

Universidad de las Ciencias Informáticas

Facultad 6



“Sistema informático para determinar el índice de control organizacional en la Facultad 6 de la Universidad de las Ciencias Informáticas.”

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

Autor:

Sergio Morales Cabrera

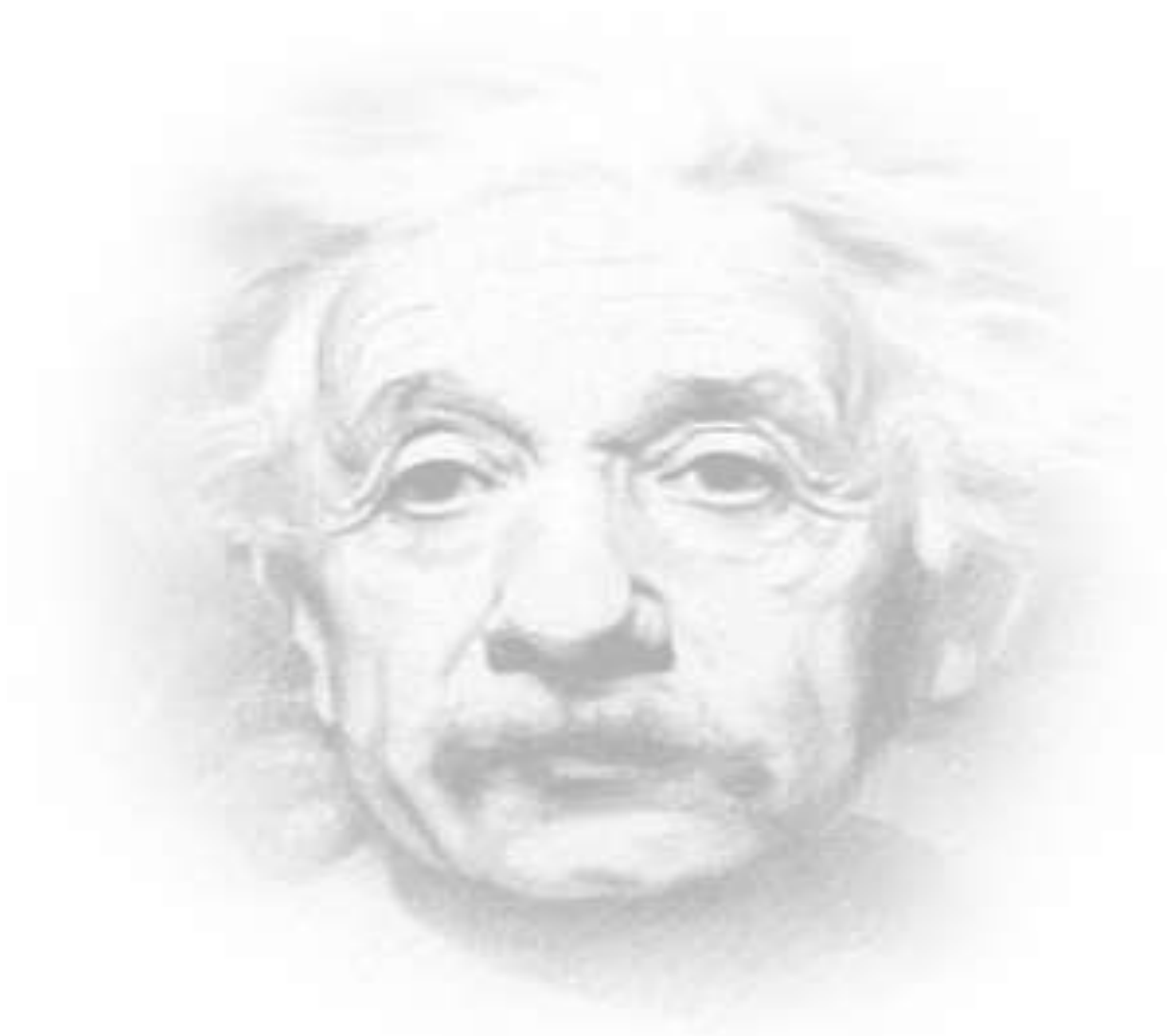
Tutores:

MSc. Omar Mar Cornelio

Ing. Bárbara Bron Fonseca

La Habana, junio de 2015

“Año 57 de la Revolución”



“No veas el estudio como algo obligatorio, sino como una forma de renacer el mundo del conocimiento”.

Albert Einstein

DECLARACIÓN DE AUTORÍA

Declaro ser autor de la presente tesis y reconozco a la Universidad de las Ciencias Informáticas (UCI) los derechos patrimoniales de la misma, con carácter exclusivo.

Para que así conste se firma la presente a los ____ días del mes de _____ del año 2015.

Firma del autor

Firma del tutor

Firma del tutor

Datos de contacto

Tutor:

MsC. Omar Mar Cornelio: Vicedecano Administrativo, vinculado a la docencia, en el curso regular diurno en la disciplina de Sistemas Digitales. Certificado como especialista en calidad para revisiones de Software; presenta experiencia como tribunal de tesis, tutor y oponente en pregrado. Actualmente realiza el doctorado de Automática en la Universidad Central de Las Villas.

