



UNIVERSIDAD DE LAS CIENCIAS INFORMÁTICAS

Facultad # 4

***Sistema de gestión de información para el control interno en el
Ministerio de Comunicaciones***

***Trabajo de diploma para optar por el título de Ingeniero en Ciencias
Informáticas***

Autores

Gustavo Rodríguez Méndez

David Ruisánchez Jaime

Tutores

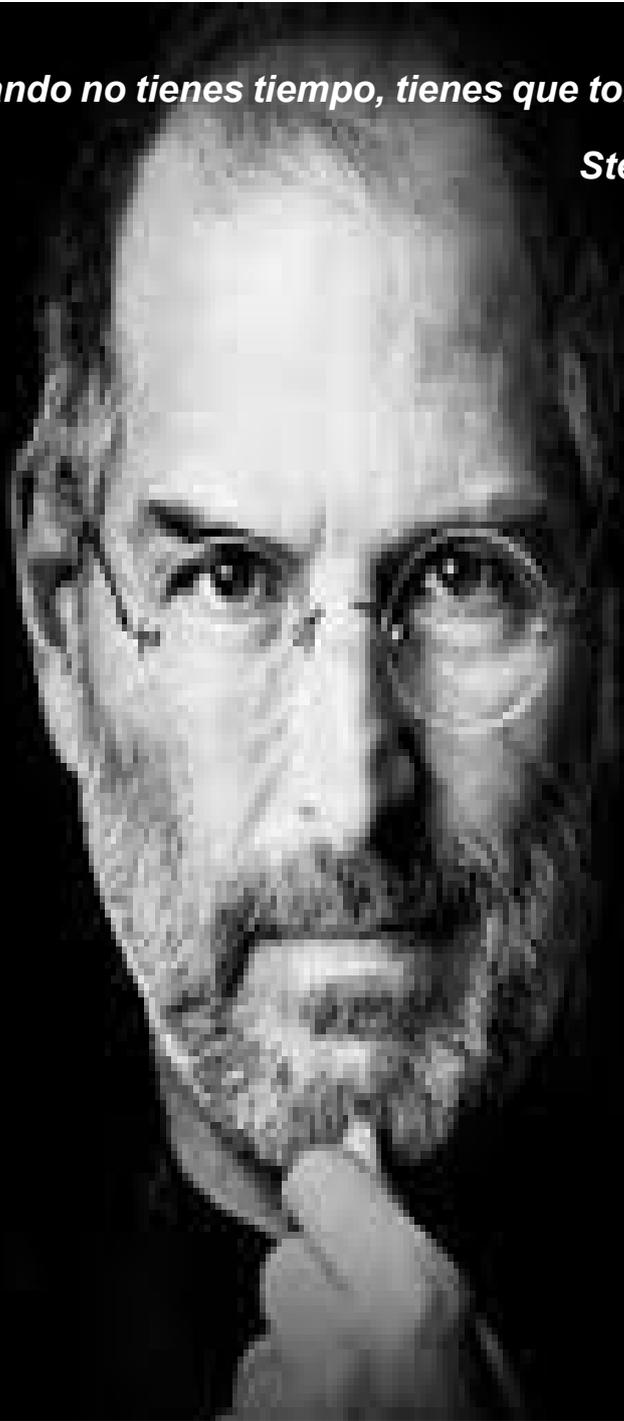
MSc. Marcel Puentes Rojas

Ing. José Manuel Santos Alonso

La Habana, junio de 2015

“A veces cuando no tienes tiempo, tienes que tomarlo prestado”

Steven Pauls Jobs



DECLARACIÓN DE AUTORÍA

Declaramos ser autores de la presente tesis y reconocemos a la Universidad de las Ciencias Informáticas y al Ministerio de Comunicaciones los derechos patrimoniales de la misma, con carácter exclusivo.

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

Gustavo Rodríguez Méndez

David Ruisánchez Jaime

Firma del autor

Firma del autor

Ing. José Manuel Santos Alonso

MSc. Marcel Puentes Rojas

Firma del tutor

Firma del tutor

Resumen

Los sistemas de gestión de información permiten a las organizaciones mejorar el procesamiento de la información. El Ministerio de Comunicaciones (MINCOM) es una institución que regula, impulsa y ordena el uso masivo de la Informática y las Telecomunicaciones de acuerdo a las políticas establecidas por el Estado Cubano. Actualmente este ministerio, en correspondencia con los Lineamientos Económicos y Sociales aprobados por el VI Congreso del Partido Comunista de Cuba pasa a ejecutar la gestión de gobierno de la informática y las comunicaciones en el país. Lo anterior ha traído consigo la generación de grandes volúmenes de datos procesados manualmente, lo cual dificulta su almacenamiento y procesamiento. El presente trabajo tiene como objetivo continuar el desarrollo del sistema que gestiona la información que se procesa dentro del control interno en el MINCOM. Para la realización del mismo fue necesario ejecutar un estudio sobre las metodologías de desarrollo de software, así como de las tecnologías y herramientas a utilizar, determinando las adecuadas. Finalmente para la validación de la propuesta de solución, fueron realizadas las pruebas establecidas por la metodología seleccionada. Como resultado de la investigación se obtuvo un sistema que contribuye al proceso de gestión de información dentro del control interno en el MINCOM.

Contenido

Introducción..... 1

CAPÍTULO 1: Fundamentación teórica. 5

1.1 Control interno 5

1.2 Sistemas de Gestión de Información..... 5

1.3 Sistemas de control interno 6

1.4 Informe COSO 6

1.4.1 Marcos de trabajo para el control interno de las Tecnologías de Información (TI) 7

1.4.1.1 Marco de trabajo COBIT 8

1.5 Soluciones Similares 9

1.5.1 Meycor COSO AG 9

1.5.2 COBIT Advisor 10

1.5.3 Meycor COBIT CSA 10

1.5.4 Sistema de gestión de información para el control interno en el Ministerio de Comunicaciones 11

1.6 Metodologías de desarrollo de software 11

1.6.1 Scrum 13

1.6.2 Programación Extrema (XP) 13

1.7 Marcos de trabajo 16

1.7.1 CodeIgniter..... 16

1.8 Lenguajes de programación..... 17

1.8.1 Lenguajes de programación del lado del cliente..... 17

1.8.1.1 CSS 17

1.8.1.2 JavaScript..... 17

1.8.1.3 HTML..... 18

1.8.2 Lenguajes de programación del lado del servidor..... 18

1.8.2.1 PHP 18

IV

1.9 Servidor Web	19
1.9.1 Apache	19
1.10 Herramientas CASE	19
1.10.1 Visual Paradigm	20
1.11 Sistemas gestores de base de datos	20
1.11.1 PostgreSQL	21
1.12 Entorno integrado de desarrollo	21
1.12.1 NetBeans	22
1.13 Conclusiones parciales del capítulo	22
CAPITULO 2: Propuesta de solución.	23
Introducción	23
2.1 Usuarios del Sistema	23
2.2 Funcionalidades del sistema	23
2.2.1 Lista de reserva del producto.	24
2.2.2 Características del sistema	24
2.3 Fase de exploración	25
2.3.1 Historias de usuario (HU)	25
2.3.2 Estimación de esfuerzo por HU	26
2.4 Planificación	27
2.4.1 Plan de duración de las Iteraciones	27
2.5 Iteraciones	28
2.5.1 Plan de entrega	29
2.6 Diseño	30
2.6.1 Prototipos de Interfaz	¡Error! Marcador no definido.
2.6.2 Descripción de las tarjetas CRC	30
2.7 Conclusiones parciales del capítulo	31
CAPÍTULO 3 Implementación y Pruebas	32

3.1	Introducción	32
3.2	Descripción de la arquitectura	32
3.3	Patrones de Diseño	33
3.3.1	Patrones GRASP	33
3.3.2	Patrones GOF	34
3.4	Modelo de datos	35
3.5	Implementación	36
3.5.1	Primera Iteración	36
3.5.2	Segunda Iteración	38
3.6	Pruebas	39
3.6.1	Pruebas Unitarias	40
3.6.2	Pruebas de aceptación	40
3.6.3	Resultado de las pruebas	41
3.7	Discusión de los resultados	42
3.8	Conclusiones Parciales	46
	Conclusiones Generales	47
	Referencias Bibliograficas	48
	Bibliografía	51
	Anexos	57
Anexo 1:	Entrevista	57
Anexo 2:	Historias de usuario	57
Anexo 3:	Tarjetas CRC	61
Anexo 4:	Descripción de las tareas de la primera iteración	62
Anexo 5:	Descripción de las tareas de la segunda iteración	65
Anexo 6:	Pruebas de aceptación	69

Índice de tablas

Tabla 1. Diferencias entre metodologías ágiles y tradicionales.....	12
Tabla 2.HU Permitir que un componente solo pueda ser evaluado por un usuario.	25
Tabla 3.HU Validar que al agregar un aspecto, los subcomponentes sean los correspondientes al componente seleccionado	26
Tabla 4.Estimación de esfuerzo por HU.....	26
Tabla 5. Plan de Iteraciones.....	27
Tabla 6. Plan de entrega.....	29
Tabla 7. Tarjeta CRC principal_admin.....	30
Tabla 8. Tarjeta CRC aspecto_controler.	31
Tabla 9. Tarea de ingeniería N01 HU1.	37
Tabla 10. Tarea de ingeniería N02 HU1.	37
Tabla 11. Tarea de ingeniería N012 HU5.	39
Tabla 12. Tarea de ingeniería N013 HU5.	39
Tabla 13. Caso de prueba de aceptación HU1_P1.....	40
Tabla 14. Resultado de las pruebas.	42
Tabla 15. HU Permitir a los administradores consultar datos estadísticos en forma de gráfica.....	57
Tabla 16. HU Mostar automáticamente el control para subir evidencia, si el aspecto es marcado de no.....	58
Tabla 17. HU Mostrar una evaluación de la guía después de ser llenada por el usuario.	58
Tabla 18. HU Asignar peso a cada aspecto de la guía de autocontrol.....	59
Tabla 19. HU Crear guías nuevas a partir de aspectos de guías existentes o a partir de una completamente desde cero.	59
Tabla 20. HU Determinar el porciento de cumplimiento de la guía.	59

Tabla 21. HU Mostrar a los administradores un resumen de los resultados de la aplicación de la guía de autocontrol en las entidades subordinadas al MINCOM.	60
Tabla 22. HU Exportar el resultado de la guía después de terminada esta.	60
Tabla 23. HU Incorporar preguntas referentes a la gestión de información basadas en el marco de trabajo COBIT como aspectos de la guía de autocontrol.	61
Tabla 24. Tarjeta CRC entidad_c.	61
Tabla 25. Tarjeta CRC componente_c.	62
Tabla 26. Tarjeta CRC subcomponente_c.	62
Tabla 27. Tarea de ingeniería N03 HU1.	62
Tabla 28. Tarea de ingeniería N04 HU1.	63
Tabla 29. Tarea de ingeniería N05 HU2.	63
Tabla 30. Tarea de ingeniería N06 HU2.	63
Tabla 31. Tarea de ingeniería N07 HU3.	64
Tabla 32. Tarea de ingeniería N08 HU3.	64
Tabla 33. Tarea de ingeniería N09.	64
Tabla 34. Tarea de ingeniería N010 HU4.	64
Tabla 35. Tarea de ingeniería N011 HU4.	65
Tabla 36. Tarea de ingeniería N014 HU6.	65
Tabla 37. Tarea de ingeniería N015 HU7.	65
Tabla 38. Tarea de ingeniería N016.	66
Tabla 39. Tarea de ingeniería N017 HU7.	66
Tabla 40. Tarea de ingeniería N018 HU8.	66
Tabla 41. Tarea de ingeniería N019 HU8.	66
Tabla 42. Tarea de ingeniería N020 HU9.	67
Tabla 43. Tarea de ingeniería N021 HU9.	67

Tabla 44. Tarea de ingeniería N022 HU10.	67
Tabla 45. Tarea de ingeniería N023 HU11.	68
Tabla 46. Tarea de ingeniería N024 HU11.	68
Tabla 47. Tarea de ingeniería N025 HU11.	68
Tabla 48. Caso de prueba de aceptación HU2_P2.....	69
Tabla 49. Caso de prueba de aceptación HU3_P3.....	69
Tabla 50. Caso de prueba de aceptación HU5_P4.....	70
Tabla 51. Caso de prueba de aceptación HU7_P6.....	70
Tabla 52. Caso de prueba de aceptación HU9_P7.....	71
Tabla 53. Caso de prueba de aceptación HU10_P8.....	71

INTRODUCCIÓN

Con el auge de las Tecnologías de la Información las Comunicaciones (TIC) a nivel mundial, la experiencia ha demostrado que aún dentro de las más diversas organizaciones, la dirección de un organismo no puede ejercerse eficazmente sino a través de un sistema de gestión de información que propicie el mejor uso de los recursos para alcanzar o superar los resultados esperados y que posibilite oportunamente, tomar las medidas necesarias para las acciones correctivas que se requieran realizar.

Esto plantea un reto para las empresas tanto productivas como de servicios, por la necesidad de lograr y mantener determinados niveles de competitividad, así como alcanzar resultados eficaces y eficientes en su gestión. Debido a que los entornos son cada vez más cambiantes con el desarrollo de las tecnologías de la información y las comunicaciones las organizaciones para mantener un alto estándar han recurrido cada vez más a los sistemas de control ratificando la gran importancia que estos presentan para controlar activos dentro de la empresa y posibilitar una mejor toma de decisiones por parte de la dirección. Con el fin de tener un eficiente control sobre los recursos de la empresa y su personal, en el mundo se han creado varias normas y estándares para gestionar el control interno de las organizaciones. Muchas entidades relacionadas con el tema han realizado importantes aportes a lo que conocemos como control interno y una mejor forma de gestionar el mismo. Dentro de los aportes más destacados se encuentra el informe COSO (por sus siglas en inglés Committee of Sponsoring Organizations). Fue creado en 1992 como iniciativa de 5 organismos de alto renombre para la mejora del control interno dentro de las organizaciones.

Basado en el informe COSO y lo planteado en la Resolución 60/11 de la Contraloría General de la República de Cuba se desarrolló una aplicación para el Ministerio de Comunicaciones en el año 2013 en correspondencia con los Lineamientos Económicos y Sociales, para llevar el control de gestión y fiscalización del Ministerio de Comunicaciones. El presente año se realizaron cambios en las leyes establecidas por la Contraloría General de la República, lo que provocó que cambiaran los requerimientos de las funcionalidades desarrolladas anteriormente. Después de realizada una entrevista al cliente la misma arrojó las siguientes deficiencias:

- La forma de visualizar la información relacionada con la guía de autocontrol en la interfaz de administración no es la correcta.

- Al adicionar un aspecto, el mismo se muestra duplicado y a la hora de obtener el procesamiento estadístico del mismo se genera un error.
- Al seleccionar el componente que está asociado a un aspecto solamente se debería mostrar los subcomponentes que estén comprendidos dentro del componente analizado y no todos en general.
- No se le asigna a cada aspecto el nivel de importancia que posea el mismo para la empresa en cuestión.
- No brinda la posibilidad a los administradores de gestionar su propia guía de autocontrol, o sea seleccionando los aspectos presentes en la guía actual que estén más acorde a su empresa y agregándole los suyos propios.
- La interfaz encargada de mostrar los gráficos correspondientes a los resultados de la aplicación de la guía de autocontrol no muestra con claridad la información necesaria y es difícil de entender.
- Además en la página de presentación de la interfaz de usuario no incorpora ninguna descripción del software y las acciones que puede realizar el usuario con el mismo.

Lo anterior unido al surgimiento de nuevos requerimientos, llevó a realizar una revisión de las funcionalidades existentes y agregar otras que solucionen los nuevos requisitos planteados por el Ministerio. Pero es necesario que tome en cuenta las más modernas normas y metodologías para los aspectos del control interno, haciendo énfasis en el control de las tecnologías de la información. Visto de esta forma es necesario realizar un estudio más detallado sobre posibles herramientas y guías que sean capaces de proporcionar una mayor comprensión sobre el tema y pueda hacer de las mismas un mayor uso para proporcionarle al software existente los parámetros suficientes para llevar un buen control de las tecnologías de la información.

Debido a lo anteriormente expuesto se plantea como **problema a resolver**: ¿Cómo facilitar a las empresas del Ministerio de Comunicaciones la aplicación de las evaluaciones de control interno acorde a sus necesidades?

Se tiene como **objeto de estudio**: sistemas de gestión de información para el control interno. Comprendido en el **campo de acción**: sistemas de gestión de información para el control interno basados en el informe COSO y el marco de trabajo COBIT en el Ministerio de Comunicaciones.

Para dar solución al problema anterior se define el siguiente **objetivo**: Desarrollar las funcionalidades que permitan a los usuarios del sistema de control interno del Ministerio de Comunicaciones aplicar evaluaciones adecuadas a las necesidades de sus organizaciones.

Preguntas Científicas

- ¿Cómo sistematizar los fundamentos teóricos relacionados con el control interno?
- ¿Cómo desarrollar las funcionalidades del sistema para chequear el control interno en el Ministerio de Comunicaciones basado en el informe COSO e incluyendo el marco de trabajo COBIT en su versión 3?
- ¿Cómo implementar las funcionalidades del sistema para chequear el control interno en el Ministerio de Comunicaciones basado en el informe COSO e incluyendo el marco de trabajo COBIT en su versión 3?
- ¿Cómo probar las funcionalidades del sistema para chequear el control interno en el Ministerio de Comunicaciones basado en el informe COSO e incluyendo el marco de trabajo COBIT en su versión 3?

Tareas a resolver:

- Revisión bibliográfica para generar el marco teórico sobre el control interno.
- Estudio de las soluciones similares y de las herramientas y tecnologías para el desarrollo de la propuesta de solución.
- Redefinición de la arquitectura y adecuación de la base de datos.
- Implementar las funcionalidades necesarias para el correcto funcionamiento del sistema para chequear el control interno en el Ministerio de Comunicaciones.
- Realizar las pruebas para verificar el correcto funcionamiento del sistema para chequear el control interno en el Ministerio de Comunicaciones.

Para cumplir los objetivos planteados se utilizarán los métodos científicos que se describen a continuación:

Métodos Teóricos:

- **Análisis-síntesis:** se utilizó para analizar cómo se han desarrollado los sistemas de gestión de información y extraer características esenciales de los mismos.
- **Histórico-lógico:** para analizar la evolución de los sistemas de gestión de información y las normas de control interno.

Métodos Empíricos:

- **Observación:** para verificar los aportes de un software de gestión del control interno en el Ministerio de Comunicaciones y determinar los problemas que el mismo puede resolver y los beneficios que puede aportar al Ministerio y sus empresas subordinadas.
- **Entrevista:** para realizar preguntas al cliente sobre la aplicación anterior y conocer sus deficiencias.

CAPÍTULO 1: FUNDAMENTACIÓN TEÓRICA.

Hoy en día las aplicaciones informáticas forman parte esencial en la mayoría de las empresas modernas. Los sistemas de gestión de información y control interno constituyen una de las herramientas que facilitan el trabajo con la información y mejoran cualquier área de la empresa que deba procesar datos para su funcionamiento, incluso la toma de decisiones.

