ (9)
Alteración de Documentos	_____ (10)
Pagos indebidos	_____
Otros conceptos	
Total (1/10)	_____ (ver leyenda)

Órgano al cual se presentó la denuncia:

Fecha en la cual se presentó:

No. del Expediente de radicación

Claves (): Para uso del operador

__ / __ / __

Proforma No. 01

Orden: _____ Fecha de emisión: _____
DEFICIENCIAS E INSUFICIENCIAS O FALLOS TÉCNICOS DETECTADOS:

Detalles de las insuficiencias o fallos técnicos detectados en las visitas de supervisión y control calificadas de Regular o Malo.

- Por ciento en que existe incumplimiento de las Normas de Auditoría en lo referido a la confección del Informe de Auditoría y Papeles de Trabajo:	Por Ciento _____
--	---------------------

• Carencia de evidencias suficientes y competentes que soporten los hallazgos señalados	_____
• falta de profundidad y claridad al exponer las deficiencias	_____
• no refieren la legislación que se incumple	_____
• no se enuncia la muestra, no se cuantifican los hechos o violaciones, no se declaran las limitaciones.	_____
• Los Papeles de Trabajo no se confeccionan ni archivan correctamente dado que no se enuncian los objetivos, metodología, conclusiones, no se numeran y carecen de referencias, no quedan evidencias claras de la exploración y el planeamiento no siempre se realiza.	_____

	Por Ciento
- Por ciento en que no se recibe la adecuada atención metodológica o visitas de supervisión y control por	_____

parte del nivel superior de su sistema.	
- Por ciento que no informa los resultados de las auditorías a los trabajadores y organizaciones políticas y de masas, de la forma en que está orientado o no existen actas que evidencien su realización.	_____
- Por ciento en que se ha emitido alguna evaluación que no se corresponde con los hallazgos de la auditoría.	_____
- Por ciento de auditores no inscriptos en el Registro.	_____

Claves (): Para uso de la secretaria

PERIODO TRABAJADO POR CADA AUDITOR E INCIDENCIAS POR CAUSAS (7)

----- Incidencias por causas (UM: Día) -----

Nombre	Desde	Hasta	Días Trab. (1)	Asit MAC (2)	Act. Aut. (3)	Vaca ciones (4)	Licen cia (5)	Dcto 91 (6)	Asun Ofic. (7)	Enfer medad (8)	Asun Pers (9)	Aus Injus (x)

(x) Claves () Incidencias. Para uso de la secretaria

Horas -----	Días -----	Horas -----	Días -----	Horas -----	Días -----	Horas -----	Días -----
1	= 0.125	3	= 0.375	5	= 0.625	7	= 0.875
2	= 0.250	4	= 0.500	6	= 0.750	8	= 1.000

Anexo 2: Encuesta a Expertos para valoración de la propuesta.

Compañero (a):

La presente tesis se propone definir una propuesta de arquitectura para la integración de los subsistemas del Sistema Informático de Gestión de Auditoría y Control (SIGAC). Por sus conocimientos y experiencia en el tema tratado ha sido seleccionado para formar parte del equipo de expertos que se utilizan para validar el trabajo.

A continuación se presentan un conjunto de interrogantes de las cuales usted debe dar su criterio. Valore el grado de factibilidad de las mismas de acuerdo a la siguiente escala:

- 5 -- Muy Adecuado
- 4 -- Bastante Adecuado
- 3 -- Adecuado
- 2 -- Poco Adecuado
- 1 -- No adecuado

Criterios a Evaluar	Peso que usted le otorga (de 1 hasta 5).
1. Criterios del método científico	
1.1 Nivel de Calidad de la Investigación	
1.2 Aportes Científicos Novedosos	
2. Criterios de Implantación	
2.1 Necesidad de Uso de la propuesta	
2.2 Satisfacción de las necesidades de integración	
3. Criterios de generalización	
3.1 Nivel de comprensión	
3.2 Facilidades de uso	
3.3 Nivel de adaptación a otras aplicaciones.	
4. Criterios de impacto	
4.1 Contribución al proceso de desarrollo de SW.	
4.2 Posibilidades de aplicación.	

A continuación se explica en qué consiste cada criterio que usted debe evaluar.

Nivel de calidad de la Investigación: calidad de la concepción teórica de la investigación.

Aportes científicos novedosos: Aporte novedoso de la propuesta de integración.

Necesidad de uso de la propuesta: Nivel de necesidad de la creación de la propuesta de integración.

Satisfacción de las necesidades de integración: Cuan provechosa puede resultar la propuesta de integración.

Nivel de comprensión: Cuan entendible resulta tanto la teoría como la propuesta.

Facilidades de uso: Cuan fácil resulta emplear la propuesta de integración.

Nivel de adaptación a otras aplicaciones: generalización de la propuesta a diferentes proyectos productivos

Contribución al proceso de desarrollo de SW: Nivel de ayuda o aportes que brinda de manera general al proceso de desarrollo de SW.

Posibilidades de aplicación: Nivel de aplicabilidad en los proyectos de la UCI.

Nombre y Apellidos:

Breve descripción curricular:

Anexo 3: Tabla distribución chi-cuadrado inversa.

k \ P	0,01	0,05	0,10	0,20	0,25	0,30	0,40	0,50	0,60	0,70	0,75	0,80	0,90	0,95	0,99
1	0,000	0,004	0,016	0,064	0,102	0,148	0,275	0,455	0,708	1,074	1,323	1,642	2,706	3,841	6,635
2	0,020	0,103	0,211	0,446	0,575	0,713	1,022	1,386	1,833	2,408	2,773	3,219	4,605	5,991	9,210
3	0,115	0,352	0,584	1,005	1,213	1,424	1,869	2,366	2,946	3,665	4,108	4,642	6,251	7,815	11,34
4	0,297	0,711	1,064	1,649	1,923	2,195	2,753	3,357	4,045	4,878	5,385	5,989	7,779	9,488	13,28

5	0,554	1,145	1,610	2,343	2,675	3,000	3,656	4,351	5,132	6,064	6,626	7,289	9,236	11,07	15,09
6	0,872	1,635	2,204	3,070	3,455	3,828	4,570	5,348	6,211	7,231	7,841	8,558	10,64	12,59	16,81
7	1,239	2,167	2,833	3,822	4,255	4,671	5,493	6,346	7,283	8,383	9,037	9,803	12,02	14,07	18,48
8	1,647	2,733	3,490	4,594	5,071	5,527	6,423	7,344	8,351	9,524	10,22	11,03	13,36	15,51	20,09
9	2,088	3,325	4,168	5,380	5,899	6,393	7,357	8,343	9,414	10,66	11,39	12,24	14,68	16,92	21,67
10	2,558	3,940	4,865	6,179	6,737	7,267	8,295	9,342	10,47	11,78	12,55	13,44	15,99	18,31	23,21