Correo electrónico: omarmar@uci.cu

Tutor:

Ing. Bárbara Bron Fonseca: Graduada en el año 2012 de Ingeniera en Ciencias Informáticas en la Universidad de las Ciencias Informáticas. Se desempeña actualmente como Especialista B en Informática en la Empresa de Tecnologías para la Defensa XETID. Posee publicaciones en revistas científicas y memorias de eventos. Certificado como especialista en calidad para revisiones de Software. Presenta experiencia como tutor y oponente en pregrado.

Correo electrónico: bbron@xetid.cu

Agradecimientos

A la memoria de mi abuelo.

- *A mi mamá, mi tía, mi abuela, a mi hermano, mi madrina y mis primos.*
- *A los profesores que siempre me han estado ayudando.*
- *A mis tutores.*
- *A la gente del barrio, en especial a mi vecina.*
- *A todas las amistades que siempre me han apoyado en especial al grupo 6503.*

Resumen

Con el objetivo de que todas las entidades estatales en Cuba, ejecuten correctamente el control interno, la Contraloría General de la República de Cuba, en su Ley 107/09, proviene de un grupo de acciones para que dichas instituciones logren adecuar, dosificar y aplicar la Guía de Autocontrol de la República en correspondencia con la Norma del Sistema de Control Interno aprobada mediante la Resolución No. 60/20. En la Facultad 6 de la Universidad de las Ciencias Informática (UCI) se aplica a través de una lista de verificación manual para la comprobación documental y física de las operaciones individuales, pero estos procesos no presentan un análisis de iteraciones previas, por lo que no se maneja una trazabilidad de los indicadores defectuosos, dificultando la toma de decisiones y la priorización de ejecuciones. El presente trabajo describe una solución a dicha problemática, utilizando la metodología de desarrollo Open Up para la implementación de un sistema Web, codificado con JavaScript y PHP 5.5.11, con *framework* de desarrollo Symfony 2.3.9 y gestor de base de datos PostgreSQL 9.3.

Palabras Claves: *Guía de Autocontrol de la República, Sistema de Control Interno.*

Abstract

In order that all state institutions in Cuba, correctly implement the internal control, the Comptroller General of the Republic of Cuba, in his law 107/09, it comes from a group of actions to achieve those institutions adapt, dispensing and application Self Guide of the Republic in line with the Standard Internal Control System approved by Resolution No. 60/20. Faculty 6 University Computing Sciences (UCI) is applied through a list of manual check for document and physical check of individual operations, but these processes do not provide an analysis of previous iterations, so no traceability of defective handles indicators, hampering the decision making and prioritization of executions. This paper describes a solution to this problem using Open Up development methodology for the implementation of a Web system encoded with JavaScript and PHP 5.5.11, with development framework Symfony manager 2.3.9 and PostgreSQL 9.3 database.

Keywords: *Guide for Self of the Republic, System of Organizational Control.*

Índice

Introducción	1
Capítulo 1: Fundamentos teóricos	6
Introducción.....	6
1.1. Conceptos asociados al sistema para determinar el índice de control interno	6
1.2. Sistemas de información para el Control Interno.....	9
1.2.1. MEYCOR coso AG, Versión: 2.0 / Año: 2009.	9
1.3. Tecnologías, herramientas y metodologías a considerar	10
1.3.1. Metodología de Desarrollo de Software	11
1.3.2. Lenguajes de modelado.....	11
1.3.3. Herramientas CASE de modelado	12
1.3.4. Lenguajes de Programación	13
1.3.5. Framework.....	15
1.3.6. IDE de desarrollo	15
1.3.7. Sistemas Gestores de Base de Datos (SGBD)	16
1.3.8. Servidor Web.....	18
Conclusiones del capítulo.....	18
Capítulo 2: Descripción del sistema	19
Introducción.....	19
2.1. Propuesta de solución	19
2.2. Modelo de Negocio.....	20
2.2.1. Actores del negocio	20
2.2.2. Trabajadores del negocio	20
2.3. Diagrama de casos de uso del negocio	21
2.4. Diagrama de actividades del modelo de negocio	22
2.5. Descripción de los casos de usos del negocio.....	23
2.6. Extracción de Requisitos	25
2.6.1. Requisitos funcionales.....	25
2.6.2. Requisitos no funcionales	27

2.7. Modelo de casos de uso	29
2.7.1. Actores del sistema	29
2.8. Diagrama de casos de uso	30
2.9. Descripción de casos de uso	30
Conclusiones del capítulo.....	35
Capítulo 3: Análisis y diseño del sistema	36
Introducción.....	36
3.1. Modelo del diseño.....	36
3.2. Descripción de patrones arquitectónicos.....	36
3.2.1. Patrones arquitectónicos	36
3.2.2. Patrón arquitectónico Modelo – Vista – Controlador	37
3.3. Patrones de diseño	39
3.3.1. Patrones GRASP.....	39
3.3.2. Patrones Gof	39
3.4. Diagrama entidad relación	40
3.5. Diagrama de clases del diseño	40
3.5.1. Diagrama de clases del diseño	41
3.6. Diagramas de interacción	41
3.7. Diagrama de despliegue	42
3.8. Diagrama de componentes	43
Conclusiones del capítulo.....	44
Capítulo 4: Implementación y Análisis de los Resultados.....	45
Introducción.....	45
4.1. Código fuente	45
4.1.1. Estándares de codificación	45
4.1.2. Nombres de clases y métodos.....	45
4.1.3. Estructura	46
4.2. Pantallas principales de la aplicación.....	47
4.3. Niveles de Prueba	48
4.4. Tipos de Prueba	49