1.1 Control interno

El control interno es el proceso integrado a las operaciones con un enfoque de mejoramiento continuo, extendido a todas las actividades inherentes a la gestión, efectuado por la dirección y el resto del personal; se implementa mediante un sistema integrado de normas y procedimientos, que contribuyen a prever y limitar los riesgos internos y externos, proporciona una seguridad razonable al logro de los objetivos institucionales y una adecuada rendición de cuentas. (1)

Manuel Mantilla Blanco expresó que el control interno, según el informe COSO es *“un proceso, efectuado por el consejo de directores, la administración, u otro del personal de una entidad, diseñado para proveer seguridad razonable en relación con el logro de objetivos en las siguientes categorías:*

- *Eficacia y eficiencia de las operaciones.*
- *Confiabledad de la información financiera.*
- *Cumplimiento de las leyes y normas aplicables”(2).*

1.2 Sistemas de Gestión de Información

Según menciona Gloria Ponjuán cuando se menciona gestión de información se refiere a la gestión que realiza un Sistema de Información y la define como: *“el proceso mediante el cual se obtienen, despliegan o utilizan recursos básicos (económicos, físicos, humanos, materiales) para manejar información dentro y para la sociedad a la que sirve. Tiene como elemento básico la gestión del ciclo de vida de este recurso y ocurre en cualquier organización. Es propia también de unidades especializadas que manejan este recurso en forma intensiva, llamadas unidades de información.”(3)*

También José A. Moreiro plantea que un sistema de gestión de información *“no es más que un sistema integrado y automatizado para proveer la información necesaria para sostener la operación, gestión y toma de decisiones en una organización. Los datos*

que se manejan son sobre todo fechas, nombres, códigos, cuentas o cálculos y números de identificación, los cuales a su vez se mezclan con datos y otros documentos procedentes de otros sistemas de información.”(4)

Hay muchos autores que abordan sobre este tema y la importancia que tiene en las empresas donde haya un gran flujo de información y se necesite controlar la misma para disminuir el tiempo y los recursos empleados en el trabajo con esta. Los sistemas de gestión de información se hacen más útiles cada día debido a su gran aporte en los procesos administrativos de las empresas modernas por eso la necesidad de utilizarlos en este tipo de ámbitos se ha incrementado notablemente.

1.3 Sistemas de control interno

Considerando la importancia del control interno para la administración de una empresa Schuster enuncia que *“el sistema de Control interno comprende el plan de organización y todos los métodos coordinados y medidas adoptadas dentro de una empresa con el fin de salvaguardar sus activos, verificar la confiabilidad y corrección de los datos contables, promover la eficiencia operativa y fomentar la adhesión a las políticas administrativas prescritas.” (5)*

La mayoría de las empresas actuales tienen la meta de ser más capaces y competitivas y muchas centran su atención en el ambiente que las rodea, como competidores, proveedores y otros factores externos, sin darse cuenta que sus mayores fortalezas son sus activos internos, por lo mismo los sistemas de control interno son esenciales para proporcionar el desarrollo de las empresas desde su interior y para elevar la seguridad y mantener el orden en los diferentes sectores que controla este sistema, principalmente en lo relacionado con el flujo de información.

1.4 Informe COSO

Cuando se habla de control interno se debe hacer énfasis en lo planteado por el informe COSO que es un prestigioso referente con respecto al tema a nivel mundial. En él se especifican los principales conceptos relacionados con el control interno, además que se realizaron modificaciones a los conceptos que existían hasta ese entonces dándole una mayor amplitud. Este informe indica que el control interno se estructura en cinco componentes:

- **Entorno de control:** Es todo aquello que hace referencia a la integridad y valores éticos, filosofía de la dirección, comité de auditoría y auditoría interna,

estructura organizacional, delegación de autoridad, prácticas de recursos humanos, etc.

- **Evaluación de riesgos:** La dirección debe conocer sus riesgos (operacionales, financieros y de cumplimiento de legalidad), documentarlos, evaluarlos y establecer estrategias para afrontarlos.
- **Controles de procedimiento:** Son los controles preventivos o detectores, manuales o automáticos que se encuentran a lo largo de cualquier procedimiento en una entidad.
- **Información y comunicación:** Todos los procedimientos establecidos para generar información suficiente y a tiempo y que permita la toma de decisiones, así como establecer los canales adecuados de comunicación.
- **Monitoreo:** Englobaría todas aquellas actividades que permiten obtener evidencia de que todo lo anterior sigue funcionando.(6)

Estos cinco componentes abarcan todas las áreas del control interno en sí pero son bastante generales, existen otras alternativas para realizar un trabajo más específico si se desea controlar un área determinada.

1.4.1 Marcos de trabajo para el control interno de las Tecnologías de Información (TI)

Cada vez más, la alta dirección se está dando cuenta del impacto significativo que la información puede tener en el éxito de una empresa. La dirección espera de las TI contribuyan al éxito del negocio y se pueda obtener una ventaja competitiva de su buen uso. Las Organizaciones necesitan saber si con la información administrada es posible garantizar:

- El logro de sus objetivos.
- La flexibilidad suficiente para aprender y adaptarse.
- El manejo juicioso de los riesgos que enfrenta la Organización.
- El análisis de las oportunidades de negocio y actuar de acuerdo a ellas.

El éxito de la Organización depende en gran medida de que se entienden los riesgos y se aprovechan los beneficios de las TI, para ello se necesita:

- Alinear la estrategia de las TI con la estrategia del negocio.
- Lograr que toda la estrategia de las TI, así como las metas fluyan de forma gradual a toda la empresa.

- Proporcionar estructuras organizativas que faciliten la implementación de las metas del negocio.
- Crear las comunicaciones efectivas entre el negocio y las TI y con los socios externos.
- Medir el desempeño de las TI.

Las empresas no pueden responder de forma efectiva a estos requerimientos de negocio y de gobierno sin adoptar e implementar un marco de Referencia de gobierno y de control para las TI.(7)

1.4.1.1 Marco de trabajo COBIT

COBIT es un marco de trabajo y un conjunto de herramientas de gobierno de tecnologías de la información para buenas prácticas dirigido principalmente al control y supervisión de las TI además que posee una serie de recursos que pueden servir de modelos de referencia para la gestión de las TI.(8)

COBIT define las actividades de TI de una organización, en un modelo genérico de procesos en cuatro dominios. Su marco de trabajo proporciona un modelo de procesos de referencia y un lenguaje común para todos los implicados en los trabajos de la organización, con el fin de que visualicen y administren las actividades de TI. La incorporación de un modelo y un lenguaje común para todas las partes de un negocio involucradas en TI es uno de los pasos iniciales más importantes hacia un buen gobierno. Brinda un marco de trabajo para la medición y monitoreo del desempeño de las Tecnologías de Información e integra las mejores prácticas administrativas. Fomenta la propiedad de los procesos, permitiendo que se definan las responsabilidades. Determina las actividades y los riesgos que requieren ser administrados.

El alcance de los cuatro dominios es:

- **Planear y organizar.** Cubre las estrategias y de la organización, identificando la manera en que las TI pueda contribuir al logro de los objetivos del negocio. La visión estratégica requiere ser planeada, comunicada y administrada desde diferentes perspectivas. Finalmente, se debe implementar una estructura organizacional y una estructura tecnológica apropiada.
- **Adquirir e implementar.** Las soluciones de las TI necesitan ser identificadas, desarrolladas o adquiridas así como la implementación e integración en los

procesos del negocio. Además, el cambio y el mantenimiento de los sistemas existentes son necesarios para garantizar que las soluciones sigan satisfaciendo los objetivos del negocio.

- **Entregar y dar soporte.** Se preocupa de la entrega de los servicios requeridos, la prestación del servicio, la administración de la seguridad y de la continuidad, el soporte del servicio a los usuarios, la administración de los datos y de las instalaciones operacionales.
- **Monitorear y Evaluar.** Este dominio abarca la administración del desempeño, el monitoreo del control interno, el cumplimiento regulatorio y la aplicación del Gobierno.(7)

Este marco de trabajo cuenta con 5 versiones hasta hoy en día, cada una de ellas orientada a un área específica de las TI estas son:

- **Auditoría**
- **Control**
- **Administración**
- **Gobierno de TI**
- **Gobierno Corporativo de TI**

En este trabajo se estará abordando principalmente la tercera versión de este marco de trabajo referida a la administración debido a que es la que más cerca se encuentra de la actualidad cubana hoy en día con respecto a las TI.

1.5 Soluciones Similares

Para obtener un mayor conocimiento sobre los sistemas de gestión de control se realizó un estudio sobre soluciones similares tanto nacionales como internacionales, que pudiesen ser utilizadas para realizar la propuesta de solución. Luego de realizar este estudio se describe las herramientas siguientes:

1.5.1 Meycor COSO AG

El software **Meycor COSO AG** ha sido desarrollado por DATASEC y permite realizar una evaluación del control interno según el informe COSO, evaluación de los riesgos y realizar auditorías de las evaluaciones. (9)

Está compuesto por 3 módulos:

- Meycor COSO AG – Eval

- Meycor COSO AG – Audit
- Meycor COSO AG – Web

Además permite realizar auditorías a las evaluaciones brindadas por el módulo anterior y el tercer módulo permite que los usuarios accedan por la red para contestar cuestionarios sobre el informe COSO y poder ver cómo está su nivel de control interno en sus organizaciones.

1.5.2 COBIT Advisor

Este es un software para automatizar el modelo COBIT y permitirle un modelo consistente para evaluar, auditar la gestión y el control de los sistemas de TI.

- Provee la base de datos del COBIT con 34 procesos, 318 objetivos de control y 34 guías de auditoría.
- Asegura una adecuada planeación de cada trabajo (alcance y cobertura).
- Ofrece facilidades para la supervisión en línea del trabajo y análisis de resultados, mediante el registro, clasificación y filtro (selección) de resúmenes de los procesos, objetivos de control, guías de auditoría y observaciones.(10)

Estos no son las únicas funcionalidades que brinda este software, en general es un software bastante completo, capaz de realizar casi cualquier actividad relacionada con el control interno y las TI.

1.5.3 Meycor COBIT CSA

Meycor COBIT (CSA) no sólo sirve para realizar un diagnóstico de la situación actual y generación de recomendaciones, sino que permite realizar evaluaciones periódicas, las cuales pueden ser comparadas entre sí para medir los avances alcanzados de un período a otro. Incorpora un módulo de auditoría (MEYCOR-AUDIT COBIT), cuyo cometido principal es la verificación de la fiabilidad de las respuestas obtenidas a los cuestionarios. Debido a que todo el diagnóstico y la generación de recomendaciones se basan en dichas respuestas, este módulo cierra el círculo, brindando una herramienta que permite realizar una verificación de la fiabilidad en forma independiente de la evaluación en sí. Además permite evaluar los aspectos de seguridad, calidad, eficacia y eficiencia.(11)

1.5.4 Sistema de gestión de información para el control interno en el Ministerio de Comunicaciones

Este Sistema fue desarrollado en la Universidad de las Ciencias Informáticas (UCI) con el fin de realizar el control interno del Ministerio de Comunicaciones y las entidades subordinadas al mismo. Cumpliendo con las normas del informe COSO, el software genera una guía de autocontrol para que las empresas pertinentes puedan realizar su propia guía de autocontrol satisfactoriamente dado que brinda la posibilidad de agregar parámetros y eliminar los que no sean del interés de la empresa. También cuenta con la opción de mostrar a los administradores un resumen sobre las empresas que han realizado la guía y sus resultados.

Después de analizar estas herramientas se decidió que lo más factible sería continuar desarrollando la aplicación realizada en la Universidad de las Ciencias Informáticas debido a que es la más adecuada a los requerimientos del Ministerio, además que esta herramienta fue realizada para el Ministerio acoplando así todos los aspectos necesarios para su buen funcionamiento. Las primeras tres herramientas son una buena opción debido que brindan muchas posibilidades en el ámbito de los softwares de control interno, pero al ser herramientas privadas no se recomienda el uso de ninguna de ellas, pues sería necesario realizar grandes gastos económicos para adquirir la licencia de alguno de estos.

1.6 Metodologías de desarrollo de software

En estos últimos años se han desarrollado dos corrientes principales en lo referente al desarrollo de software, estas son las llamadas metodologías pesadas y las ágiles. La diferencia fundamental entre estas metodologías reside en que las metodologías pesadas intentan conseguir el objetivo común por medio del orden y la documentación y las ágiles tratan de mejorar la calidad del software por medio de la comunicación directa e inmediata entre las personas que están relacionadas en el proceso.

El desarrollo de software no es una tarea fácil, prueba de ello es que existen numerosas propuestas metodológicas que inciden en distintas dimensiones del proceso de desarrollo. Por una parte se encuentran aquellas propuestas más tradicionales que se centran especialmente en el control del proceso, estableciendo rigurosamente las actividades involucradas, los artefactos que se deben producir y las herramientas y notaciones que se usarán. Estas propuestas han demostrado ser

efectivas y necesarias en un gran número de proyectos, pero también han presentado problemas en otros muchos.(12)

Se muestran en la siguiente tabla, la filosofía de las metodologías ágiles, las cuales dan mayor valor al individuo, a la colaboración con el cliente y al desarrollo incremental del software con iteraciones muy cortas. Este enfoque ha mostrado su efectividad en proyectos con requisitos muy cambiantes y cuando se exige reducir drásticamente los tiempos de desarrollo pero manteniendo una alta calidad.

Tabla 1. Diferencias entre metodologías ágiles y tradicionales.

Metodología Ágil	Metodología Tradicional
Pocos artefactos. El modelado es prescindible, modelos desechables	Más artefactos. El modelado es esencial, mantenimiento de modelos
Pocos roles, más genéricos y flexibles	Más roles, más específicos
No existe un contrato tradicional, debe ser bastante flexible	Existe un contrato prefijado
Cliente es parte del equipo de desarrollo (además in-situ)	El cliente interactúa con el equipo de desarrollo mediante reuniones
Orientada a proyectos pequeños. Corta duración (o entregas frecuentes), equipos pequeños (< 10 integrantes) y trabajando en el mismo sitio	Aplicables a proyectos de cualquier tamaño, pero suelen ser especialmente efectivas/usadas en proyectos grandes y con equipos posiblemente dispersos
La arquitectura se va definiendo y mejorando a lo largo del proyecto	Se promueve que la arquitectura se defina tempranamente en el proyecto
Énfasis en los aspectos humanos: el individuo y el trabajo en equipo	Énfasis en la definición del proceso: roles, actividades y artefactos
Basadas en heurísticas provenientes de prácticas de producción de código	Basadas en normas provenientes de estándares seguidos por el entorno de desarrollo
Se esperan cambios durante el proyecto	Se espera que no ocurran cambios de gran impacto durante el proyecto

Después de analizar lo descrito en la tabla anterior se decidió utilizar una metodología ágil debido a las ventajas que brinda para equipos de desarrollo pequeños que no requieran de un gran flujo de documentación. Además que es bastante flexible a cambios en el software mientras aún se trabaja en el mismo a diferencia de las

pesadas o tradicionales como son mayormente conocidas. De estas metodologías ágiles se estudian a continuación Scrum y XP (eXtreming Programming).

1.6.1 Scrum

SCRUM se ha destacado como la metodología ágil más utilizada en el mundo con una gran aceptación, en la que se definen claramente unos roles, artefactos y reuniones específicas que miden de manera constante el desarrollo del proyecto y beneficia la comunicación del equipo.(13)

Desarrollada por Ken Schwaber, Jeff Sutherland y Mike Beedle. Define un marco para la gestión de proyectos, que se ha utilizado con éxito durante los últimos años. Está especialmente indicada para proyectos con un rápido cambio de requisitos. Sus principales características se pueden resumir en dos. El desarrollo de software se realiza mediante iteraciones, denominadas sprints, con una duración de 30 días. El resultado de cada sprint es un incremento ejecutable que se muestra al cliente. La segunda característica importante son las reuniones a lo largo proyecto, entre ellas destaca la reunión diaria de 15 minutos del equipo de desarrollo para coordinación e integración.(12)

1.6.2 Programación Extrema (XP)

Según Jose Carlos Cortizo, Diego Gil y Miguel Ruiz, XP es una *“metodología ligera o ágil para el desarrollo de software eficiente y altamente efectivo”*(14)

La gran controversia en la definición de la programación extrema viene de la mano de la palabra metodología. Muchos defensores a ultranza de la XP consideran que no se puede denominar metodología porque elimina toda la parte burocrática anteriormente asociada a las metodologías para el desarrollo de software (continua documentación, diseños varios y papeleos excesivos). Por otro lado, obviar que proporciona una guía o un proceso repetitivo para el desarrollo de software sería obviar lo que en sí es. Así pues, se puede tomar como bueno el consenso de denominarla como metodología ligera (o ágil), debido a que elimina la “ pesada ” carga del papeleo a los desarrolladores.(14)

Como metodología, la Programación Extrema, presenta muchos puntos comunes con el desarrollo incremental, comenzando por el hecho de que el software desarrollado con XP se realiza de forma incremental. Para ver todos los puntos en que se centra la metodología XP, se dividirá por fases:

Fase I: Exploración

En esta fase, los clientes plantean a grandes rasgos las historias de usuario que son de interés para la primera entrega del producto. Al mismo tiempo el equipo de desarrollo se familiariza con las herramientas, tecnologías y prácticas que se utilizarán en el proyecto. Se prueba la tecnología y se exploran las posibilidades de la arquitectura del sistema construyendo un prototipo. La fase de exploración toma de pocas semanas a pocos meses, dependiendo del tamaño y familiaridad que tengan los programadores con la tecnología.

Fase II: Planificación de la Entrega

En esta fase el cliente establece la prioridad de cada historia de usuario y correspondientemente, los programadores realizan una estimación del esfuerzo necesario de cada una de ellas. Se toman acuerdos sobre el contenido de la primera entrega y se determina un cronograma en conjunto con el cliente. Una entrega debería obtenerse en no más de tres meses. Esta fase dura unos pocos días.

Las estimaciones de esfuerzo asociado a la implementación de las historias la establecen los programadores utilizando como medida el punto. Un punto, equivale a una semana ideal de programación. Las historias generalmente valen de 1 a 3 puntos. Por otra parte, el equipo de desarrollo mantiene un registro de la "velocidad" de desarrollo, establecida en puntos por iteración, basándose principalmente en la suma de puntos correspondientes a las historias de usuario que fueron terminadas en la última iteración.

La planificación se puede realizar basándose en el tiempo o el alcance. La velocidad del proyecto es utilizada para establecer cuántas historias se pueden implementar antes de una fecha determinada o cuánto tiempo tomará implementar un conjunto de historias. Al planificar por tiempo, se multiplica el número de iteraciones por la velocidad del proyecto, determinándose cuántos puntos se pueden completar. Al planificar según alcance del sistema, se divide la suma de puntos de las historias de usuario seleccionadas entre la velocidad del proyecto, obteniendo el número de iteraciones necesarias para su implementación.