4.5. Métodos de Prueba	49
4.5.1. Pruebas funcionales de caja negra	49
4.6. Pruebas de seguridad.....	52
4.7. Demostración de la hipótesis	53
Conclusiones del capítulo.....	56
Conclusiones generales	56
Recomendaciones.....	57
Referencias Bibliográficas.....	58
Bibliografía.....	60
Anexos.....	63
Anexo 1: Entrevista realizada para identificar errores en el proceso de aplicación de la Guía de Autocontrol de la República en la Facultad 6 de la Universidad de las Ciencias Informáticas.	63
Anexo 2: Cuestionario de la entrevista para validar la hipótesis investigativa.	65

Índice de figuras

Figura 1: Propuesta de solución.....	20
Figura 2: Caso de uso del negocio.....	22
Figura 3: Diagrama de actividades: Determinar índice de control interno.....	22
Figura 4: Diagrama de actividades: Aplicar Guía de Autocontrol de la República.	23
Figura 5: Diagrama de casos de uso del sistema.....	30
Figura 6: Patrón arquitectónico Modelo Vista Controlador.	38
Figura 7: Diagrama Entidad Relación.....	40
Figura 8: Diagrama de clases del diseño: Gestionar criterios.....	41
Figura 9: Diagrama de secuencia: Otorgar importancia.	42
Figura 10: Diagrama de despliegue.	43
Figura 11: Diagrama de componentes: MainBundle.....	44
Figura 12: Estándar de codificación CamelCase.....	46
Figura 13: Estándar de codificación UpperCamelCase.	46
Figura 14: Página principal de administración: Usuarios.....	47
Figura 15: Página principal de la administración: Componentes.	48
Figura 16: Iteraciones de las pruebas funcionales.	52
Figura 17: Tiempo de cálculo del índice de control.	56

Índice de tablas

Tabla 1: Características fundamentales de la solución existente.	9
Tabla 2: Actores del negocio.....	20
Tabla 3: Trabajadores del negocio.....	21
Tabla 4: Caso de uso del negocio: Aplicar Guía de Autocontrol de la República.....	23

Tabla 5: Caso de uso: Determinar el índice de control interno.	24
Tabla 6: Actores del sistema.	29
Tabla 7: Descripción de caso de uso: Gestionar criterios.	30
Tabla 8: Caso de prueba: Otorgar importancia.	49
Tabla 9: Caso de prueba: Evaluar criterio.	50
Tabla 10: Caso de prueba: Gestionar componentes.	51
Tabla 11: Validación de la hipótesis.	53

Introducción

Con el desarrollo de la actividad cognoscitiva del hombre, los cambios en las fuerzas productivas, los avances en los modos de producción y del momento histórico vigente, las Tecnologías de la Información y las Comunicaciones (TIC) se han introducido de forma masiva en todas las aristas sociales. Redefiniendo de esa forma los paradigmas de trabajos tradicionales, a partir de la integración tecnológica.

En la actualidad la actividad productiva de una organización requiere en gran medida de una serie de factores internos. Los mismos permiten a través de procedimientos previamente definidos y actividades concretas, la correcta elaboración de un plan de trabajo que facilite la toma de decisiones. Donde este último es considerado por naturaleza la esencia fundamental del desarrollo y del conocimiento y de esa manera se le considera la razón de ser de las empresas, organismos e instituciones que presentan una productividad de carácter metódico y sistemático. Cada decisión es el resultado de un proceso dinámico, del pensamiento y deliberación, que recibe influencias de muchas fuerzas, por lo que la toma de decisiones se basa en el trabajo grupal, la relación entre varias disciplinas, la efectividad de los procesos comunicativos y el flujo de trabajo (Salinas R, 2004). Cuba es una nación que presenta desde el año 1989 una reestructuración de los procesos administrativos y de gestión control interno para enfocar la actividad productiva hacia un futuro seguro, ajustado a las condiciones reales que presentan las normas internacionales y poder vincular la ciencia con las actividades laborales.

La organización de las tareas productivas, así como su control, es una tarea decisiva en el plano económico de las organizaciones cubanas. Reflejada en la actualización del modelo económico en los Lineamientos, donde la Ley de la Planificación hace varios años tiene un carácter jerárquico, jurídico, rector y primario; por lo que, en agosto de 2009 la Asamblea Nacional del Poder Popular, crea la Contraloría General de la República de Cuba. La cual presenta entre sus funciones establecidas, según lo señalado en el artículo (Gobierno de la República de Cuba, 2009), normar, supervisar y evaluar el Sistema de Control Interno e informar o manifestar su mejoramiento y perfeccionamiento continuo para optimizar los procesos internos.

Para evaluar las tendencias de control que poseen las organizaciones, la Contraloría General de la República de Cuba, establece la Guía de Autocontrol de la República. Donde cada entidad, en períodos de tiempo previamente definidos, revisa el comportamiento de dichos parámetros con el fin de identificar los indicadores críticos de una organización. De esa forma permiten establecer una sistematicidad en el cumplimiento del trabajo, ya que el paso de los años presenta un común denominador, lo que no se

controla con efectividad, no se cumple o se ejecuta superficialmente, (Castro Raúl, 2009). Además es muy difícil lograr una alta cohesión en los elementos que intervienen en un sistema de control interno.

En la Universidad de las Ciencias Informáticas (UCI) se cuenta con un ambiente muy dinámico que necesita ser supervisado por la Dirección de Supervisión y Control (DSC), que tiene como misión evaluar y supervisar la implementación del Sistema de Control Interno (SCI), con profesionales de que garanticen el correcto funcionamiento de las actividades laborales y corrijan las que se ejecutan incorrectamente. La facultad número 6 de la UCI, aplica la Guía de Autocontrol de la República para obtener un valor cuantitativo del funcionamiento de los procesos internos, basándose en el cálculo del índice organizacional y posteriormente determinar qué factores internos funcionan correctamente, sin embargo, el autor en su práctica pudo identificar mediante las entrevistas aplicadas a técnicos, especialistas y directivos las siguientes irregularidades:

- Cuando la facultad 6 posee muchos elementos por corregir no se logran identificar los elementos críticos en los cuales fortalecer las funciones de control.
- No se realizan todas las iteraciones de autocontrol previstas debido a la presencia de cálculos engorrosos y complejos.
- El trabajo se realiza de forma manual, por lo que no se logra sistematicidad en el control para lograr cumplir con todos sus elementos.
- No se vincula el desarrollo de las nuevas tecnologías en la agilización de los procesos de autocontrol.
- No se cuenta con una trazabilidad de los indicadores insatisfactorios detectados en iteraciones previas.
- La aplicación de la Guía de Autocontrol de la República requiere enormes recursos humanos y de tiempo para su aplicación, afectando los periodos de entrega definidos.
- Debido al trabajo manual, la determinación del Índice del control organizacional es vulnerable ante errores de cálculos y resultados irreales.

Atendiendo a la situación antes planteada se define como **problema a resolver**: ¿Cómo contribuir a la trazabilidad de los datos y a la disminución del margen de errores y de tiempo en el cálculo del índice de control organizacional, en la aplicación de la Guía de Autocontrol de la República en la Facultad 6 de la Universidad de las Ciencias Informáticas?

Para dar solución al problema identificado se traza como **objetivo general**: Desarrollar un sistema informático que contribuya a la trazabilidad de los datos y a la disminución del margen de errores y de tiempo en el cálculo del índice de control organizacional, en la aplicación de la Guía de Autocontrol de la República en la Facultad 6 de la Universidad de las Ciencias Informáticas.

Una vez planteado el objetivo general se determina como **objeto de estudio**: El Sistema de Control Interno en la Universidad de las Ciencias Informáticas, enmarcado en el **campo de acción**: La aplicación de la Guía de Autocontrol de la República en la Facultad 6 de la Universidad de las Ciencias Informáticas.

Hipótesis: Con el desarrollo de una aplicación informática, se contribuirá a la trazabilidad de los datos, se eliminarán errores de cálculo y se disminuirá el tiempo del cálculo del índice de control organizacional, en la aplicación de la Guía de Autocontrol de la República en la Facultad 6 de la Universidad de las Ciencias Informáticas.

Para guiar el proceso investigativo del presente trabajo se definen las siguientes **tareas investigativas**:

- Estudio científico sobre las soluciones existentes.
- Estudio de las herramientas y tecnologías que serán utilizadas en el desarrollo del sistema.
- Realización del diseño de la solución propuesta.
- Desarrollo de la propuesta de solución.

Para obtener los conocimientos necesarios con la finalidad de hacer posible el cumplimiento del objetivo trazado en el trabajo, se llevó a cabo una investigación en la que se utilizaron los siguientes métodos de la investigación científica:

Métodos empíricos usados para poder determinar el grado de complejidad que adquiere el problema y para identificar si quedaron o no satisfechas todas las necesidades previstas a través de:

- **Entrevista**: Aplicada a directivos, técnicos y especialistas de la facultad 6 de la UCI, con el objetivo de obtener información sobre el proceso de aplicación de la Guía de Autocontrol de la República, como parte del cumplimiento de lo normado en el Sistema de Control Interno y a partir de las respuestas obtenidas, identificar algunas de las irregularidades y deficiencias iniciales que se cometen en dicho proceso (ver anexo 1 y 2).

Métodos teóricos usados para estudiar las características del objeto de investigación que no son observables directamente, para apoyar el análisis y la síntesis y su relación con otros fenómenos:

- **Analítico-Sintético:** Aplicado por el autor para descomponer el problema de investigación en elementos por separados y profundizar en el estudio de cada uno de ellos, para luego sintetizarlos en la solución propuesta. Entre los elementos identificados se encuentran técnicas multicriterio y el consenso de expertos aplicadas a la modelación matemática para garantizar control interno, función de los sistemas de información en las actividades laborales para la gestión del control y el manejo de los indicadores críticos que son básicos en las organizaciones nacionales, de manera que permita construir correctamente el marco teórico de la investigación.
- **Histórico-Lógico:** Aplicado por el autor para constatar teóricamente cómo han evolucionado los sistemas de información en el control de las actividades laborales, los cuales varían a partir de las leyes y cambios aplicados en dependencia de las normas internacionales y para profundizar en los lineamientos establecidos en el Sistema de Control Interno de forma tal que permita un análisis real a la hora de identificar los problemas que existen en la aplicación de la Guía de Autocontrol de la República.