Fase III: Iteraciones

Esta fase incluye varias iteraciones sobre el sistema antes de ser entregado. El Plan de Entrega está compuesto por iteraciones de no más de tres semanas. En la primera

iteración se puede intentar establecer una arquitectura del sistema que pueda ser utilizada durante el resto del proyecto. Esto se logra escogiendo las historias que fuercen la creación de esta arquitectura, sin embargo, esto no siempre es posible ya que es el cliente quien decide qué historias se implementarán en cada iteración (para maximizar el valor de negocio). Al final de la última iteración el sistema estará listo para entrar en producción.

Los elementos que deben tomarse en cuenta durante la elaboración del Plan de la Iteración son: historias de usuario no abordadas, velocidad del proyecto, pruebas de aceptación no superadas en la iteración anterior y tareas no terminadas en la iteración anterior. Todo el trabajo de la iteración es expresado en tareas de programación, cada una de ellas es asignada a un programador como responsable, pero llevadas a cabo por parejas de programadores.

Fase IV: Producción

La fase de producción requiere de pruebas adicionales y revisiones de rendimiento antes de que el sistema sea trasladado al entorno del cliente. Al mismo tiempo, se deben tomar decisiones sobre la inclusión de nuevas características a la versión actual, debido a cambios durante esta fase.

Es posible que se rebaje el tiempo que toma cada iteración, de tres a una semana. Las ideas que han sido propuestas y las sugerencias son documentadas para su posterior implementación (por ejemplo, durante la fase de mantenimiento).

Fase V: Mantenimiento

Mientras la primera versión se encuentra en producción, el proyecto XP debe mantener el sistema en funcionamiento al mismo tiempo que desarrolla nuevas iteraciones. Para realizar esto se requiere de tareas de soporte para el cliente. De esta forma, la velocidad de desarrollo puede bajar después de la puesta del sistema en producción. La fase de mantenimiento puede requerir nuevo personal dentro del equipo y cambios en su estructura.

Fase VI: Muerte del Proyecto

Es cuando el cliente no tiene más historias para ser incluidas en el sistema. Esto requiere que se satisfagan las necesidades del cliente en otros aspectos como rendimiento y confiabilidad del sistema. Se genera la documentación final del sistema y no se realizan más cambios en la arquitectura. La muerte del proyecto también ocurre

cuando el sistema no genera los beneficios esperados por el cliente o cuando no hay presupuesto para mantenerlo. (15)

La XP proporciona una serie de ventajas a la hora de afrontar ciertos proyectos, que deben ser tenidas muy en cuenta , porque el uso de esta metodología en ciertos contextos puede ahorrar mucho tiempo y muchos recursos.(14)

Después de ver una pequeña reseña de las principales metodologías ágiles utilizadas hoy en día, se decidió utilizar XP debido a que es una metodología adecuada para equipos de trabajo pequeños y que a su vez es capaz de eliminar toda la gran carga que supone la documentación para los programadores. Además que proporciona una serie de ventajas con respecto a otras metodologías como:

- Programación organizada.
- Menor tasa de errores.
- Satisfacción del programador.
- Solución de errores de programas.
- Implementa una forma de trabajo fácilmente adaptable a las circunstancias.(16)

También se tuvo en cuenta que la versión anterior del software fue desarrollada teniendo en cuenta esta metodología.

1.7 Marcos de trabajo

Un marco de trabajo o framework simplifica el desarrollo de una aplicación mediante la automatización de algunos de los patrones utilizados para resolver las tareas comunes. Además proporciona estructura al código fuente, forzando al desarrollador a crear código más legible y más fácil de mantener. Por último, un marco de trabajo facilita la programación de aplicaciones, ya que encapsula operaciones complejas en instrucciones sencillas(17).

1.7.1 CodeIgniter

CodeIgniter es un conjunto de herramientas para personas que construyen su aplicación web usando PHP. El objetivo de este marco de trabajo es permitirles a los usuarios desarrollar aplicaciones con mayor rapidez. Esto es posible porque provee una amplia gama de componentes reutilizables, evitando que se tengan que re implementar componentes comunes en todas las aplicaciones de gestión. Se caracteriza por su sencillez. Es justamente restrictivo sobre qué caracteres utilizar en las cadenas URL para ayudar a minimizar la posibilidad de ataque utilizando esta vía.

La arquitectura de seguridad que este marco presenta es muy sencilla, solo implementa un conjunto de restricciones, validaciones y filtros que no garantizan la seguridad en un entorno que requiera de una seguridad estricta (18).

Luego de realizar un estudio del marco de trabajo se decidió utilizar el mismo debido a que la versión anterior de la aplicación había sido desarrollada en este marco de trabajo, además que este brinda ciertas facilidades dada su ligereza y fácil utilización debido a que este marco de trabajo brinda un conjunto de bibliotecas para tareas comunes ahorrando tiempo de trabajo a los desarrolladores a diferencia de si se realizase el proyecto desde cero sin utilizar ningún marco de trabajo.

1.8 Lenguajes de programación

Según la definición teórica, como lenguaje se entiende a un sistema de comunicación que posee una determinada estructura, contenido y uso. La programación es, en el vocabulario propio de la informática, el procedimiento de escritura del código fuente de un software. De esta manera, puede decirse que la programación le indica al programa informático qué acción tiene que llevar a cabo y cuál es el modo de concretarla.

Con estas nociones en claro, podemos afirmar que un lenguaje de programación es aquella estructura que, con una cierta base sintáctica y semántica, imparte distintas instrucciones a un programa de computadora(19).

1.8.1 Lenguajes de programación del lado del cliente

1.8.1.1 CSS

CSS es un lenguaje de estilo que define la presentación de los documentos HTML. Por ejemplo, CSS abarca cuestiones relativas a fuentes, colores, márgenes, líneas, altura, anchura, imágenes de fondo, posicionamiento avanzado y muchos otros temas. CSS ofrece numerosas opciones y es más preciso y sofisticado. CSS está soportado por todos los navegadores hoy día(20).

Se utilizará CSS en su versión 3 para trabajar los estilos de las páginas dado que este ofrece un mayor control sobre los estilos de las páginas, además que es bastante sencillo de utilizar.

1.8.1.2 JavaScript

JavaScript es un lenguaje de programación que se utiliza principalmente para crear páginas web dinámicas.

Una página web dinámica es aquella que incorpora efectos como texto que aparece y desaparece, animaciones, acciones que se activan al pulsar botones y ventanas con mensajes de aviso al usuario.

Técnicamente, JavaScript es un lenguaje de programación interpretado, por lo que no es necesario compilar los programas para ejecutarlos. En otras palabras, los programas escritos con JavaScript se pueden probar directamente en cualquier navegador sin necesidad de procesos intermedios(21).

1.8.1.3 HTML

HTML (por sus siglas en inglés HyperText Markup Language) es un lenguaje de programación que se utiliza para el desarrollo de páginas de Internet.

Se trata de un formato abierto que surgió a partir de las etiquetas SGML (Standard Generalized Markup Language). Concepto traducido generalmente como “Estándar de Lenguaje de Marcado Generalizado” y que se entiende como un sistema que permite ordenar y etiquetar diversos documentos dentro de una lista. Este lenguaje es el que se utiliza para especificar los nombres de las etiquetas que se utilizarán al ordenar, no existen reglas para dicha organización, por eso se dice que es un sistema de formato abierto(22).

1.8.2 Lenguajes de programación del lado del servidor

1.8.2.1 PHP

PHP, acrónimo de "PHP: Hypertext Preprocessor", es un lenguaje de 'scripting' de propósito general y de código abierto que está especialmente pensado para el desarrollo web y que puede ser embebido en páginas HTML. Su sintaxis recurre a C, Java y Perl, siendo así sencillo de aprender. La meta principal de este lenguaje es permitir a los desarrolladores web escribir dinámica y rápidamente páginas web generadas; aunque se puede hacer mucho más con PHP(23).

Las cuatro grandes características de PHP son:

- **Velocidad**
- **Estabilidad**
- **Seguridad**
- **Simplicidad**

Buena otra característica a tener en cuenta sería la conectividad. PHP dispone de una amplia gama de librerías y agregarle extensiones es muy fácil. Esto le permite al PHP ser utilizado en muchas áreas diferentes, tales como encriptado, gráficos, XML y otras. Por eso se decidió el uso de PHP en su versión 5.5.11, además de que para una aplicación web es lo más recomendado.

1.9 Servidor Web

Un servidor web o servidor HTTP es un programa informático que procesa una aplicación del lado del servidor, realizando conexiones bidireccionales y/o unidireccionales y síncronas o asíncronas con el cliente y generando o cediendo una respuesta en cualquier lenguaje o aplicación del lado del cliente. El código recibido por el cliente suele ser compilado y ejecutado por un navegador web.

1.9.1 Apache

Apache es software libre y uno de los más utilizados hoy en día. Es un proyecto de la Fundación de Software Apache, con el objetivo de suministrar un servidor seguro, eficiente, y extensible que proporcione servicios HTTP en correspondencia con los estándares HTTP actuales.

Principales características:

- Multiplataforma.
- Modular: Puede ser adaptado a diferentes entornos y necesidades, con los diferentes módulos de apoyo que proporciona y con la API de programación de módulos, para el desarrollo de módulos específicos.
- Extensible: gracias a ser modular se han desarrollado diversas extensiones entre las que destaca PHP, un lenguaje de programación del lado del servidor.(24)

Se utilizará Apache en su versión 2.4.4 debido a que es un servidor web de código libre además es fácil el trabajo con el mismo y su instalación es bastante sencilla.

1.10 Herramientas CASE

CASE es un acrónimo para Computer-Aided Software Engineering, aunque existen algunas variaciones para lo que actualmente se entiende por CASE. Esencialmente, un CASE es una herramienta que ayuda al ingeniero de software a desarrollar y mantener software.(25)

1.10.1 Visual Paradigm

Visual Paradigm para UML es una herramienta profesional que soporta el ciclo de vida completo del desarrollo de software: análisis y diseño orientados a objetos, construcción, pruebas y despliegue. Permite dibujar todos los tipos de diagramas de clases, generar código desde diagramas y generar documentación. Algunas características de esta herramienta son(26):

- Posibilita vincular al mismo con la base de datos y el IDE de desarrollo.
- Posee distribución automática de diagramas - Reorganización de las figuras y conectores de los diagramas UML.
- Permite generar código php y realizar ingeniería directa e inversa.

1.11 Sistemas gestores de base de datos

Según Teresa Garzón una base de datos es *“un fondo común de información interrelacionada para ser accedida mediante consultas. También se puede definir como un sistema computarizado para mantener información de un individuo o de una organización y hacer que esté disponible cuando se solicite”* (27).

Características de una base de datos:

- Integridad de los datos.
- No redundancia de datos.
- Restricciones de seguridad y confidencialidad.
- Múltiples visitas de los datos.
- Protección para fallos.
- Interfaz de alto nivel.

Si una base de datos es un sistema computarizado para guardar información, un gestor de base de datos es un sistema para controlar y gestionar la base de datos, Teresa Garzón plantea que *“es una colección de programas que permiten a los usuarios crear y mantener una base de datos. Software de propósito general que facilita los procesos de definición, construcción y manipulación de la base de datos para distintas aplicaciones”*(27).

Las base de datos se pueden mantener manualmente pero este proceso tendría gran costo de tiempo y esfuerzo, sin embargo utilizando un sistema gestor de base de datos

se ahorrarían todas estas molestias y además sería más fácil asegurar todas las características de una base de datos mencionadas anteriormente.

1.11.1 PostgreSQL

PostgreSQL es un Sistema Gestor de Base de Datos objeto-relacional basado en el proyecto POSTGRES creado por el Departamento de Investigación de la Universidad de California en Berkeley. Aunque la licencia es propiedad de dicha universidad, está distribuido bajo la licencia BSD y con su código fuente disponible libremente; es decir, es libre para utilizar, copiar, modificar y distribuir, sin importar para los fines que se utilice. PostgreSQL utiliza como modelo la arquitectura cliente/servidor, además de utilizar multiprocesos en vez de multihilos, este propicia que un fallo en uno de los procesos del sistema no afecte el resto del mismo y continúe funcionando; presenta confiabilidad, estabilidad y mantenimiento de la integridad de los datos; permite definir un nuevo tipo de tabla, a partir de otra previamente definida, todo esto gracias a que posee características de la Programación Orientada a Objetos (POO), como puede ser la herencia, tipos de datos, funciones, restricciones y disparadores, reglas e integridad transaccional(28).

Se decidió utilizar este sistema gestor de base de datos en su versión 9.2.4 porque el mismo es software libre, además de la experiencia de los desarrolladores en el uso de esta herramienta. También se decidió el uso de este no solo por las ventajas y las facilidades que brinda sino porque el software en su versión anterior estaba desarrollado en PostgreSQL y no sería práctico realizar estos cambios.

1.12 Entorno integrado de desarrollo

Un entorno de desarrollo integrado, llamado también IDE (siglas en inglés de integrated development environment), es un programa informático compuesto por un conjunto de herramientas de programación. Un IDE es un entorno de programación que ha sido empaquetado como un programa de aplicación; es decir, consiste en un editor de código, un compilador, un depurador y un constructor de interfaz gráfica (GUI).

- **Editor de código fuente:** Editor de texto que sirve para editar el código fuente de aplicaciones informáticas.
- **Un compilador:** Es un traductor de código fuente, lo traduce a un lenguaje que sea legible para las máquinas.

- **Un depurador:** Es una aplicación que tiene como función probar y eliminar posibles errores en un programa en desarrollo.
- **Constructor de interfaz gráfica:** Herramienta que sirve para crear y diseñar las interfaces con las cuales habrá interacción entre la aplicación y el usuario.

1.12.1 NetBeans

NetBeans es un entorno de desarrollo muy completo y profesional. Contiene muchas funcionalidades, para distintos tipos de aplicaciones y para facilitar al máximo la programación, la prueba y la depuración de las aplicaciones que se desarrollan. También incorpora un editor propio(29).

NetBeans una herramienta para que los programadores puedan escribir, compilar, depurar y ejecutar programas. Está escrito en Java pero puede servir para casi cualquier otro lenguaje de programación. Existe además un número importante de módulos para extender. NetBeans es un producto libre y gratuito sin restricciones de uso(30).

Debido a que es software libre y es capaz de soportar PHP el lenguaje de programación que se utilizará para el desarrollo de la aplicación se decidió utilizar el mismo en su versión 7.4. Además es el IDE con el que mejor relacionado está el equipo de desarrollo.

1.13 Conclusiones parciales del capítulo

El estudio realizado permitió familiarizarse más con los conceptos relacionados al control interno. Además al analizar otras soluciones similares se recopilaron de las mismas ideas, para la posterior implementación en la propuesta de solución. Para el desarrollo de estas se decidió utilizar PHP 5.5.11 como lenguaje de programación, CodeIgniter 2.1.4 como marco de trabajo, el IDE NetBeans 7.4, de servidor web Apache 2.2.4, XP como metodología de desarrollo y el sistema gestor de BD PostgreSQL 9.2.4 debido a las características mencionadas anteriormente.

CAPITULO 2: PROPUESTA DE SOLUCIÓN.

Introducción

El control interno en cualquier empresa es un factor clave, es por esto que muchas empresas utilizan software de control para un buen funcionamiento dentro de estas. En el presente capítulo se propone una solución al problema de la gestión del control interno del ministerio de la comunicaciones, pues se describen los aspectos funcionales y no funcionales, las historias de usuarios (HU), plan de iteraciones, plan de entrega y tarjetas CRC (Clases, Responsabilidad y Colaboración).

2.1 Usuarios del Sistema

Los usuarios del sistema son aquellas personas o sistemas que interactúan con la aplicación y obtienen un resultado a partir de los procesos que se ejecutan. En la presente investigación se definen los siguientes roles:

Entidad: es el usuario más básico de la aplicación solo tiene acceso a determinadas funcionalidades del sistema, como subir evidencia al sistema con relación a algún aspecto o crear reportes, además de dar respuesta a la guía de autocontrol.

Administrador: usuario con mayores privilegios en el sistema. Tiene acceso a funcionalidades críticas en la aplicación, incluyendo la gestión de usuarios y aquellas que determinan el funcionamiento del sistema.

2.2 Funcionalidades del sistema

Antes de establecer una solución al problema es necesario tener un amplio conocimiento de las necesidades del cliente, para esto es necesario realizar una recopilación de los requerimientos que se desea posea el sistema. Existen varias formas de capturar estos requerimientos, por ejemplo mediante entrevistas con el cliente, observando los procesos que se deben informatizar.

Los requerimientos del sistema son:

- Condición o capacidad requerida por el usuario para resolver un problema o alcanzar un objetivo.
- Condición o capacidad que debe satisfacer o poseer un sistema o una componente de un sistema para satisfacer un contrato, un estándar, una especificación u otro documento formalmente impuesto.
- Representación documentada de una condición o capacidad examinadas

anteriormente.(31)

2.2.1 Lista de reserva del producto.

Los requisitos funcionales describen lo que el sistema debe hacer y dependen del tipo de software a desarrollarse. A continuación se listan los principales requerimientos funcionales que debe tener el sistema.

1. Permitir que un componente solo pueda ser evaluado por un usuario.
2. Validar que al agregar un aspecto, los subcomponentes sean los correspondientes al componente seleccionado.
3. Permitir a los administradores consultar datos estadísticos en forma de gráfica.
4. Mostrar automáticamente el control para subir evidencia, si el aspecto es marcado de no.
5. Mostrar una evaluación de la guía después de ser llenada por el usuario.
6. Asignar peso a cada aspecto de la guía de autocontrol.
7. Crear guías nuevas a partir de aspectos de guías existentes o a partir de una completamente desde cero.
8. Determinar el porcentaje de cumplimiento de la guía.
9. Mostrar a los administradores un resumen de los resultados de la aplicación de la guía de autocontrol en las entidades subordinadas al MINCOM.
10. Exportar el resultado de la guía después de terminada esta.
11. Incorporar preguntas referentes a la gestión de las tecnologías de la información basados en el marco de trabajo COBIT como aspectos de la guía de autocontrol.

2.2.2 Características del sistema.

Los requerimientos no funcionales se refieren a las propiedades emergentes del sistema como la fiabilidad, el tiempo de respuesta, entre otros. A veces también se les conoce como restricciones del sistema dado que pueden restringir el proceso a utilizar para desarrollar el sistema. Los principales requisitos no funcionales que presenta este sistema son:

Funcionalidad: El software es capaz de proporcionar un conjunto de funciones para realizar tareas dependiendo del usuario que esté interactuando con el mismo y brinda los resultados esperados por el usuario.

Seguridad: El acceso a la información está dado dependiendo del rol de los usuarios y para proteger la información solo los usuarios con cierto nivel de permisos pueden interactuar con la misma.

Eficiencia: El software debe ser capaz de dar respuestas en menos de 10 segundos luego de ser realizada alguna petición por parte del usuario.

Software: Por parte del cliente solo necesitará un navegador web y del lado del servidor el gestor de base de datos PostgreSQL y un servidor web, Apache.

Hardware: Para el correcto funcionamiento del sistema es necesario que la PC servidor posea al menos 1GB de memoria RAM, microprocesador Pentium4 o superior y capacidad de 1GB libre en el disco duro.

2.3 Fase de exploración

En esta etapa es donde el equipo de desarrollo se familiariza con las herramientas, tecnologías y prácticas que se utilizarán en el proyecto. Se prueba la tecnología y se exploran las posibilidades de la arquitectura del sistema construyendo un prototipo.

2.3.1 Historias de usuario (HU)

En la programación extrema, todos los requerimientos se expresan como escenarios (llamados *historias de usuario*), los cuales se implementan directamente como una serie de tareas(32).

Por lo tanto se definieron 11 historias de usuario de las cuales 6 de ellas representan nuevas funcionalidades, 4 de ellas son mejoras o corrección de posibles fallos en la versión anterior del software, la historia de usuario restante responde a la investigación relacionada con el marco de trabajo COBIT y su incorporación al software. A continuación se muestran las tablas de las 2 primeras HU, la información de las HU restantes se encuentran en el [anexo 2](#).

Tabla 2.HU Permitir que un componente solo pueda ser evaluado por un usuario.

Historia de Usuario	
No.: 1	Nombre: Permitir que un componente solo pueda ser evaluado por un usuario.
Usuario: Administrador.	

Prioridad en el Negocio: Medio.	Nivel de Complejidad: Medio.
Estimación: 1	Iteración Asignada: 1
Descripción: La empresa que esté realizando la guía de autocontrol si tiene más de un usuario, el administrador deberá distribuir los componentes a evaluar entre estos usuarios.	
Información adicional (Observaciones): Esto brinda la posibilidad a la empresa de no evaluar componentes dos veces.	

Tabla 3.HU Validar que al agregar un aspecto, los subcomponentes sean los correspondientes al componente seleccionado

Historia de Usuario	
No.: 2	Nombre: Validar que al agregar un aspecto, los subcomponentes sean los correspondientes al componente seleccionado.
Usuario: Administrador.	
Prioridad en el Negocio: Medio.	Nivel de Complejidad: Bajo.
Estimación: 0.5	Iteración Asignada: 1
Descripción: El administrador tiene la posibilidad de, al agregar un aspecto, a la hora de elegir el componente deseado, solo se le muestren los subcomponentes correspondientes a dicho componente.	
Información adicional (Observaciones):	

2.3.2 Estimación de esfuerzo por HU

Las estimaciones de esfuerzo asociado a la implementación de las historias de usuario la establecen los programadores utilizando como medida el punto. Un punto, equivale a una semana ideal de programación. Las historias generalmente valen de 1 a 3 puntos.

Tabla 4. Estimación de esfuerzo por HU.

No	Historia de Usuario	Estimación (Semanas)
1	Permitir que un componente solo pueda ser evaluado por un usuario.	1

2	Validar que al agregar un aspecto, los subcomponentes sean los correspondientes al componente seleccionado.	0.5
3	Permitir a los administradores consultar datos estadísticos en forma de tabla.	1
4	Mostrar automáticamente el control para subir evidencia, si el aspecto es marcado de no.	1
5	Mostrar una evaluación de la guía después de ser llenada por el usuario.	1
6	Asignar peso a cada aspecto de la guía de autocontrol.	0.5
7	Crear guías nuevas: A partir de aspectos de guías existentes o a partir de una completamente desde cero.	1.5
8	Determinar el porcentaje de cumplimiento de la guía.	1
9	Mostrar a los administradores un resumen de los resultados de la aplicación de la guía de autocontrol en las entidades subordinadas al MINCOM.	1
10	Exportar el resultado de la guía después de terminada esta.	1
11	Incorporar preguntas referentes a la gestión de información basados en el marco de trabajo COBIT como aspectos de la guía de autocontrol.	1.5

2.4 Planificación

En esta fase el cliente establece la prioridad de cada historia de usuario y correspondientemente, los programadores realizan una estimación del esfuerzo necesario de cada una de ellas. Se toman acuerdos sobre el contenido de la primera entrega y se determina un cronograma en conjunto con el cliente.

2.4.1 Plan de duración de las Iteraciones

Tabla 5. Plan de Iteraciones.

Iteraciones	Orden de las Historias de Usuario a implementar	Cantidad de tiempo de trabajo (semanas)
Iteración 1	<ol style="list-style-type: none"> 1. Permitir que un componente solo pueda ser evaluado por un usuario. 2. Validar que al agregar un aspecto, los subcomponentes sean los correspondientes al componente seleccionado. 3. Permitir a los administradores consultar datos estadísticos en forma de tabla. 4. Mostrar automáticamente el control para subir evidencia, si el aspecto es marcado de no. 	3.5
Iteración 2	<ol style="list-style-type: none"> 5. Mostrar una evaluación de la guía después de ser llenada por el usuario. 6. Asignar peso a cada aspecto de la guía de autocontrol. 7. Crear guías nuevas a partir de aspectos de guías existentes o a partir de una completamente desde cero. 8. Determinar el porcentaje de cumplimiento de la guía. 9. Mostrar a los administradores un resumen de los resultados de la aplicación de la guía de autocontrol en las entidades subordinadas al MINCOM. 10. Exportar el resultado de la guía después de terminada esta 11. Incorporar preguntas referentes a la gestión de información basados en el marco de trabajo COBIT como aspectos de la guía de autocontrol. 	7.5

2.5 Iteraciones

Esta fase incluye varias iteraciones sobre el sistema antes de ser entregado. El Plan de Entrega está compuesto por iteraciones de no más de tres semanas. En la primera iteración se puede intentar establecer una arquitectura del sistema que pueda ser

Para el desarrollo de la presente investigación la primera iteración fue conformada según las funcionalidades que la aplicación ya tenía y había que optimizar, la segunda se conformó con las funcionalidades restantes.

2.5.1 Plan de entrega

Tabla 6. Plan de entrega.

Historias de Usuarios	Iteración 1	Iteración 2	Iteración 3
Permitir que un componente solo pueda ser evaluado por un usuario.	versión 1.0	Finalizado	Finalizado
Validar que al agregar un aspecto, los subcomponentes sean los correspondientes al componente seleccionado.	versión 1.0	Finalizado	Finalizado
Permitir a los administradores consultar datos estadísticos en forma de tabla.	versión 1.0	Finalizado	Finalizado
Mostrar automáticamente el control para subir evidencia, si el aspecto es marcado de no.	versión 1.0	Finalizado	Finalizado
Mostrar una evaluación de la guía después de ser llenada por el usuario.	-	versión 1.0	Finalizado
Asignar peso a cada aspecto de la guía de autocontrol.	-	versión 1.0	Finalizado
Crear guías nuevas a partir de aspectos de guías existentes o a partir de una completamente desde cero.	-	versión 1.0	Finalizado
Determinar el porcentaje de cumplimiento de la guía.	-	versión 1.0	Finalizado
Mostrar a los administradores un resumen de los resultados de la aplicación de la guía de autocontrol en	-	versión 1.0	Finalizado

las entidades subordinadas al MINCOM.			
Exportar el resultado de la guía después de terminada esta.	-	versión 1.0	Finalizado
Incorporar preguntas referentes a la gestión de información basados en el marco de trabajo COBIT como aspectos de la guía de autocontrol.	-	versión 1.0	Finalizado

2.6 Diseño

En la etapa de diseño es donde son evaluadas las historias de usuario por el equipo del proyecto para dividir las en tareas, cada tarea representa una característica distinta del sistema, estas tareas se representan por medio de las tarjetas CRC.(33)

2.6.1 Descripción de las tarjetas CRC

Las tarjetas CRC identifican y organizan las clases bajo el paradigma orientado a objetos (lo que incluye asignación de responsabilidades), cada tarjeta contiene el nombre de la clase (que representa una o más historias de usuario), una descripción de las responsabilidades o métodos asociados con la clase, así como la lista de las clases con que se relaciona o que colaboran con ella.(33)

En las tablas 7 y 8 se muestra la descripción de las tarjetas CRC correspondientes a las clases en las que se realizaron cambios a su relación con otras clases. Se pueden observar en el [anexo 3](#) las tarjetas CRC de otras clases que sufrieron cambios.

Tabla 7. Tarjeta CRC principal_admin.

Clase: principal_admin	
Responsabilidad	Colaboración
<ul style="list-style-type: none"> Es la clase encargada de ejecutar todas las operaciones que conciernen al administrador. 	recursos layout_admin load input

	editar_guia_v listar_guia seguridad_helper resumun_root ministerio_model
--	--

Tabla 8. Tarjeta CRC aspecto_controler.

Clase: aspecto_controler	
Responsabilidad	Colaboración
<ul style="list-style-type: none"> • Ejecuta las acciones del usuario relacionadas con los aspectos • Permite crear un aspecto • Permite eliminar un aspecto. 	recursos layout_admin load input listar_aspecto crear_aspecto_form seguridad_helper ministerio_model

2.7 Conclusiones parciales del capítulo

En este capítulo se vieron las principales funcionalidades del software y partiendo de las mismas se crearon las historias de usuarios pertinentes, las que facilitarán tener una mayor comprensión de la propuesta de solución. Se transitó por varias fases de desarrollo de software utilizando la metodología XP dando una mayor organización y al equipo de desarrollo permitiendo diseñar un plan de entrega con las iteraciones que se realizarán para desarrollar satisfactoriamente la aplicación. También se modelaron artefactos en la fase de diseño que serán de gran ayuda para las fases que se verán más adelante que son la de implementación y mantenimiento. La descripción de la propuesta de solución realizada en este capítulo será de gran importancia para la implementación de los modelos identificados en el mismo.

CAPÍTULO 3 IMPLEMENTACIÓN Y PRUEBAS

3.1 Introducción

Basados en lo que propone la metodología XP sobre la implementación en parejas se implementarán las historias de usuarios definidas en el capítulo anterior para dar cumplimiento a los requisitos funcionales del sistema. XP no propone los artefactos a utilizar en la implementación, simplemente deja en manos de los desarrolladores la elección de los mismos, para que estos tengan en cuenta cuáles son las necesidades del negocio y qué artefactos son los más indicados para darle cumplimiento a las mismas, además de permitirle utilizar las herramientas con las que más familiarizados estén para simplificar el trabajo y hacer más fácil su desarrollo.

3.2 Descripción de la arquitectura

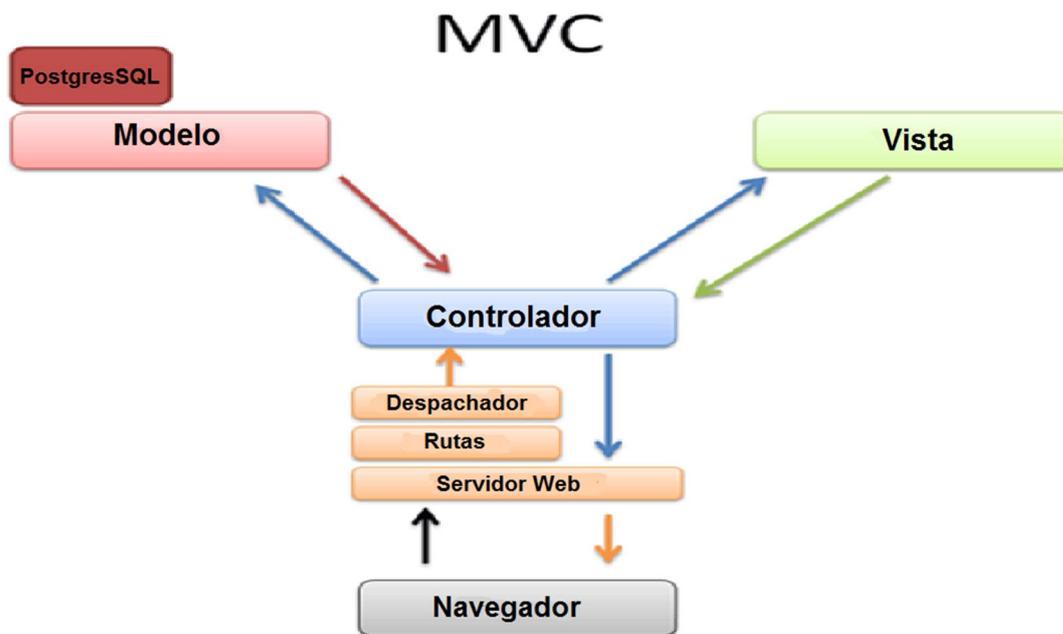


Figura 1 Representación del estilo MVC

El patrón Modelo-Vista-Controlador (MVC) surge con el objetivo de reducir el esfuerzo de programación, necesario en la implementación de sistemas múltiples y sincronizados de los mismos datos, a partir de estandarizar el diseño de las aplicaciones. El patrón MVC es un paradigma que divide las partes que conforman una aplicación en el Modelo, las Vistas y los Controladores, permitiendo la implementación por separado de cada elemento, garantizando así la actualización y mantenimiento del software de forma sencilla y en un reducido espacio de tiempo.(34)

El **Modelo** es el objeto que representa los datos del programa. Maneja los datos y controla todas sus transformaciones. El Modelo no tiene conocimiento específico de los Controladores o de las Vistas, ni siquiera contiene referencias a ellos. Es el propio sistema el que tiene encomendada la responsabilidad de mantener enlaces entre el Modelo y sus Vistas y notificar a las Vistas cuando cambia el Modelo.

La **Vista** es el objeto que maneja la presentación visual de los datos representados por el Modelo. Genera una representación visual del Modelo y muestra los datos al usuario. Interactúa preferentemente con el Controlador, pero es posible que trate directamente con el Modelo a través de una referencia al propio Modelo.

El **Controlador** es el objeto que proporciona significado a las órdenes del usuario, actuando sobre los datos representados por el Modelo, centra toda la interacción entre la Vista y el Modelo. Cuando se realiza algún cambio, entra en acción, bien sea por cambios en la información del Modelo o por alteraciones de la Vista. Interactúa con el Modelo a través de una referencia al propio Modelo.(34)

Luego de ver en que consiste la arquitectura modelo-vista-controlador es válido mencionar que la aplicación anterior fue desarrollada utilizando esta arquitectura, por lo que se decidió mantener el desarrollo con el mismo patrón de arquitectura.

3.3 Patrones de Diseño

Un patrón de diseño es una abstracción de una solución en un nivel alto. Los patrones solucionan problemas que existen en muchos niveles de abstracción. Hay patrones que abarcan las distintas etapas del desarrollo.(35)

3.3.1 Patrones GRASP

Lo esencial de un diseño de objetos lo constituye el diseño de las interacciones de objetos y la asignación de responsabilidades. Las decisiones que se tomen pueden influir profundamente en la extensibilidad, claridad y mantenimiento del sistema de software de objetos, además en el grado y calidad de los componentes reutilizables.

- **Alta cohesión:** Es un principio evaluativo que aplica un diseñador mientras evalúa todas las decisiones de diseño. Indica la relación que existe entre los elementos de un mismo módulo. Es la medida de la relación funcional de los elementos de un módulo. El objetivo es organizar estos elementos de manera que los que tengan una mayor relación a la hora de realizar una tarea pertenezcan al mismo módulo y los elementos no relacionados, se encuentren en módulos separados.

- **Bajo acoplamiento:** Impulsa la asignación de responsabilidades de manera que su localización no incremente el acoplamiento hasta un nivel que lleve a los resultados negativos que puede producir un acoplamiento alto. Es el grado de interdependencia entre los módulos. Un buen diseño se caracteriza por un acoplamiento mínimo, es decir, unos módulos tan independientes los unos de los otros como sea posible.
- **Controlador:** Proporciona guías acerca de las opciones generalmente aceptadas y adecuadas para manejar eventos. Es conveniente utilizar la misma clase controlador para todos los eventos del sistema, de manera que es posible manejar la información acerca del estado del mismo en el controlador.
- **Creador:** Guía la asignación de responsabilidades relacionadas con la creación de objetos, una tarea muy común. La intención básica del patrón es encontrar un creador que necesite conectarse al objeto creado en alguna situación.
- **Experto:** Se utiliza con frecuencia en la asignación de responsabilidades; es un principio de guía básico que se utiliza continuamente en el diseño de objetos. Expresa la intuición común de que los objetos hacen las cosas relacionadas con la información que tienen.(35)

El patrón experto se evidencia en las clases controladoras `principal_entidad` y `principal_admin` que son las encargadas de mostrar todos los datos necesarios a los usuarios administradores y los usuarios entidades. La alta cohesión se ve referenciada en varias clases pero donde es más evidente es en la clase `principal_admin` cuando delega responsabilidades a la clase `entidad_c` para ejecutar las operaciones que conciernen a una entidad. El bajo acoplamiento se puede observar prácticamente en todas las clases al no existir una conexión directa entre ellas, las mismas solo dependen de algunas clases propias del framework.

Además de los patrones de diseño anteriormente mencionados también se utilizó el patrón Polimorfismo cuando se identifican variaciones en un comportamiento para asignar la responsabilidad del comportamiento a la clase y utilizar polimorfismo para implementar los comportamientos alternativos.

3.3.2 Patrones GOF

Se distinguen tres tipos de patrones GoF: patrones de comportamiento, patrones creacionales y patrones estructurales. Los patrones de comportamiento definen la comunicación e iteración entre los objetos de un sistema. El propósito de este tipo de

patrón es reducir el acoplamiento entre los objetos. Los patrones creacionales tratan con las formas de crear instancias de objetos y por último los patrones estructurales describen cómo clases y objetos pueden ser combinados para formar grandes estructuras y proporcionar nuevas funcionalidades.

- **Observer:** Define una dependencia de uno-a-muchos entre objetos, de forma que cuando un objeto cambie de estado se notifique y actualicen automáticamente todos los objetos que dependen de él.

Este patrón se pone de manifiesto cuando se realizan cambios en las respuestas de la guía, los datos cambiados deben actualizarse en la interfaz principal de administración en la tabla de resumen de la guía.

3.4 Modelo de datos

Para cumplir con los requisitos establecidos fue necesario realizar ajustes a la base de datos para ello se crearon 3 nuevas tablas y se realizaron cambios en otra para permitir ponderar los aspectos.

El siguiente diagrama muestra la relación entre las tablas de la BD.