El presente trabajo de diploma se ha estructurado de la siguiente forma:

Capítulo 1: Fundamentación teórica

Se especifican los conceptos que son tratados a lo largo del documento y que tienen vital importancia para la comprensión del mismo. Se realiza un estudio de los sistemas informáticos para determinar el índice de control de una organización, utilizadas hoy en Cuba y en el mundo y se describen y justifican las diferentes herramientas, metodologías y lenguajes de programación a utilizar durante la construcción del sistema.

Capítulo 2: Características del Sistema

Se describe las características del sistema a desarrollar. Se define el modelo de dominio, los requisitos funcionales y no funcionales del sistema para comprender mejor su funcionamiento, así como, los actores, diagrama de casos de uso del sistema y los patrones de casos de uso utilizados.

Capítulo 3: Análisis y Diseño del Sistema

Está relacionado con el diseño del sistema. Se describen los patrones de diseño empleados. Se presentan los diagramas de clases del diseño, el diagrama entidad relación, el diagrama de componentes y el diagrama de despliegue del sistema con el objetivo de tener una idea concreta, de cómo quedará.

Capítulo 4: Implementación y Análisis de los Resultados

Está enfocado a la fase de implementación para dar solución a los requerimientos funcionales identificados. Se describen los estándares de codificación, el tratamiento de errores para darle solución al sistema propuesto y se demuestra la hipótesis investigativa.

Capítulo 1: Fundamentos teóricos

Introducción

El presente capítulo introduce los conceptos básicos relacionados con el funcionamiento de los sistemas que se encargan de gestionar las actividades de control. Son analizados los sistemas profesionales de corte nacional e internacional, que son usados actualmente y son seleccionadas las herramientas y metodologías a utilizar en el desarrollo del software, justificando el uso de cada una de ellas.

1.1. Conceptos asociados al sistema para determinar el índice de control interno

Gestión de la información

La gestión de la información representa la forma de organizar, evaluar, presentar y comparar los datos en un determinado contexto, controlando su calidad, de manera que esta sea veraz, oportuna, significativa, exacta y útil; y que esta información esté disponible en el momento que se le necesite. Ella se encamina al manejo de la información, documentos, metodologías, informes, publicaciones, soportes y flujos en función de los objetivos estratégicos de una organización (Ledo, 2012).

La gestión de la información se vincula con la generación y la aplicación de estrategias, el establecimiento de políticas, así como con el desarrollo de una cultura organizacional y social dirigida al uso racional, efectivo y eficiente de la información en función de los objetivos y metas trazadas en materia de desempeño y de calidad.

Un Sistema de Información está constituido por los métodos y procedimientos establecidos para registrar, procesar, resumir e informar sobre las operaciones de una entidad. La calidad de la información que brinda el sistema afecta la capacidad de los directivos y ejecutivos para adoptar decisiones adecuadas que permitan controlar las actividades de la entidad (Contraloría General de la República, 2009).

Gestión de la información en el Sistema de Control Interno

Un Sistema de Control Interno se implanta con el fin de detectar, en el plazo deseado, cualquier desviación respecto a los objetivos de rentabilidad establecidos por la empresa o entidad y de limitar las sorpresas. El Sistema de Control Interno está entrelazado con las actividades operativas de la entidad y existe por razones empresariales fundamentales. Es más efectivo cuando los controles se incorporan a la

infraestructura de la sociedad y forman parte de la esencia de la empresa. Mediante los controles se fomenta la calidad y las iniciativas de delegación de poderes. Se evitan gastos innecesarios y se permite una respuesta rápida ante las circunstancias cambiantes.

En Cuba, la dirección del Gobierno a través de los ministerios de Finanzas y Precios y de Auditoría y Control, en unión de otros órganos y organismos del Estado y más recientemente de la Contraloría General de la República, han desarrollado un constante y sostenido esfuerzo por consolidar el control interno en las diferentes entidades, de modo que este constituye una preocupación de las organizaciones, en mayor o menor grado, con diferentes enfoques y terminologías, lo cual ha permitido que progresivamente se hayan planteado distintas concepciones acerca de esa actividad. Así como también de los principios y elementos que deben conocerse e implantarse en la entidad cubana actual. En el 2003 puede considerarse como el año del control interno en todo el país, puesto que los ministerios de Finanzas y Precios y Auditoría y Control emitieron disposiciones que obligan a garantizar el control de todos los recursos que el Estado pone a disposición de las entidades para la creación de bienes y la satisfacción de las necesidades de la sociedad.

La Contraloría General de la República norma el marco general conceptual que el Sistema de Control Interno está compuesto por los jefes de los órganos, organismos y entidades nacionales y los dirigentes a ellos subordinados. Además diseñan e implementan auditorías de acuerdo con los principios establecidos en la ley y con las particularidades de su sistema (Contraloría General de la República, 2009). Las direcciones y departamentos de la Contraloría General de la República tienen en su esfera la competencia entre las funciones y atribuciones específicas, las siguientes:

- Asesorar el diseño e implementación del Sistema de Control Interno y comprobar su efectividad.
- Ejecutar la auditoría interna en las estructuras organizativas de la Contraloría General de la República, de acuerdo con el plan aprobado.
- Cualquier otra que deba entenderse referida a su actuación.