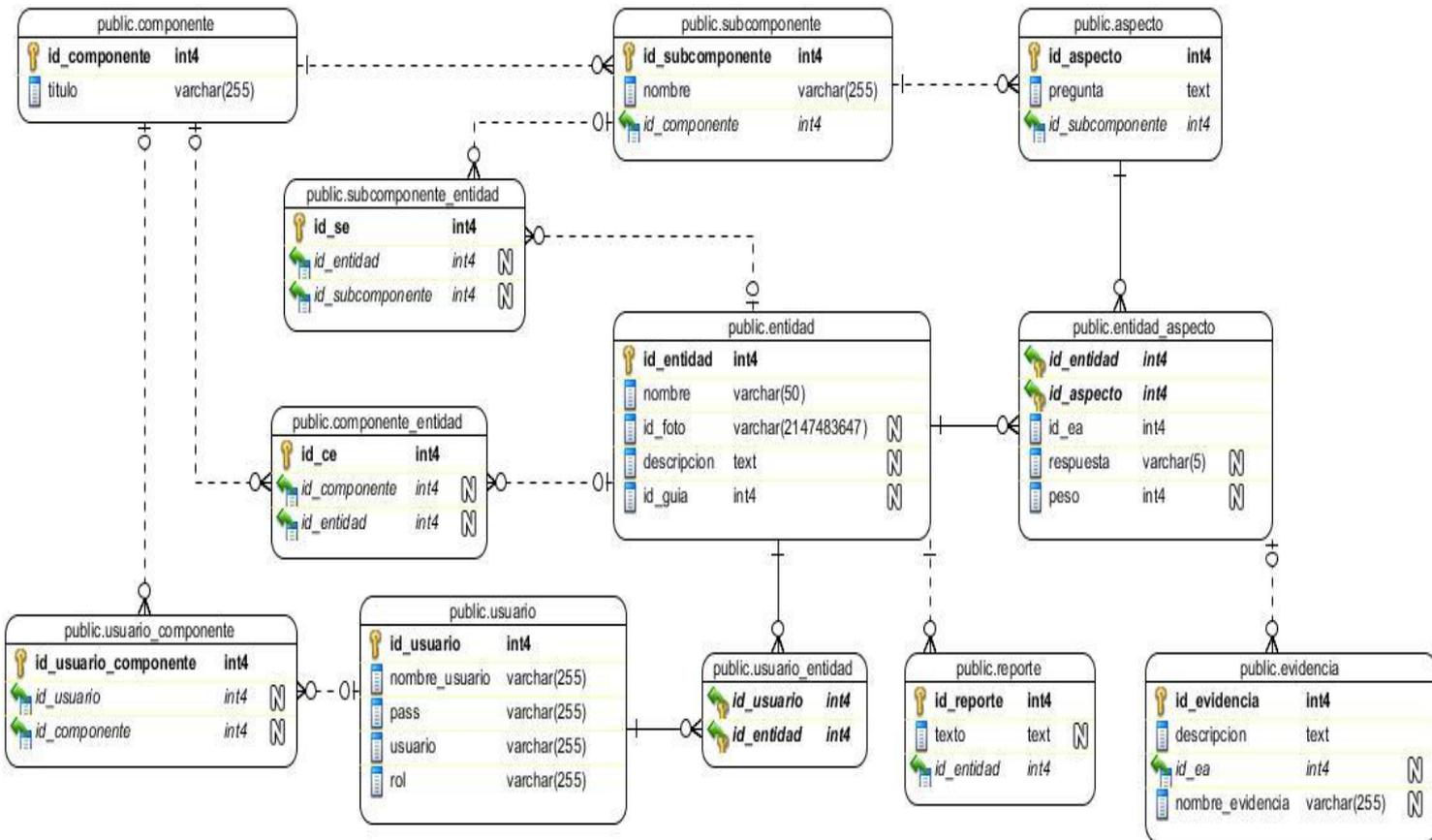


Figura 2 Modelo de datos

3.5 Implementación

Para la realización de cada historia de usuario es necesario descomponerlas en varias tareas de ingeniería para detallar más las tareas realizadas por el equipo de desarrollo y realizar posibles cambios de ser necesario. Estas tareas son puramente técnicas, no necesariamente entendibles por el cliente.

3.5.1 Primera Iteración

En la primera iteración se desarrollaron las 4 primeras HU como se definió en el plan de iteraciones.

Se definieron las siguientes tareas de ingeniería para esta iteración:

HU N⁰1: Permitir que un componente solo pueda ser evaluado por un usuario.

1. Tarea N⁰1: Realizar un estudio de la aplicación anterior.
2. Tarea N⁰2: Estudiar el framework utilizado para el desarrollo de la aplicación.
3. Tarea N⁰3: Elaborar la seguridad del sitio.

4. Tarea N^o4: Implementar la funcionalidad evaluar componente.

HU N^o2: Validar que al agregar un aspecto, los subcomponentes sean los correspondientes al componente seleccionado.

1. Tarea N^o5: Realizar cambios en la BD para poder asignar a los componentes y subcomponentes una entidad.
2. Tarea N^o6: Realizar los cambios en la clase aspecto_c.

HU N^o3: Permitir a los administradores consultar datos estadísticos en forma de gráfica.

1. Tarea N^o7: Realizar un estudio sobre la librería highcharts para el trabajo con las gráficas.
2. Tarea N^o8: Realizar los cálculos correspondientes a las estadísticas que se requieran mostrar.
3. Tarea N^o9: Mostrar los datos calculados en la interfaz de administración.

HU N^o4: Mostrar automáticamente el control para subir evidencia, si el aspecto es marcado de no.

1. Tarea N^o10: Verificar que los botones de evaluar los aspectos funcionen correctamente.
2. Tarea N^o11: Modificar la funcionalidad de subir evidencias.

Las tablas 9 y 10 muestran las primeras 2 tareas de ingeniería referentes al primer caso de HU realizado en la primera iteración, las tareas restantes de esta iteración se muestran en el [anexo 4](#).

Tabla 9. Tarea de ingeniería N^o1 HU1.

Tarea	
Número: 1	Número de HU: 1
Nombre: Realizar un estudio de la aplicación anterior.	
Tipo de tarea: Investigación	Estimación: 3 días
Fecha inicio: 20 febrero 2015	Fecha fin: 23 febrero 2015
Responsable: Gustavo Rodríguez Méndez y David Ruisánchez Jaime	
Descripción: Familiarización del equipo de desarrollo con la versión anterior del software y sus funcionalidades.	

Tabla 10. Tarea de ingeniería N^o2 HU1.

Tarea	
Número: 2	Número de HU: 1
Nombre: Estudiar el framework utilizado para el desarrollo de la aplicación.	
Tipo de tarea: Investigación	Estimación: 3 días
Fecha inicio: 24 febrero 2015	Fecha fin: 27 febrero 2015
Responsable: Gustavo Rodríguez Méndez y David Ruisánchez Jaime	
Descripción: Se realiza un estudio del framework utilizado para que el equipo de desarrollo sea capaz de familiarizarse con el mismo.	

3.5.2 Segunda Iteración

En esta iteración se desarrollaran las 7 HU restantes de las mismas se definieron las tareas de ingeniería siguientes:

HU N⁰5: Mostrar una evaluación de la guía después de ser llenada por el usuario.

1. Tarea N⁰12: Realizar los cálculos necesarios para realizar la evaluación de la guía.
2. Tarea N⁰13: Mostrar la evaluación a los administradores.

HU N⁰6: Asignar peso a cada aspecto de la guía de autocontrol.

1. Tarea N⁰14: Ponderar los aspectos mediante un peso dado por el usuario.

HU N⁰7: Crear guías nuevas a partir de aspectos de guías existentes o a partir de una completamente desde cero.

1. Tarea N⁰15: Permitir a cada entidad poder crear su propia guía.
2. Tarea N⁰16: Permitir a cada entidad poder modificar su guía.
3. Tarea N⁰17: Permitir a cada entidad poder utilizar una guía ya existente.

HU N⁰8: Determinar el porcentaje de cumplimiento de la guía.

1. Tarea N⁰18: Realizar los cálculos para determinar el porcentaje.
2. Tarea N⁰19: Mostrar el porcentaje de cumplimiento en la tabla anteriormente generada.

HU N⁰9: Mostrar a los administradores un resumen de los resultados de la aplicación de la guía de autocontrol en las entidades subordinadas al MINCOM.

1. Tarea N⁰20: Recopilar los datos necesarios para confeccionar la tabla que se desea mostrar.

2. Tarea N°21: Generar la tabla a mostrar.

HU N°10: Exportar el resultado de la guía después de terminada esta.

1. Tarea N°22: Exportar la tabla anteriormente generada.

HU N°11: Incorporar preguntas referentes a la gestión de información basados en el marco de trabajo COBIT como aspectos de la guía de autocontrol.

1. Tarea N°23: Realizar una investigación sobre el marco de trabajo COBIT.
2. Tarea N°24: Redactar las preguntas correspondientes.
3. Tarea N°25: Incorporar estas preguntas en la guía de autocontrol.

En las siguientes tablas se muestran las tareas de ingeniería correspondientes a la segunda iteración, las tareas restantes se pueden visualizar en el [anexo 5](#).

Tabla 11. Tarea de ingeniería N°12 HU5.

Tarea	
Número: 12	Número de HU: 5
Nombre: Realizar los cálculos necesarios para realizar la evaluación de la guía.	
Tipo de tarea: Desarrollo	Estimación: 5 días
Fecha inicio: 23 marzo 2015	Fecha fin: 28 marzo 2015
Responsable: Gustavo Rodríguez Méndez	
Descripción: Calcular mediante una fórmula matemática los resultados de la aplicación de la guía.	

Tabla 12. Tarea de ingeniería N°13 HU5.

Tarea	
Número: 13	Número de HU: 5
Nombre: Mostrar la evaluación a los administradores.	
Tipo de tarea: Desarrollo	Estimación: 2 días
Fecha inicio: 28 marzo 2015	Fecha fin: 30 marzo 2015
Responsable: Gustavo Rodríguez Méndez	
Descripción: Mostrar mediante una etiqueta cual fue el resultado luego de evaluar la guía.	

3.6 Pruebas

El único instrumento adecuado para determinar el estatus de la calidad de un producto software es el proceso de pruebas. En este proceso se ejecutan pruebas dirigidas a

componentes del software o al sistema de software en su totalidad, con el objetivo de medir el grado en que el software cumple con los requerimientos.

3.6.1 Pruebas Unitarias

Las pruebas unitarias o pruebas de caja blanca se basan en realizar un conjunto de pruebas al código del sistema. Estas pruebas aseguran que un determinado módulo o clase (en dependencia del lenguaje de programación que se esté utilizando) cumpla con el comportamiento esperado. Permiten al programador saber si una determinada funcionalidad se puede agregar al sistema existente sin alterar el funcionamiento actual del mismo. Se realizan a los principales algoritmos o procedimientos(36).

Ejemplo de prueba unitaria realizada

```
public function testPorcentaje() {
    $this->Instance->load->model('ministerio_model', 'em');
    $result = $this->em->porcentaje(50, 100);
    $this->assertEquals(50, $result);
    $result = $this->em->porcentaje(25, 100);
    $this->assertEquals(25, $result);
    $result = $this->em->porcentaje(25, 50);
    $this->assertEquals(50, $result);
}
```

Figura 3 Fragmento de código de las pruebas unitarias

3.6.2 Pruebas de aceptación

El uso de cualquier producto de software tiene que estar justificado por las ventajas que ofrece. Sin embargo, antes de empezar a usarlo es muy difícil determinar si sus ventajas realmente justifican su uso. El mejor instrumento para esta determinación es la llamada 'prueba de aceptación'. En esta prueba se evalúa el grado de calidad del software con relación a todos los aspectos relevantes para que el uso del producto se justifique.

Tabla 13. Caso de prueba de aceptación HU1_P1.

Caso de prueba de aceptación	
Código: HU1_P1	Historia de Usuario: 1
Nombre: Permitir que un componente solo pueda ser evaluado por un usuario.	

<p>Descripción: Prueba para verificar que un componente solo pueda ser evaluado por un solo usuario.</p>
<p>Condiciones de Ejecución: El administrador debe haber creado más de un solo usuario perteneciente a su entidad,</p>
<p>Entrada/ Pasos de ejecución: El administrador debe asignar manualmente los componentes que evaluará cada usuario.</p>
<p>Resultado Esperado: Si ya están asignados los componentes a otro usuario no debe permitir que se le asignen a otro usuario.</p>
<p>Evaluación de la Prueba: Prueba satisfactoria.</p>

Durante la realización de las pruebas se encontraron varias no conformidades, en el caso de prueba anteriormente mencionado se encontró una no conformidad significativa debido a que a la hora de asignar los componentes a evaluar por cada usuario la aplicación se los asignaba a los mismos pero los mismos podían ser asignados a otros usuarios siendo esto un error, esta no conformidad fue erradicada permitiendo el correcto funcionamiento de esta funcionalidad.

3.6.3 Resultado de las pruebas

Se encontraron además varias no conformidades no significativas como tildes y errores ortográficos presentes en las interfaces de la aplicación, entre ellos estaban los nombres de los componentes en la interfaz que muestra las gráficas los cuales no presentaban acentos en varias palabras, también cuando se agregaba una evidencia en el mensaje mostrado la palabra éxito no estaba acentuada, estas no conformidades fueron completamente erradicadas. También se encontraron otras no conformidades significativas como en la funcionalidad de crear guía, si se creaba una nueva entidad cuando se utilizaba la opción de utilizar una guía existente no mostraba los datos requeridos. Cuando el usuario se registraba como root y accedía a la opción crear usuario la interfaz no mostraba los componentes bloqueados para el mismo debido a que a los administradores no se le asignan componentes a evaluar, esta es una no conformidad significativa.

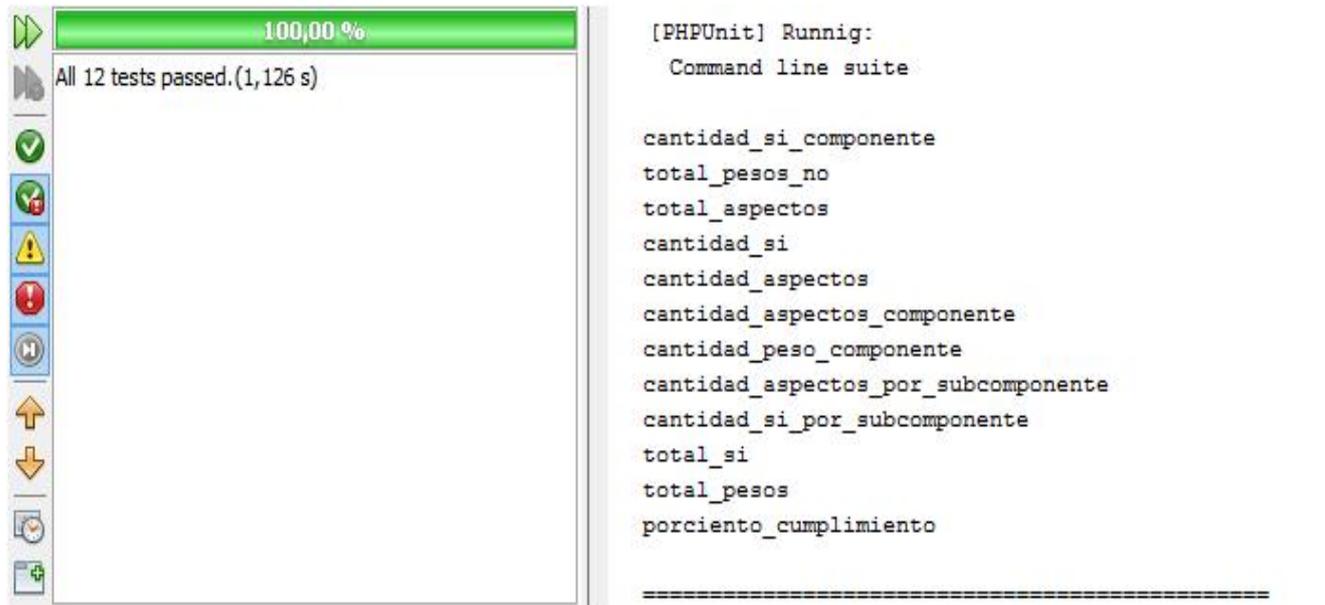


Figura 4 Resultados de las pruebas unitarias

En la tabla que se presenta a continuación se muestran los resultados de las pruebas en cada iteración:

Tabla 14. Resultado de las pruebas.

	Iteración 1	Iteración 2
No conformidades	14	8
Significativas	10	4
No significativas	4	4
Resueltas	14	8
Pendientes	0	0

3.7 Discusión de los resultados.

Para tener una mayor comprensión del software obtenido se prosigue a realizar una breve descripción del mismo luego de concluida la implementación, donde se explicarán los principales aspectos del Sistema de gestión de información para el control interno del Ministerio de Comunicaciones.

En la siguiente imagen se muestra la interfaz principal del sistema correspondiente al módulo Administración, donde el administrador puede realizar una serie de acciones con el sistema dentro de las cuales se encuentran:

- Visualizar las gráficas correspondientes a los resultados de la aplicación de la guía de autocontrol en su empresa.
- Listar todas las entidades pertenecientes al ministerio.
- Crear un usuario nuevo en la aplicación.
- Crear un nuevo componte en la aplicación.
- Crear un nuevo subcomponente en la aplicación.
- Crear un aspecto nuevo en la aplicación.
- Mostrar las incidencias redactadas por los usuarios de cada entidad.
- Exportar los resultados de la aplicación de la guía de autocontrol en forma de imagen.

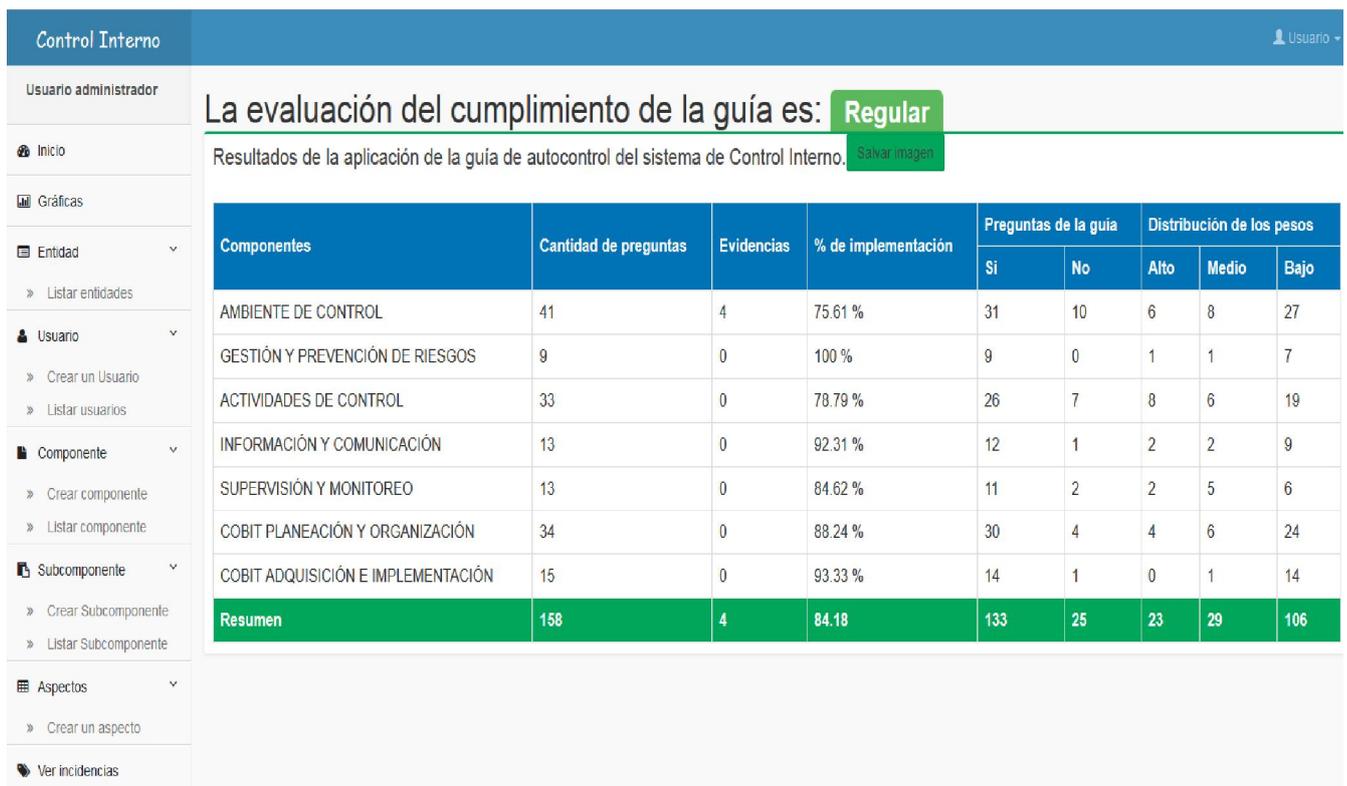


Figura 5 Interfaz gráfica del módulo Administración

Como se mencionó anteriormente una de las funciones que puede realizar el administrador es la de listar entidades, la imagen siguiente muestra el listado de entidades y las acciones que puede realizar el administrador con el mismo.