Es evidente la importancia que reviste para la recuperación de la economía, el control de los recursos materiales, financieros y humanos. La utilización del control interno en las entidades permite que se reduzcan en gran magnitud las ilegalidades y corrupción dentro del marco laboral. Con la aplicación del Sistema de Control Interno en todos sus aspectos se podrá obtener resultados de eficiencia, vigencia y economía en la gestión empresarial.

El control se define como conjunto de acciones que se ejecutan para comprobar la aplicación de las políticas del Estado, así como del cumplimiento del plan de la economía y su presupuesto (Folleto SistemaCI, Resolución No. 60/11, 2009). El control interno representa 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 (Folleto SistemaCI, Artículo 3, 2009).

La Contraloría General de la República como norma de control interno establece que las organizaciones deben realizar una actualización del SCI con una frecuencia no mayor a seis meses denominado autocontrol. El autocontrol es la obligación que tienen los directivos superiores, directivos, ejecutivos y funcionarios de los órganos, organismos, organizaciones y demás entidades, estas últimas con independencia del tipo de propiedad y forma de organización, de autoevaluar su gestión de manera permanente; y cuando proceda, elaborar un plan para corregir las fallas e insuficiencias, adoptar las medidas administrativas que correspondan y dar seguimiento al mismo en el órgano colegiado de dirección, comunicar sus resultados al nivel superior y rendir cuenta a los trabajadores (Folleto SistemaCI, Artículo 6, 2009).

Como instrumento de autocontrol, la Contraloría General de la República ha creado la Guía de Autocontrol de la República, la cual posee un conjunto de componentes e indicadores evaluativos. Un componente del SCI se entiende por elementos que de manera integrada ejercen una función específica para el cumplimiento de los objetivos del Sistema de Control Interno; se estructuran en normas de carácter general (Folleto SistemaCI, Resolución No. 60/11, 2009).

El control interno consta de cinco componentes interrelacionados que se derivan de la forma, cómo la administración maneja el negocio, y están integrados a los procesos administrativos. Los componentes pueden considerarse como un conjunto de normas que son utilizadas para medir el control interno y determinar su efectividad.

El SCI es objeto de evaluación para conocer la eficacia de su funcionamiento y permitir su retroalimentación y mejora continua; la evaluación comprende un conjunto de actividades de control incorporadas a los procesos, actividades y operaciones que se llevan a cabo mediante el seguimiento de sus resultados.

Cuando se detecte alguna deficiencia en el funcionamiento del SCI, los jefes de los órganos, organismos, organizaciones y demás entidades emprenden oportunamente las acciones preventivas y correctivas pertinentes para su fortalecimiento, de conformidad con los objetivos y recursos disponibles. También verifican de manera sistemática, los avances y logros en la implementación de las acciones adoptadas. Los órganos, organismos, organizaciones y demás entidades mediante la aplicación del autocontrol sistemático garantizan su función de control, adecuan, actualizan y gradúan la aplicación de la Guía de Autocontrol de la República emitida por la Contraloría General de la República en correspondencia con su actividad y sus características (Folleto SistemaCI, Resolución No. 60/11, 2009).

En la actualidad, el proceso de aplicación de la Guía de Autocontrol de la República es realizado manualmente, ya que no se utilizan herramientas digitales para la gestión y análisis de datos, aunque en el país constituye una de las principales tareas en las organizaciones independientemente de su calidad e importancia en la actividad laboral, el mismo es más efectivo cuando los controles se incorporan a la infraestructura de la sociedad y forman parte de la esencia de la empresa. Mediante los controles se fomenta la calidad y las iniciativas de delegación de poderes; es decir se evitan gastos innecesarios y se permite una respuesta rápida ante las circunstancias cambiantes (Solórzano P., 2007).

1.2. Sistemas de información para el Control Interno

A continuación se expone un análisis sobre el sistema MEYCOR que presenta como principal característica la gestión de actividades de control, desplegado fuera del territorio nacional.

1.2.1. MEYCOR coso AG, Versión: 2.0 / Año: 2009.

Durante la investigación se descubrió la existencia de una herramienta en Uruguay para la implementación del control interno basado en informes de gestión de actividades: El software MEYCOR COSO AG ha sido desarrollado por DATASEC. El producto permite realizar una evaluación del control interno, evaluando riesgos y auditando dichas evaluaciones. Está compuesto por dos módulos: MEYCOR COSO AG – EVAL y MEYCOR COSOAG – AUDIT. El módulo MEYCOR COSO AG - EVAL constituye la parte de recopilación de información de la organización, así como también de la evaluación del control interno y los riesgos. El producto permite ingresar el organigrama de la organización, sus procesos de negocio, sus objetivos de negocio y las tareas que se realizan para llevar a cabo cada proceso.

Tabla 1: Características fundamentales de la solución existente.

CARACTERÍSTICAS	MEYCOR COSO AG
Posibilidad de operaciones matemáticas.	No posee
Seguimiento a los datos registrados.	Si posee
Integración con otras herramientas.	Si posee
Incorporación de funcionalidades agregadas.	Si posee
Posibilidad de registrar datos, planillas o usuarios.	No posee

La herramienta descrita anteriormente se enfoca a un estándar abierto de interoperabilidad semántica para el diseño del Sistema de Información Integrado para la gestión de información asociada a los datos y metadatos asociados al cuestionario del estándar COSO, específicamente MEYCOR COSO AG. Pero resalta la necesidad de manejar información de control para el correcto funcionamiento de las entidades.

Después de realizar un análisis del sistema expuesto con anterioridad, se puede decir que en su generalidad, no cumple con la necesidad que se plantea en la problemática. Es decir, la información que gestiona este sistema, no coincide con la información que se desea controlar, no manejan los mismos tipos de datos en cada uno de ellos y no presentan una modelación matemática para garantizar el control interno.

1.3. Tecnologías, herramientas y metodologías a considerar

Las tecnologías, herramientas y metodologías escogidas para el desarrollo del sistema se exponen en el siguiente subtema, para lo cual se analizaron las distintas alternativas y las necesidades del proyecto. El principal elemento que se tuvo en consideración fue que las herramientas debían ser compatibles, basadas en software libre, gratuitas y sin restricciones de uso.

1.3.1. Metodología de Desarrollo de Software

Una metodología es una guía para el desarrollo del software; que hace posible que todo el personal de un proyecto se vincule y pueda entenderse, estas son un conjunto de procedimientos, técnicas y ayuda a la documentación para el desarrollo de productos software, en la que se indican, paso a paso, todas las actividades a realizar para lograr informatizar el proceso deseado; muestran qué personas deben participar en el desarrollo de las actividades y el papel que desempeñan. Además, detallan la información que se debe producir como resultado de una actividad y la información necesaria para comenzarla.

Metodología de Desarrollo de Software (Open Up)

Open UP es una metodología de desarrollo de software, basada en RUP (Rational Unified Process), que contiene el conjunto mínimo de prácticas que ayudan a un equipo de desarrollo de software a realizar un producto de alta calidad, de una forma eficiente. Open UP, es un proceso unificado, iterativo e incremental, que se centra en el desarrollo colaborativo de software para generar sistemas de calidad. Los elementos que forman Open Up son tareas, disciplinas, artefactos y procesos. El ciclo de vida de un proyecto, según la metodología Open UP, permite que los integrantes del equipo de desarrollo aporten con micro-incrementos, que pueden ser el resultado del trabajo de unas pocas horas o unos pocos días. El progreso se puede visualizar diariamente, ya que la aplicación va evolucionando en función de este micro-incremento. El objetivo de Open UP es ayudar al equipo de desarrollo, a lo largo de todo el ciclo de vida de las iteraciones, para que sea capaz de añadir valor de negocio a los clientes, de una forma predecible, con la entrega de un software operativo y funcional al final de cada iteración. El ciclo de vida del proyecto provee a los clientes de: una visión del proyecto, transparencia y los medios para que controlen la financiación, el riesgo, el ámbito, el valor de retorno esperado (Infante A., 2013).

Todo proyecto en Open UP consta de cuatro fases: inicio, elaboración, construcción y transición. Cada una de estas fases se divide a su vez en iteraciones (OpenUp Basic. Eclipse Foundation, 2011)

Metodología de Desarrollo de Software seleccionado

Se escoge como metodología de desarrollo de software a Open Up, que permite al sistema integrarse a requerimientos cambiantes que se pueden dar durante el desarrollo del proyecto, es una metodología ágil, que presenta documentación, muy favorable para proyectos que presentan pocos integrantes y es aplicable a sistemas de bajo costo.

1.3.2. Lenguajes de modelado

El lenguaje de modelado pretende dar apoyo a la mayoría de los procesos de desarrollo de software, son desarrollados con el esfuerzo de simplificar y consolidar el gran número de métodos de desarrollo que han surgido. (Jacobson, 2000).

Lenguaje Unificado de Modelado 2.0 (UML)

UML es un lenguaje para visualizar, especificar, construir y documentar los artefactos de un sistema. Está compuesto por diversos elementos gráficos que se combinan para conformar diagramas y permite la modelación de sistemas con tecnología orientada a objetos. Este lenguaje es fácil de aprender y usar, conforma la colección de las mejores técnicas de ingeniería que han probado ser un éxito en el modelado de sistemas grandes y complejos. Con el uso de UML se facilita la asimilación y entendimiento del contenido, se minimiza el tiempo invertido en el desarrollo del sistema y la documentación del proyecto se confecciona de una forma ordenada.

Lenguaje de modelado seleccionado

Se escoge UML 2.0 por ser un lenguaje reconocido y probado a nivel internacional además de poseer disimiles ventajas en cuanto a su uso y facilidad de comprensión.

1.3.3. Herramientas CASE de modelado

La Ingeniería de Software Asistida por Computadora, CASE por sus siglas en inglés, constituye la aplicación de métodos y técnicas utilizadas para ayudar a las actividades del proceso del software, como el análisis de requerimientos, el modelo de sistemas, la depuración y las pruebas. Las herramientas CASE representan una forma que permite Modelar los Procesos de Negocios de las empresas y desarrollar los Sistemas de Información.