Control Interno		Usuario +
Usuario administrador	Listado de entidades	
Inicio	Nombre	Acciones
Gráficas	ACS	[Editar] [Ver] [Eliminar]
Entidad <	ALBET	[Editar] [Ver] [Eliminar]
Usuario <	CALISOFT	[Editar] [Ver] [Eliminar]
Componente <	CDEA	[Editar] [Ver] [Eliminar]
Subcomponente <	COPAL	[Editar] [Ver] [Eliminar]
Aspectos <	COPEXTEL	[Editar] [Ver] [Eliminar]
Ver incidencias	DELTA	[Editar] [Ver] [Eliminar]
	DESOFT	[Editar] [Ver] [Eliminar]
	ECC	[Editar] [Ver] [Eliminar]
	ESAC	[Editar] [Ver] [Eliminar]
	ETECSA	[Editar] [Ver] [Eliminar]
	FORDES	[Editar] [Ver] [Eliminar]
	GE	[Editar] [Ver] [Eliminar]

Figura 6 Interfaz gráfica del listado de entidades

Luego de mostrado el listado de entidades el administrador es capaz de editar el nombre de las mismas, cambiar aspectos de su guía de autocontrol o simplemente eliminarlas de la aplicación. Para realizar estas acciones el administrador debe tener los permisos correspondientes, debido a que si es administrador de una entidad específica solo podrá modificar los datos relacionados con su entidad. Este cambio resuelve un problema de seguridad de la versión anterior.

Otra de las interfaces críticas del software es la que permite crear aspectos pues los mismos pueden presentar un formato diferente al momento de ser redactados debido a que algunos de ellos incluyen subaspectos y esta interfaz debe ser capaz de proporcionar un adecuado formato para su inclusión en la guía como se muestra en la Figura 7. En la versión anterior no era posible asociar a un aspecto varios subaspectos, por lo que esta facilidad también representa una mejora con respecto a la versión 1.0.

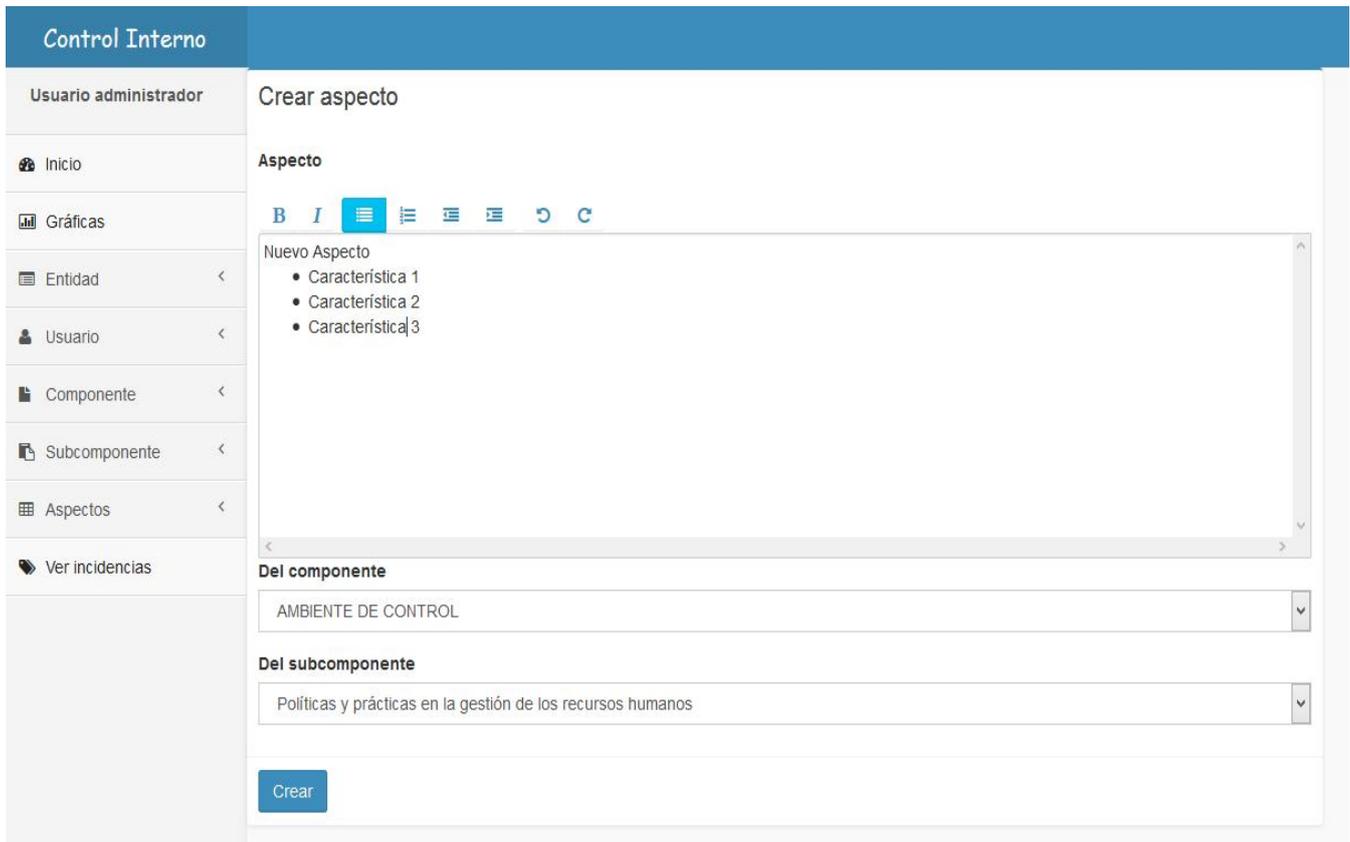


Figura 7 Interfaz gráfica para crear aspectos a las guías en edición

Por último se muestra la interfaz principal del módulo Entidad que es la encargada de mostrar la guía de autocontrol a los usuarios encargados de dar respuesta a la misma. En ella los usuarios son capaces de determinar el nivel de importancia de los aspectos acorde a las necesidades de su organización y de presentar evidencias que expongan las razones por las que no se cumplen los aspectos de la guía. Lo anteriormente expuesto se evidencia en la siguiente imagen.

Componente			
AMBIENTE DE CONTROL			
Subcomponente			
Planeación, planes de trabajo anual, mensual e individual			
Aspecto	Ponderar	Respuesta	Acciones
Definidos los objetivos de trabajo de la entidad.	Alto	<input type="radio"/> Si <input checked="" type="radio"/> No	 No existe evidencia
El plan anual de actividades asegura la correspondencia entre los objetivos de trabajo, las actividades y los recursos de la entidad, respondiendo a la instrumentación de las actividades recogidas en el plan de actividades principales del órgano u organismo al que se subordina, de acuerdo a las regulación aprobada por el Presidente de los Consejos de Estado y de Ministros para la planificación.	Bajo	<input checked="" type="radio"/> Si <input type="radio"/> No	
Para la elaboración del plan anual de actividades, se tiene en cuenta entre otros aspectos los siguientes: <ul style="list-style-type: none"> • La misión de la entidad. • Los objetivos de trabajo de la entidad. • Las funciones definidas de cada área y puesto de trabajo. • Las actividades a realizar en cada proceso o subproceso. • Los riesgos más relevantes o sea aquellos que ponen en peligro el cumplimiento de los objetivos y la misión de la entidad. • El Plan de actividades debe contener las tareas de consulta y discusión del Plan económico y el Presupuesto anual con los trabajadores en todas sus 	Bajo	<input checked="" type="radio"/> Si <input type="radio"/> No	

Figura 8 Interfaz gráfica de la guía de autocontrol

Luego de presenciar las principales interfaces de la aplicación y las funcionalidades que estas brindan a los usuarios del sistema es válido mencionar que la mejora al sistema cumple con todos los requisitos especificados por el cliente y es capaz de proporcionar una herramienta para monitorear el control interno dentro de las entidades subordinadas al Ministerio de Comunicaciones.

3.8 Conclusiones Parciales

Los resultados descritos en el presente capítulo muestran cómo el sistema propuesto gestiona la información necesaria para conocer en cualquier momento el estado del control interno en las empresas del MINCOM. La realización de las pruebas de unidad minimizó los errores en el código desarrollado, lo que permitió un correcto funcionamiento del sistema.

CONCLUSIONES GENERALES

Con la culminación del presente trabajo se arribó a las siguientes conclusiones:

- El estudio realizado a los diferentes sistemas de gestión de información evidenció la necesidad de incluir al Sistema de Gestión de Información para el Control Interno del MINCOM indicadores relacionados con las Tecnologías de la Información.
- Con la realización de las nuevas funcionalidades para la el software del Ministerio de Comunicaciones se garantiza un mejor funcionamiento del mismo y una mayor comunicación con el usuario.
- Al término del proceso de pruebas realizado a la solución propuesta, mediante el empleo de las pruebas de aceptación se demostró el correcto funcionamiento de la aplicación y sus funcionalidades, dando cumplimiento a las expectativas y necesidades del cliente.

REFERENCIAS BIBLIOGRAFICAS

1. NO, RESOLUCIÓN. CONTRALORÍA GENERAL DE LA REPÚBLICA DE CUBA. [online]. [Accessed 8 February 2015]. Available from: <http://www.medigraphic.com/pdfs/infodir/efd-2013/efd1316q.pdf>
2. (COSO), Committe of sponsoring organizations of the threadway commission and BLANCO, S. A. M. *Control interno: informe COSO* [online]. Ecoe Ediciones, 2005. ISBN 9789586483469. Available from: https://books.google.com/cu/books?id=p_LQDu4L66MC
3. DANTE, G. P. *Gestión de información en las organizaciones: principios, conceptos y aplicaciones* [online]. Universidad de Chile, 1998. ISBN 9789567782000. Available from: <http://books.google.co.ve/books?id=ZrVpAAAACAAJ>
4. MOREIRO GONZÁLEZ, José Antonio. *Introducción al estudio de la información y la documentación*. Editora de la Universidad de Antioquia, 1998.
5. SCHUSTER, J. A. *Control interno* [online]. Macchi, 1992. ISBN 9789505372065. Available from: <http://books.google.com/cu/books?id=A1uyAAAACAAJ>
6. BARQUERO, Miguel. El Control Interno en trabajos de fiscalización. *Auditoría pública: revista de los Organos Autónomos de Control Externo*. 2009. No. 47, p. 47–56.
7. El Marco de Trabajo de COBIT. | Sociedad de la Información. [online]. [Accessed 20 March 2015]. Available from: <https://societaddelainformacion.wordpress.com/2007/06/24/el-marco-de-trabajo-de-cobit/>
8. 4. COBIT(Objetivos de Control para la Información y Tecnologías Relacionadas) - Auditoria en Informatica CUN. [online]. [Accessed 17 March 2015]. Available from: <https://sites.google.com/site/auditoriaeninformaticacun/cobit-1>
9. SOFTWARE & CONSULTING. [online]. [Accessed 8 February 2015]. Available from: <http://www.soft-consulting.pe/software/detalle/3>
10. *Microsoft Word - Cobit Audit.doc - Cobit Audit.pdf* [online]. [Accessed 17 March 2015]. Available from: <http://www.amsriskconsulting.com/Cobit%20Audit.pdf>
11. Meycor COBIT CSA | Meycor. [online]. [Accessed 17 March 2015]. Available from: <http://www.meycor-soft.com/es/meycor-csa>
12. CANÓS, Joseph, LETELIER, Patricio and PENADÉS, M^a Carmen. Metodologías Ágiles en el desarrollo de Software. *Universidad Politécnica de Valencia, Valencia* [online]. 2003. [Accessed 8 February 2015]. Available from: http://www.willydev.net/Willydev_old/Root/descargas/prev/ToDoAgil.Pdf

13. CEDIEL, Orlando Torres. SCRUM, la metodología ágil de desarrollo más utilizada, sus dificultades y el reto de solución a través de una aplicación móvil. *Tecnología Investigación y Academia* [online]. 2015. Vol. 2, no. 2. [Accessed 9 February 2015]. Available from: <http://revistas.udistrital.edu.co/ojs/index.php/tia/article/view/5753>
14. *todo.PDF - xp.pdf* [online]. [Accessed 9 February 2015]. Available from: <https://met2unaj.files.wordpress.com/2014/09/xp.pdf>
15. CyTA. [online]. [Accessed 15 April 2015]. Available from: http://www.cyta.com.ar/ta0502/b_v5n2a1.htm
16. Ventajas y Desventajas - Metodología XP. [online]. [Accessed 17 March 2015]. Available from: <https://sites.google.com/site/xpmetodologia/marco-teorico/ventajas>
17. 1.1. Symfony en pocas palabras (Symfony 1.2, la guía definitiva). [online]. [Accessed 9 February 2015]. Available from: http://librosweb.es/symfony/capitulo_1/symfony_en_pocas_palabras.html
18. BARYOLO, Oiner Gómez. Sistema de gestión integral de seguridad Acaxia. *Serie Científica* [online]. 2011. Vol. 4, no. 7. [Accessed 9 February 2015]. Available from: <https://publicaciones.uci.cu/index.php/SC/article/view/702>
19. Definición de lenguaje de programación - Qué es, Significado y Concepto. [online]. [Accessed 9 February 2015]. Available from: <http://definicion.de/lenguaje-de-programacion/>
20. Lección 1: ¿Qué es CSS?spatutorial - HTML.net. [online]. [Accessed 9 February 2015]. Available from: <http://es.html.net/tutorials/css/lesson1.php>
21. Capítulo 1. Introducción (Introducción a JavaScript). [online]. [Accessed 9 February 2015]. Available from: http://librosweb.es/javascript/capitulo_1.html
22. Definición de html - Qué es, Significado y Concepto. [online]. [Accessed 9 February 2015]. Available from: <http://definicion.de/html/>
23. PHP: Prefacio - Manual. [online]. [Accessed 9 February 2015]. Available from: <http://php.net/manual/es/preface.php>
24. Apache - openSUSE. [online]. [Accessed 17 March 2015]. Available from: <https://es.opensuse.org/Apache>
25. *Microsoft Word - intro_case_SA.doc - intro_case_SA.pdf* [online]. [Accessed 9 February 2015]. Available from: http://users.dsic.upv.es/asignaturas/eui/mtp/doc-practicas/intro_case_SA.pdf
26. Software Design Tools for Agile Teams, with UML, BPMN and More. [online]. [Accessed 9 February 2015]. Available from: <http://www.visual-paradigm.com/>

27. MARÍA TERESA GARZÓN PÉREZ. Sistemas gestores de base de datos. *Revista Digital CSI* [online]. May 2010. Vol. 30. Available from: http://www.csi-csif.es/andalucia/mod_ense-csifrevistad_30.html
28. Sobre PostgreSQL | www.postgresql.org.es [online]. [Accessed 9 February 2015]. Available from: http://www.postgresql.org.es/sobre_postgresql
29. NetBeans. [online]. [Accessed 9 February 2015]. Available from: <https://www.fdi.ucm.es/profesor/luis/fp/devtools/NetBeansUso.html>
30. Bienvenido a NetBeans y www.netbeans.org, Portal del IDE Java de Código Abierto. [online]. [Accessed 9 February 2015]. Available from: https://netbeans.org/index_es.html
31. MARIÑO, Sonia, GODOY, María, ALFONZO, Pedro, ACEVEDO, Juan, SOLIS, Laura Gómez and VÁZQUEZ, Agustina Fernández. Accesibilidad en la definición de requerimientos no funcionales. Revisión de herramientas. *Multiciencias* [online]. 2012. Vol. 12, no. 3. [Accessed 23 April 2015]. Available from: <http://www.produccioncientifica.luz.edu.ve/index.php/multiciencias/article/view/16913>
32. SOMMERVILLE, Ian. *Ingeniería de Software*. Séptima Edición. Madrid : Pearson Educación, 2005. ISBN 84-7829-074-5.
33. GÓMEZ, Alveiro Rosado, DUARTE, Alexander Quintero and GÜEVARA, Cesar Daniel Meneses. Desarrollo ágil de software aplicando programación extrema. *Revista Ingenio*. 2014. Vol. 5, no. 1, p. 24–29.
34. GONZÁLEZ, Yanette Díaz and ROMERO, Yenisleidy Fernández. Patrón Modelo-Vista-Controlador. *Revista Telem@tica*. 2012. Vol. 11, no. 1, p. 47–57.
35. GIRALDO, Gloria L., ACEVEDO, Juan F. and MORENO, David A. Una ontología para la representación de conceptos de diseño de software. *Revista Avances en Sistemas e Informática*. 2011. Vol. 8, no. 3, p. 103–110.
36. PRESSMAN, Roger S. *Ingeniería del Software, Un Enfoque Práctico*. 5ta edición. McGraw-Hill, 2002. ISBN 8448132149.

BIBLIOGRAFÍA

1.1. Symfony en pocas palabras (Symfony 1.2, la guía definitiva). [en línea] [sin fecha]. [Consulta: 9 febrero 2015]. Disponible en: http://librosweb.es/symfony/capitulo_1/symfony_en_pocas_palabras.html.

4. COBIT(Objetivos de Control para la Información y Tecnologías Relacionadas) - Auditoria en Informatica CUN. [en línea] [sin fecha]. [Consulta: 17 marzo 2015]. Disponible en: <https://sites.google.com/site/auditoriaeninformaticacun/cobit-1>.

AJA QUIROGA, L. 2002. Gestión de información, gestión del conocimiento y gestión de la calidad en las organizaciones. *Acimed*, vol. 10, no. 5, pp. 7–8.

Apache - openSUSE. [en línea] [sin fecha]. [Consulta: 17 marzo 2015]. Disponible en: <https://es.opensuse.org/Apache>.

ARNOL, K. y GOSLING, J. 1997. *El Lenguaje de programación Java*. S.l.: Pearson Educación.

AVILA MÉNDEZ, M.A. y CABRERA CAMPOVERDE, D.X. 2013. Evaluación al control interno al área de clientes en las Heladerías Tutto Fredo SA, período: 1 de enero al 30 de noviembre del 2011. [en línea], [Consulta: 8 febrero 2015]. Disponible en: <http://dspace.ucuenca.edu.ec/handle/123456789/343>.

BARQUERO, M. 2009. El Control interno en trabajos de fiscalización. *Auditoría pública: revista de los Organos Autónomos de Control Externo*, no. 47, pp. 47–56.

BARYOLO, O.G. 2011. Sistema de gestión integral de seguridad Acaxia. *Serie Científica* [en línea], vol. 4, no. 7. [Consulta: 9 febrero 2015]. Disponible en: <https://publicaciones.uci.cu/index.php/SC/article/view/702>.

Bienvenido a CodeIgniter : Guía de Usuario de CodeIgniter. [en línea] [sin fecha]. [Consulta: 9 febrero 2015]. Disponible en: http://escodeigniter.com/guia_usuario/.

Bienvenido a NetBeans y www.netbeans.org, Portal del IDE Java de Código Abierto. [en línea] [sin fecha]. [Consulta: 9 febrero 2015]. Disponible en: https://netbeans.org/index_es.html.

CANÓS, J., LETELIER, P. y PENADÉS, M.C. 2003. Metodologías Ágiles en el desarrollo de Software. *Universidad Politécnica de Valencia, Valencia* [en línea], [Consulta: 8 febrero 2015]. Disponible en: http://www.willydev.net/Willydev_old/Root/descargas/prev/TodoAgil.Pdf.

Capítulo 1. Introducción (Introducción a JavaScript). [en línea] [sin fecha]. [Consulta: 9 febrero 2015]. Disponible en: http://librosweb.es/javascript/capitulo_1.html.

CEDIEL, O.T. 2015. SCRUM, la metodología ágil de desarrollo más utilizada, sus dificultades y el reto de solución a través de una aplicación móvil. *Tecnología*

Investigación y Academia [en línea], vol. 2, no. 2. [Consulta: 9 febrero 2015]. Disponible en: <http://revistas.udistrital.edu.co/ojs/index.php/tia/article/view/5753>.

(COSO), C. of sponsoring organizations of the threadway commission y BLANCO, S.A.M. 2005. *Control interno: informe COSO* [en línea]. S.l.: Ecoe Ediciones. ISBN 9789586483469. Disponible en: https://books.google.com.cu/books?id=p_LQDu4L66MC.

CYOARTE STUDIO S.L. [en línea] [sin fecha]. [Consulta: 9 febrero 2015]. Disponible en: <http://www.cyoarte.com/corporate/modules/rmlib/print.php?id=3>.

CyTA. [en línea] [sin fecha]. [Consulta: 15 abril 2015]. Disponible en: http://www.cyta.com.ar/ta0502/b_v5n2a1.htm.

DANTE, G.P. 1998. *Gestión de información en las organizaciones: principios, conceptos y aplicaciones* [en línea]. S.l.: Universidad de Chile. ISBN 9789567782000. Disponible en: <http://books.google.co.ve/books?id=ZrVpAAAACAAJ>.

Definición de html - Qué es, Significado y Concepto. [en línea] [sin fecha]. [Consulta: 9 febrero 2015]. Disponible en: <http://definicion.de/html/>.

Definición de lenguaje de programación - Qué es, Significado y Concepto. [en línea] [sin fecha]. [Consulta: 9 febrero 2015]. Disponible en: <http://definicion.de/lenguaje-de-programacion/>.

DERNIAME, J.C., KABA, B.A. y WASTELL, D. 1999. *Software process: principles, methodology, and technology*. Berlin; New York: Springer. ISBN 3540655166 9783540655169.

Diseño de patrones. [en línea] [sin fecha]. [Consulta: 17 abril 2015]. Disponible en: <http://es.slideshare.net/EdwinRomanCastrillon/diseo-de-patrones>.

Eclipse IDE. [en línea] [sin fecha]. [Consulta: 9 febrero 2015]. Disponible en: <http://www.genbetadev.com/herramientas/eclipse-ide>.

El Informe COSO I y II. [en línea] [sin fecha]. [Consulta: 19 marzo 2015]. Disponible en: <http://www.auditool.org/blog/control-interno/290-el-informe-coso-i-y-ii>.

El Marco de Trabajo de COBIT. | Sociedad de la Información. [en línea] [sin fecha]. [Consulta: 20 marzo 2015]. Disponible en: <https://sociedaddelainformacion.wordpress.com/2007/06/24/el-marco-de-trabajo-de-cobit/>.

ENTORNO DE DESARROLLO INTEGRADO(IDE) | alanss18. [en línea] [sin fecha]. [Consulta: 9 febrero 2015]. Disponible en: <https://alanss18.wordpress.com/2013/01/25/entorno-de-desarrollo-integradoide/>.

GARCÍA, A.E. 1998. *Sistemas de Información. Saint Claire: Buenos Aires* [en línea],

[Consulta: 8 febrero 2015]. Disponible en: <http://www.uab.cat/guiesdocents/2014-15/g102389t2501572a2014-15iESP.pdf>.

GARCÍA ORTIZ, R.M. y ZAPATA CHIGUANO, J.M. 2014. Diseño de un sistema de control interno para la cuenta de inventario en el centro comercial «Su Economía» en el cantón La Maná, provincia de Cotopaxi, para el periodo 1 de enero al 31 de marzo del 2013. [en línea], [Consulta: 8 febrero 2015]. Disponible en: <http://repositorio.utc.edu.ec/handle/27000/1970>.

GIRALDO, G.L., ACEVEDO, J.F. y MORENO, D.A. 2011. Una ontología para la representación de conceptos de diseño de software. *Revista Avances en Sistemas e Informática*, vol. 8, no. 3, pp. 103–110.

GÓMEZ, A.R., DUARTE, A.Q. y GÜEVARA, C.D.M. 2014. Desarrollo ágil de software aplicando programación extrema. *Revista Ingenio*, vol. 5, no. 1, pp. 24–29.

GONZÁLEZ, Y.D. y ROMERO, Y.F. 2012. Patrón Modelo-Vista-Controlador. *Revista Telemática*, vol. 11, no. 1, pp. 47–57.

HERNÁNDEZ, J.L. 1990. La gestión de la información en las organizaciones: una disciplina emergente. *Documentación de las Ciencias de la Información*, vol. 13, pp. 133.

<http://www.javahispano.org/articles.print.action?id=76-cualxpfdrrup.pdf> [en línea], [sin fecha]. S.l.: s.n. [Consulta: 9 febrero 2015]. Disponible en: <http://www.willydev.net/descargas/articulos/general/cualxpfdrrup.pdf>.

IBM - Rational Rose Enterprise. [en línea] [sin fecha]. [Consulta: 9 febrero 2015]. Disponible en: <http://www-03.ibm.com/software/products/es/enterprise>.

Introducción a Spring Framework Java | Picando Código. [en línea] [sin fecha]. [Consulta: 9 febrero 2015]. Disponible en: <http://picandocodigo.net/2010/introduccion-a-spring-framework-java/>.

ISO 20000 Calidad de los Servicios TI | Normas ISO. [en línea] [sin fecha]. [Consulta: 8 febrero 2015]. Disponible en: <http://www.normas-iso.com/iso-20000>.

ITIL: ¿qué es y para qué sirve? (parte 1) | Magazciturum. [en línea] [sin fecha]. [Consulta: 20 marzo 2015]. Disponible en: <http://www.magazciturum.com.mx/?p=50>.

LARMAN, C. 2002. *UML y patrones: una introducción al análisis y diseño orientado a objetos y al proceso unificado*. Madrid [etc.]: Prentice Hall. ISBN 8420534382 9788420534381.

Las pruebas de software forman el único instrumento adecuado para determinar la calidad de software. [en línea] [sin fecha]. [Consulta: 28 mayo 2015]. Disponible en: <http://pruebasdesoftware.com/laspruebasdesoftware.htm>.

Lección 1: ¿Qué es CSS?spatutorial - HTML.net. [en línea] [sin fecha]. [Consulta: 9 febrero 2015]. Disponible en: <http://es.html.net/tutorials/css/lesson1.php>.

MARÍA TERESA GARZÓN PÉREZ 2010. Sistemas gestores de base de datos. *Revista Digital CSI* [en línea], vol. 30. Disponible en: http://www.csi-csif.es/andalucia/mod_ense-csifrevistad_30.html.

MARIÑO, S., GODOY, M., ALFONZO, P., ACEVEDO, J., SOLIS, L.G. y VÁZQUEZ, A.F. 2012. Accesibilidad en la definición de requerimientos no funcionales. Revisión de herramientas. *Multiciencias* [en línea], vol. 12, no. 3. [Consulta: 23 abril 2015]. Disponible en: <http://www.produccioncientifica.luz.edu.ve/index.php/multiciencias/article/view/16913>.

METODOLOGIA Rational Unified Process (RUP) - RUP vs. XP.pdf [en línea], [sin fecha]. S.l.: s.n. [Consulta: 9 febrero 2015]. Disponible en: <http://www.usmp.edu.pe/publicaciones/boletin/fia/info49/articulos/RUP%20vs.%20XP.pdf>.

Meycor COBIT CSA | Meycor. [en línea] [sin fecha]. [Consulta: 17 marzo 2015]. Disponible en: <http://www.meycor-soft.com/es/meycor-csa>.

Meycor COSO Web | Meycor. [en línea] [sin fecha]. [Consulta: 8 febrero 2015]. Disponible en: <http://www.meycor-soft.com/es/meycor-coso>.

Microsoft Word - Cobit Audit.doc - Cobit Audit.pdf [en línea], [sin fecha]. S.l.: s.n. [Consulta: 17 marzo 2015]. Disponible en: <http://www.amsriskconsulting.com/Cobit%20Audit.pdf>.

Microsoft Word - intro_case_SA.doc - intro_case_SA.pdf [en línea], [sin fecha]. S.l.: s.n. [Consulta: 9 febrero 2015]. Disponible en: http://users.dsic.upv.es/asignaturas/eui/mtp/doc-practicas/intro_case_SA.pdf.

MOREIRO GONZÁLEZ, J.A. 1998. *Introducción al estudio de la información y la documentación*. S.l.: Editora de la Universidad de Antioquia.

NetBeans. [en línea] [sin fecha]. [Consulta: 9 febrero 2015]. Disponible en: <https://www.fdi.ucm.es/profesor/luis/fp/devtools/NetBeansUso.html>.

NO, R. [sin fecha]. CONTRALORÍA GENERAL DE LA REPÚBLICA DE CUBA. [en línea], [Consulta: 8 febrero 2015]. Disponible en: <http://www.medigraphic.com/pdfs/infodir/ifd-2013/ifd1316q.pdf>.

Normas ISO. [en línea] [sin fecha]. [Consulta: 9 febrero 2015]. Disponible en: <http://www.unlu.edu.ar/~ope20156/normasiso.htm>.

OMAR HERNANDEZ [sin fecha]. Presentacion Control interno Coso Es. [en línea]. [Consulta: 7 febrero 2015]. Disponible en: <http://es.slideshare.net/realma/presentacion->

control-interno-coso-es.

Online Browsing Platform (OBP). [en línea] [sin fecha]. [Consulta: 8 febrero 2015]. Disponible en: <https://www.iso.org/obp/ui/#iso:std:iso-iec:20000:-1:ed-2:v1:en>.

Oracle | Hardware and Software, Engineered to Work Together. [en línea] [sin fecha]. [Consulta: 9 febrero 2015]. Disponible en: <http://www.oracle.com/index.html>.

ORGANIZACION Y GESTION DE LA INFORMACION Y LA COMUNICACION - sistemasdeinformacion.pdf [en línea], [sin fecha]. S.l.: s.n. [Consulta: 8 febrero 2015]. Disponible en: <http://www.ual.es/~jmrodri/sistemasdeinformacion.pdf>.

Patrones de diseño: qué son y por qué debes usarlos. [en línea] [sin fecha]. [Consulta: 17 abril 2015]. Disponible en: <http://www.genbetadev.com/metodologias-de-programacion/patrones-de-diseno-que-son-y-por-que-debes-usarlos>.

PHP: Prefacio - Manual. [en línea] [sin fecha]. [Consulta: 9 febrero 2015]. Disponible en: <http://php.net/manual/es/preface.php>.

PINCIROLI, F. 2005. Necesidad del empleo de herramientas estándares en XP. *XI Congreso Argentino de Ciencias de la Computación* [en línea]. S.l.: s.n., [Consulta: 16 abril 2015]. Disponible en: <http://sedici.unlp.edu.ar/handle/10915/23050>.

¿Qué es ITIL? | Seguridad Informática. [en línea] [sin fecha]. [Consulta: 20 marzo 2015]. Disponible en: <https://seguinfo.wordpress.com/2008/12/03/%C2%BFque-es-itol-2/>.

Qué es Oracle. [en línea] [sin fecha]. [Consulta: 9 febrero 2015]. Disponible en: <http://www.desarrolloweb.com/articulos/840.php>.

¿Qué es y para qué sirve HTML? El lenguaje más importante para crear páginas webs. HTML tags (CU00704B). [en línea] [sin fecha]. [Consulta: 9 febrero 2015]. Disponible en:

http://www.aprenderaprogramar.com/index.php?option=com_content&view=article&id=435:ique-es-y-para-que-sirve-html-el-lenguaje-mas-importante-para-crear-paginas-webs-html-tags-cu00704b&catid=69:tutorial-basico-programador-web-html-desde-cero&Itemid=192.

¿Qué son las normas ISO? [en línea] [sin fecha]. [Consulta: 9 febrero 2015]. Disponible en: <http://www.muyhistoria.es/curiosidades/preguntas-respuestas/que-son-las-normas-iso-641381741552>.

SCHUSTER, J.A. 1992. *Control interno* [en línea]. S.l.: Macchi. ISBN 9789505372065. Disponible en: <http://books.google.com.cu/books?id=A1uyAAAACAAJ>.

Sistemas de Control. [en línea] [sin fecha]. [Consulta: 7 febrero 2015]. Disponible en: http://es.slideshare.net/ptah_enki/sistemas-de-control.

Sobre PostgreSQL | www.postgresql.org.es. [en línea] [sin fecha]. [Consulta: 9 febrero 2015]. Disponible en: http://www.postgresql.org.es/sobre_postgresql.

SOFTWARE & CONSULTING. [en línea] [sin fecha]. [Consulta: 8 febrero 2015]. Disponible en: <http://www.soft-consulting.pe/software/detalle/3>.

Software Design Tools for Agile Teams, with UML, BPMN and More. [en línea] [sin fecha]. [Consulta: 9 febrero 2015]. Disponible en: <http://www.visual-paradigm.com/>.

TATO, L.B. y CARVAJAL, S.B. 2008. El informe de control interno en la Administración Pública. *Auditoría pública: revista de los Organos Autónomos de Control Externo*, no. 46, pp. 51–60.

TodoAgil.Pdf [en línea], [sin fecha]. S.l.: s.n. [Consulta: 15 abril 2015]. Disponible en: http://www.willydev.net/Willydev_old/Root/descargas/prev/TodoAgil.Pdf.

todo.PDF - xp.pdf [en línea], [sin fecha]. S.l.: s.n. [Consulta: 9 febrero 2015]. Disponible en: <https://met2unaj.files.wordpress.com/2014/09/xp.pdf>.

VALDÉS, M. de las M.F. y DANTE, G.P. 2008. Análisis conceptual de las principales interacciones entre la gestión de información, la gestión documental y la gestión del conocimiento. *Acimed: revista cubana de los profesionales de la información y la comunicación en salud*, vol. 18, no. 1, pp. 7.

Ventajas y Desventajas - Metodología XP. [en línea] [sin fecha]. [Consulta: 17 marzo 2015]. Disponible en: <https://sites.google.com/site/xpmetodologia/marco-teorico/ventajas>.

ANEXOS

Anexo 1: Entrevista

Nº	Preguntas
1	¿Con el cambio en las leyes de la contraloría general de la república, el sistema continuó cumpliendo con los requerimientos captados anteriormente en el MINCOM?
2	¿A raíz de estos cambios y con el tiempo que ha interactuado con el sistema ha surgido alguna funcionalidad que se pueda considerar de importancia para el trabajo en el proceso de gestión de control que aplica el MINCOM?
3	¿Se han detectado posibles errores que imposibiliten el manejo de la información?
4	¿La guía de autocontrol se adapta a cada una de las entidades pertenecientes al MINCOM?
5	¿Los administradores tienen alguna posibilidad de gestionar su propia guía de autocontrol?
6	¿Los aspectos o preguntas que conforman la guía de autocontrol siguen algún estándar conocido?

Anexo 2: Historias de usuario

Tabla 15. HU Permitir a los administradores consultar datos estadísticos en forma de gráfica.

Historia de Usuario	
No.: 3	Nombre: Permitir a los administradores consultar datos estadísticos en forma de gráfica.
Usuario: Administrador.	
Prioridad en el Negocio: Medio.	Nivel de Complejidad: Alto.

Estimación: 1	Iteración Asignada: 1
Descripción: El administrador tiene la posibilidad de consultar los datos estadísticos que brinda la guía después de terminada de llenar por el usuario, estos datos se muestran en forma de gráfica para una mejor comprensión y análisis del estado de la empresa a la cual se le aplico dicha guía de autocontrol.	
Información adicional (Observaciones):	

Tabla 16. HU Mostar automáticamente el control para subir evidencia, si el aspecto es marcado de no.

Historia de Usuario	
No.: 4	Nombre: Mostar automáticamente el control para subir evidencia, si el aspecto es marcado de no.
Usuario: Entidad.	
Prioridad en el Negocio: Alto.	Nivel de Complejidad: Medio.
Estimación: 1	Iteración Asignada: 1
Descripción: El usuario tiene la posibilidad de subir una evidencia del porque el aspecto fue marcado de no, en otro caso el control permanece oculto para el usuario.	
Información adicional (Observaciones): El usuario puede subir cualquier documento o foto que evidencie por qué no lo cumple.	

Tabla 17. HU Mostrar una evaluación de la guía después de ser llenada por el usuario.

Historia de Usuario	
No.: 5	Nombre: Mostrar una evaluación de la guía después de ser llenada por el usuario.
Usuario: Administrador.	
Prioridad en el Negocio: Medio.	Nivel de Complejidad: Medio.
Estimación: 1	Iteración Asignada: 2
Descripción: El administrador tiene la posibilidad de consultar una evaluación de la guía de autocontrol después de ser llenada por el usuario, esta evaluación le permitirá apreciar al administrador si la empresa se encuentra en el estado indicado por el MINCOM.	

Información adicional (Observaciones): Esta evaluación depende de un cálculo basado en el peso de cada aspecto para la empresa en cuestión.

Tabla 18. HU Asignar peso a cada aspecto de la guía de autocontrol.

Historia de Usuario	
No.: 6	Nombre: Asignar peso a cada aspecto de la guía de autocontrol.
Usuario: Entidad.	
Prioridad en el Negocio: Medio.	Nivel de Complejidad: Bajo.
Estimación: 0.5	Iteración Asignada: 2
Descripción: El administrador tiene la posibilidad de asignarle un peso a cada aspecto de la guía de autocontrol dependiendo de lo importante que sea para la empresa.	
Información adicional (Observaciones): Este peso para cada aspecto esta dado en tres valores (Alto, Medio, Bajo).	

Tabla 19. HU Crear guías nuevas a partir de aspectos de guías existentes o a partir de una completamente desde cero.

Historia de Usuario	
No.: 7	Nombre: Crear guías nuevas a partir de aspectos de guías existentes o a partir de una completamente desde cero.
Usuario: Administrador.	
Prioridad en el Negocio: Medio.	Nivel de Complejidad: Alto.
Estimación: 1.5	Iteración Asignada: 2
Descripción: El administrador tiene la posibilidad de crear o confeccionar una nueva guía de autocontrol con aspectos de una guía ya existente o de la que brinda el MINCOM.	
Información adicional (Observaciones):	

Tabla 20. HU Determinar el porcentaje de cumplimiento de la guía.

Historia de Usuario	
No.: 8	Nombre: Determinar el porcentaje de cumplimiento de la guía.

Usuario: Administrador.	
Prioridad en el Negocio: Bajo.	Nivel de Complejidad: Medio.
Estimación: 1	Iteración Asignada: 2
Descripción: El administrador tiene la posibilidad de ver el porcentaje de cumplimiento de la guía de autocontrol, este porcentaje es determinado por la cantidad de sí marcados por el usuario a la hora de llenarla.	
Información adicional (Observaciones):	

Tabla 21. HU Mostrar a los administradores un resumen de los resultados de la aplicación de la guía de autocontrol en las entidades subordinadas al MINCOM.

Historia de Usuario	
No.: 9	Nombre: Mostrar a los administradores un resumen de los resultados de la aplicación de la guía de autocontrol en las entidades subordinadas al MINCOM.
Usuario: Administrador.	
Prioridad en el Negocio: Bajo.	Nivel de Complejidad: Medio.
Estimación: 1	Iteración Asignada: 2
Descripción: El administrador tiene la posibilidad de ver una tabla con el resumen de los resultados estadísticos que brinda la aplicación después de ser aplicada la guía de autocontrol, este resumen es el que posteriormente será exportado y enviado al MINCOM.	
Información adicional (Observaciones):	

Tabla 22. HU Exportar el resultado de la guía después de terminada esta.

Historia de Usuario	
No.: 10	Nombre: Exportar el resultado de la guía después de terminada esta.
Usuario: Administrador.	
Prioridad en el Negocio: Alto.	Nivel de Complejidad: Alto.
Estimación: 1	Iteración Asignada: 2

Descripción: El administrador tiene la posibilidad de exportar el resumen de los datos estadísticos recogidos en la guía de autocontrol después de terminada por el usuario.
Información adicional (Observaciones): Este resultado será mandado después a la oficina de encriptado de la empresa en cuestión.

Tabla 23. HU Incorporar preguntas referentes a la gestión de información basadas en el marco de trabajo COBIT como aspectos de la guía de autocontrol.

Historia de Usuario	
No.: 11	Nombre: Incorporar preguntas referentes a la gestión de información basadas en el marco de trabajo COBIT como aspectos de la guía de autocontrol.
Usuario: Administrador.	
Prioridad en el Negocio: Alta.	Nivel de Complejidad: Bajo.
Estimación: 1.5	Iteración Asignada: 2
Descripción: El administrador debe confeccionar preguntas referentes a la gestión de información basándose en el marco de trabajo COBIT e incluirlas en el componente correspondiente a la gestión de información para que la guía de autocontrol	
Información adicional (Observaciones):	

Anexo 3: Tarjetas CRC

Tabla 24. Tarjeta CRC entidad_c.

Clase: entidad_c	
Responsabilidad	Colaboración
<ul style="list-style-type: none"> Es la clase encargada de ejecutar todas las acciones seleccionadas por el usuario referente a las entidades del ministerio. Crear una entidad con sus respectivos atributos. Elimina la entidad 	recursos layout_admin load input crear_entidad_form listar_entidades seguridad_helper

	form_validation ministerio_model
--	-------------------------------------

Tabla 25. Tarjeta CRC componente_c.

Clase: componente_c	
Responsabilidad	Colaboración
<ul style="list-style-type: none"> Ejecuta las acciones seleccionadas por el usuario referente a los componentes de la guía de autocontrol. Crear un componente especificándose el nombre. Eliminar componente 	recursos crear_componente_form layout_admin form_validation ministerio_model listar_componentes input load

Tabla 26. Tarjeta CRC subcomponente_c.

Clase: subcomponente_c	
Responsabilidad	Colaboración
<ul style="list-style-type: none"> Ejecuta las acciones seleccionadas por el usuario referente a los subcomponentes de la guía de autocontrol. Crear un subcomponente con sus respectivos atributos. Eliminar Subcomponentes 	recursos crear_subcomponente_form layout_admin load input listar_subcomponentes ministerio_model form Validation

Anexo 4: Descripción de las tareas de la primera iteración

Tabla 27. Tarea de ingeniería N°3 HU1.

Tarea	
Número: 3	Número de HU: 1
Nombre: Elaborar la seguridad del sitio	
Tipo de tarea: Desarrollo	Estimación: 1 día
Fecha inicio: 28 febrero 2015	Fecha fin: 1 marzo 2015
Responsable: Gustavo Rodríguez Méndez	
Descripción: Se implementa la seguridad básica de la aplicación.	

Tabla 28. Tarea de ingeniería N°4 HU1.

Tarea	
Número: 4	Número de HU: 1
Nombre: Implementar la funcionalidad evaluar componente	
Tipo de tarea: Desarrollo	Estimación: 5 días
Fecha inicio: 1 marzo 2015	Fecha fin: 5 marzo 2015
Responsable: Gustavo Rodríguez Méndez	
Descripción: Se implementa la seguridad básica de la aplicación mediante la clase seguridad_helper.	

Tabla 29. Tarea de ingeniería N°5 HU2.

Tarea	
Número: 5	Número de HU: 2
Nombre: Realizar cambios en la BD para poder asignar a los componentes y subcomponentes una entidad.	
Tipo de tarea: Desarrollo	Estimación: 2 días
Fecha inicio: 6 marzo 2015	Fecha fin: 8 marzo 2015
Responsable: David Ruisanchez Jaime	
Descripción: Se crean tablas nuevas en la BD con la relación componente entidad y subcomponente entidad.	

Tabla 30. Tarea de ingeniería N°6 HU2.

Tarea	
Número: 6	Número de HU: 2
Nombre: Realizar los cambios en la clase aspecto_c.	
Tipo de tarea: Desarrollo	Estimación: 2 días

Fecha inicio: 8 marzo 2015	Fecha fin: 10 marzo 2015
Responsable: David Ruisanchez Jaime	
Descripción: Se valida que cada subcomponente pertenezca al componente en cuestión.	

Tabla 31. Tarea de ingeniería N°7 HU3.

Tarea	
Número: 7	Número de HU: 3
Nombre: Realizar un estudio sobre la librería highcharts para el trabajo con las gráficas.	
Tipo de tarea: Investigación	Estimación: 1 día
Fecha inicio: 10 marzo 2015	Fecha fin: 11 marzo 2015
Responsable: David Ruisanchez Jaime	
Descripción: Se realizan las operaciones matemáticas necesarias para mostrar los datos al administrador.	

Tabla 32. Tarea de ingeniería N°8 HU3.

Tarea	
Número: 8	Número de HU: 3
Nombre: Realizar los cálculos correspondientes a las estadísticas que se requieran mostrar.	
Tipo de tarea: Mejora	Estimación: 2 días
Fecha inicio: 11 marzo 2015	Fecha fin: 13 marzo 2015
Responsable: David Ruisanchez Jaime	
Descripción: Se realizan las operaciones matemáticas necesarias para mostrar los datos al administrador.	

Tabla 33. Tarea de ingeniería N°9.

Tarea	
Número: 9	Número de HU: 3
Nombre: Mostrar los datos calculados en la interfaz de administración.	
Tipo de tarea: Mejora	Estimación: 4 días
Fecha inicio: 13 marzo 2015	Fecha fin: 17 marzo 2015
Responsable: David Ruisanchez Jaime	
Descripción: Se genera el código correspondiente y los parámetros necesarios para generar las gráficas.	

Tabla 34. Tarea de ingeniería N°10 HU4.

Tarea	
Número: 10	Número de HU: 4
Nombre: Verificar que los botones de evaluar los aspectos funcionen correctamente.	
Tipo de tarea: Corrección	Estimación: 1 días
Fecha inicio: 17 marzo 2015	Fecha fin: 18 marzo 2015
Responsable: Gustavo Rodríguez Méndez	
Descripción: Verificar que los botones de evaluar aspectos realicen las operaciones esperadas.	

Tabla 35. Tarea de ingeniería N°11 HU4.

Tarea	
Número: 11	Número de HU: 4
Nombre: Modificar la funcionalidad de subir evidencias.	
Tipo de tarea: Mejora	Estimación: 5 días
Fecha inicio: 18 marzo 2015	Fecha fin: 23 marzo 2015
Responsable: Gustavo Rodríguez Méndez	
Descripción: Realizar los cambios correspondientes en el código para que se muestren los resultados esperados.	

Anexo 5: Descripción de las tareas de la segunda iteración

Tabla 36. Tarea de ingeniería N°14 HU6.

Tarea	
Número: 14	Número de HU: 6
Nombre: Ponderar los aspectos mediante un peso dado por el usuario.	
Tipo de tarea: Desarrollo	Estimación: 2 días
Fecha inicio: 30 marzo 2015	Fecha fin: 1 abril 2015
Responsable: David Ruisanchez Jaime	
Descripción: Asignar a cada componente un peso determinado por el nivel de importancia que este tenga para la empresa.	

Tabla 37. Tarea de ingeniería N°15 HU7.

Tarea	
Número: 15	Número de HU: 7
Nombre: Permitir a cada entidad poder crear su propia guía.	

Tipo de tarea: Desarrollo	Estimación: 5 días
Fecha inicio: 1 abril 2015	Fecha fin: 5 abril 2015
Responsable: Gustavo Rodríguez Méndez	
Descripción: Cada entidad debe ser capaz de crear su propia guía.	

Tabla 38. Tarea de ingeniería N°16.

Tarea	
Número: 16	Número de HU: 7
Nombre: Permitir a cada entidad poder modificar su guía.	
Tipo de tarea: Desarrollo	Estimación: 4 días
Fecha inicio: 5 abril 2015	Fecha fin: 9 abril 2015
Responsable: Gustavo Rodríguez Méndez	
Descripción: Cada entidad debe ser capaz de modificar su guía.	

Tabla 39. Tarea de ingeniería N°17 HU7.

Tarea	
Número: 17	Número de HU: 7
Nombre: Permitir a cada entidad poder utilizar una guía ya existente.	
Tipo de tarea: Desarrollo	Estimación: 4 días
Fecha inicio: 5 abril 2015	Fecha fin: 9 abril 2015
Responsable: Gustavo Rodríguez Méndez	
Descripción: Cada entidad debe ser capaz de utilizar una guía existente o de otra entidad.	

Tabla 40. Tarea de ingeniería N°18 HU8.

Tarea	
Número: 18	Número de HU: 8
Nombre: Realizar los cálculos para determinar el porcentaje.	
Tipo de tarea: Desarrollo	Estimación: 3 días
Fecha inicio: 9 abril 2015	Fecha fin: 12 abril 2015
Responsable: David Ruisánchez Jaime	
Descripción: Capturar la cantidad de aspectos evaluados de si y no, además del total de los mismos para calcular el porcentaje de cumplimiento de la guía.	

Tabla 41. Tarea de ingeniería N°19 HU8.

Tarea	
Número: 19	Número de HU: 8
Nombre: Mostrar el porcentaje de cumplimiento en la tabla anteriormente generada.	
Tipo de tarea: Desarrollo	Estimación: 1 días
Fecha inicio: 9 abril 2015	Fecha fin: 10 abril 2015
Responsable: David Ruisánchez Jaime	
Descripción: Luego de calculado el porcentaje de cumplimiento, este debe mostrarse en la tabla resumen de la interfaz de administración.	

Tabla 42. Tarea de ingeniería N°20 HU9.

Tarea	
Número: 20	Número de HU: 9
Nombre: Recopilar los datos necesarios para confeccionar la tabla que se desea mostrar.	
Tipo de tarea: Desarrollo	Estimación: 5 días
Fecha inicio: 10 abril 2015	Fecha fin: 15 abril 2015
Responsable: David Ruisánchez Jaime	
Descripción: Tomar los valores de los cálculos anteriores: porcentaje, cantidad de si y no, cantidad de aspectos por peso, cantidad de aspectos por componente, total de aspectos de la guía y evidencias por componente.	

Tabla 43. Tarea de ingeniería N°21 HU9.

Tarea	
Número: 21	Número de HU: 9
Nombre: Generar la tabla a mostrar.	
Tipo de tarea: Desarrollo	Estimación: 2 días
Fecha inicio: 15 abril 2015	Fecha fin: 17 abril 2015
Responsable: David Ruisánchez Jaime	
Descripción: Crear la tabla donde se mostraran todos los datos obtenidos anteriormente.	

Tabla 44. Tarea de ingeniería N°22 HU10.

Tarea	
Número: 22	Número de HU: 10
Nombre: Exportar la tabla anteriormente generada.	
Tipo de tarea: Desarrollo	Estimación: 6 días

Fecha inicio: 17 abril 2015	Fecha fin: 23 abril 2015
Responsable: Gustavo Rodríguez Méndez	
Descripción: Después de generada la tabla esta debe ser exportada en forma de imagen.	

Tabla 45. Tarea de ingeniería N°23 HU11.

Tarea	
Número: 23	Número de HU: 11
Nombre: Realizar una investigación sobre el marco de trabajo COBIT.	
Tipo de tarea: Investigación	Estimación: 5 días
Fecha inicio: 23 abril 2015	Fecha fin: 28 abril 2015
Responsable: Gustavo Rodríguez Méndez	
Descripción: Se realiza una investigación sobre el marco de trabajo COBIT y sus versiones para determinar cuál era más factible utilizar en el trabajo.	

Tabla 46. Tarea de ingeniería N°24 HU11.

Tarea	
Número: 24	Número de HU: 11
Nombre: Redactar las preguntas correspondientes.	
Tipo de tarea: Investigación	Estimación: 4 días
Fecha inicio: 28 abril 2015	Fecha fin: 2 mayo 2015
Responsable: Gustavo Rodríguez Méndez	
Descripción: Luego de analizar COBIT se procede a redactar los aspectos correspondientes.	

Tabla 47. Tarea de ingeniería N°25 HU11.

Tarea	
Número: 25	Número de HU: 11
Nombre: Incorporar estas preguntas en la guía de autocontrol.	
Tipo de tarea: Desarrollo	Estimación: 1 días
Fecha inicio: 2 mayo 2015	Fecha fin: 3 mayo 2015
Responsable: Gustavo Rodríguez Méndez	

Descripción: Se agregan los aspectos redactados a la guía del Ministerio de Comunicaciones.

Anexo 6: Pruebas de aceptación

Tabla 48. Caso de prueba de aceptación HU2_P2.

Caso de prueba de aceptación	
Código: HU2_P2	Historia de Usuario: 2
Nombre: Validar que al agregar un aspecto el subcomponente al que pertenece se haga corresponder con su componente.	
Descripción: Prueba para verificar que un subcomponente no pueda estar repetido en componentes diferentes.	
Condiciones de Ejecución: El administrador eligió la opción de agregar un nuevo aspecto.	
Entrada/ Pasos de ejecución: Se entra el nombre del aspecto y se elige a que componente y subcomponente pertenece.	
Resultado Esperado: Solo se deben mostrar los subcomponentes pertenecientes al componente seleccionado por el administrador.	
Evaluación de la Prueba: Prueba satisfactoria.	

Tabla 49. Caso de prueba de aceptación HU3_P3.

Caso de prueba de aceptación	
Código: HU3_P3	Historia de Usuario: 3
Nombre: Mostrar gráficas a los administradores.	
Descripción: Se muestran los datos de realizar la guía a los administradores en forma de gráfica.	
Condiciones de Ejecución: El administrador eligió la opción de gráficas.	

Entrada/ Pasos de ejecución: La guía debe estar respondida para que se puedan mostrar los datos.
Resultado Esperado: Solo se deben mostrar las gráficas correspondientes con los resultados obtenidos en la aplicación de la guía.
Evaluación de la Prueba: Prueba satisfactoria.

Tabla 50. Caso de prueba de aceptación HU5_P4.

Caso de prueba de aceptación	
Código: HU5_P4	Historia de Usuario: 5
Nombre: Mostrar la evaluación correspondiente a la guía de autocontrol.	
Descripción: Se debe mostrar una evaluación de bien, mal o regular al administrador después de aplicar la guía basada en el peso de los aspectos y las respuestas a los mismos.	
Condiciones de Ejecución: Aunque no se haya realizado ninguna acción en la guía siempre se mostrara la evaluación, por defecto sale en mal.	
Entrada/ Pasos de ejecución: La guía debe estar respondida para que se puedan mostrar los datos reales correspondientes a la empresa.	
Resultado Esperado: Debe mostrar una etiqueta con la evaluación de la empresa.	
Evaluación de la Prueba: Prueba satisfactoria.	

Tabla 51. Caso de prueba de aceptación HU7_P6.

Caso de prueba de aceptación	
Código: HU7_P6	Historia de Usuario: 7
Nombre: Crear nuevas guías.	
Descripción: El usuario administrador debe tener la posibilidad de crear una nueva guía para su entidad.	
Condiciones de Ejecución: El usuario debe estar autenticado como administrador.	

Entrada/ Pasos de ejecución: El administrador debe acceder a la opción mostrar guía de la entidad.
Resultado Esperado: El administrador debe poder eliminar completamente la guía por defecto y crear una completamente desde cero, copiar una guía de otra entidad, o modificar la suya propia agregando nuevos aspectos, componentes y/o subcomponentes.
Evaluación de la Prueba: Prueba satisfactoria.

Tabla 52. Caso de prueba de aceptación HU9_P7.

Caso de prueba de aceptación	
Código: HU9_P7	Historia de Usuario: 9
Nombre: Mostrar los resultados de la aplicación de la guía a los administradores.	
Descripción: Se debe mostrar una tabla con los datos de la aplicación de la guía de autocontrol a los administradores.	
Condiciones de Ejecución: Siempre se mostrará el resumen aunque no se haya respondido la guía, solo que todos los datos estarán vacíos.	
Entrada/ Pasos de ejecución: El usuario debe estar autenticado como administrador.	
Resultado Esperado: Se debe mostrar la tabla con los resultados de la guía.	
Evaluación de la Prueba: Prueba satisfactoria.	

Tabla 53. Caso de prueba de aceptación HU10_P8.

Caso de prueba de aceptación	
Código: HU10_P8	Historia de Usuario: 10
Nombre: Exportar el resultado de la guía	
Descripción: Se debe exportar en una imagen la guía.	
Condiciones de Ejecución: el administrador elige la opción de salvar resumen como imagen.	
Entrada/ Pasos de ejecución: El usuario debe estar autenticado como administrador.	

Resultado Esperado: se abrirá una nueva pestaña en el navegador con la imagen y se elige donde guardarla.

Evaluación de la Prueba: Prueba satisfactoria.