Visual Paradigm 8.0

Visual Paradigm es una herramienta visual de Ingeniería de Software para el modelado. Brinda una colección de menús, barras de herramientas y ventanas que forman el área de trabajo, lo cual permite crear diferentes tipos de diagramas en un ambiente completamente visual, manejar de manera rápida aplicaciones de mayor calidad y a un menor coste. Permite dibujar todos los tipos de diagramas de clases, código inverso, generar código desde diagramas y generar documentación (Martín T., 2013).

Es una herramienta profesional multiplataforma, que proporciona tutoriales, demostraciones interactivas y proyectos UML. Posee como peculiaridad sobre el resto de las herramientas que cuenta con una potente funcionalidad para la creación de interfaces de usuarios de las aplicaciones.

Herramienta CASE de modelado seleccionada

Visual Paradigm se selecciona por brindar licencias gratuitas y comerciales. Es una herramienta multiplataforma, que presenta amplia facilidad de uso, abundantes tutoriales de UML y una potente funcionalidad para la creación de interfaces.

1.3.4. Lenguajes de Programación

En la actualidad existen diferentes lenguajes de programación, estos han surgido debido a las tendencias y necesidades de las plataformas existentes.

Desde el comienzo de Internet, surgieron diferentes demandas por los usuarios y se dieron soluciones mediante lenguajes estáticos. Con el transcurso del tiempo, las tecnologías se desarrollaron y surgieron nuevos problemas a dar solución. Esto dio lugar a desarrollar lenguajes de programación para la Web dinámica, que permitieran interactuar con los usuarios y utilizaran sistemas de bases de datos.

Lenguajes de Programación para el lado del Servidor PHP3 5.5.11

PHP es un lenguaje script que corre del lado del servidor en la Arquitectura Cliente – Servidor. Fue desarrollado originalmente por Rasmus Cerdorf en 1994. Es utilizado para generar páginas Web dinámicas. Su programación es segura y confiable ya que de ninguna manera en el navegador se accede al código fuente en PHP, sino solo a su resultado en HTML4 (González Enrique, 2010)

Con PHP se puede procesar la información de formularios, generar páginas con contenidos dinámicos, entre muchas más cosas. Permite aplicar técnicas de programación orientada a objetos.

Como se ha diseñado para su uso en la Web, incorpora gran cantidad de funciones integradas para realizar útiles tareas relacionadas con la Web. Puede generar imágenes GIF a un instante, establecer conexiones a otros servicios de red, enviar correos electrónicos, trabajar con cookies y generar documentos PDF (González Enrique, 2010).

Una de sus características es su portabilidad ya que está disponible para diferentes sistemas operativos. Dispone de una conexión propia a todos los sistemas de base de datos. Además de MySQL, puede conectarse directamente a bases de datos como PostgreSQL, Microsoft SQL, Oracle, entre otras. Es libre,

por lo que se presenta como una alternativa de fácil acceso para todos, no limita su distribución y se puede ampliar con nuevas funcionalidades si se desea. Se puede utilizar como módulo de Apache, lo que lo hace extremadamente veloz. Por estar completamente escrito en C, se ejecuta rápidamente utilizando poca memoria.

Lenguajes de Programación para el lado del Cliente JavaScript

Es uno de los lenguajes más utilizados en la creación de aplicaciones web presentando funcionalidades para la creación de contenidos dinámicos. Es un lenguaje de programación para el lado del cliente debido a que la mayor carga de procesamiento la soporta el navegador. Su uso está destinado principalmente para la creación de efectos visuales así como para la creación de interfaces de usuario. Las aplicaciones que utilizan este lenguaje casi siempre están incrustadas en los códigos HTML de las aplicaciones web y están destinadas para realizar las acciones con el cliente.

CSS

Es un lenguaje que permite la definición de hojas de estilos diseñadas para el control de las interfaces de usuario definidas en los códigos HTML. Separa los contenidos de la presentación de las aplicaciones lo que permite el diseño de documentos bien definido y muy complejo. Cuando son creados los contenidos de estilos es permitido definir los diferentes tamaños, colores y tipos de fuentes para las letras, así como la gestión de imágenes.

Biblioteca de JavaScript: ExtJS 3.4.0

Es una biblioteca del Lenguaje JavaScript que facilita la implementación de aplicaciones web que contiene tecnologías como Ajax, DHTML y DOM con grandes funcionalidades para la implementación de Interfaces de Usuarios avanzadas. Fue diseñada originalmente como una extensión de la librería YUI por Sencha y actualmente contiene además librerías de JQuery y Prototype y su licencia es de Código Abierto.

Lenguajes de programación seleccionados

Los lenguajes de programación del lado del cliente escogidos fueron CSS y JavaScript, este último se escogió ya que los programas escritos en él no requieren de mucha memoria ni tiempo adicional de transmisión, por ser pequeños y compactos y es independiente de la plataforma, hardware o sistema operativo y funciona correctamente siempre y cuando exista un navegador que lo soporte.

El lenguaje escogido para el lado del servidor fue PHP por ser un lenguaje multiplataforma que posee capacidad de conexión con la mayoría de los manejadores de base de datos que se utilizan en la actualidad, destaca su conectividad con MySQL. Este lenguaje es libre y la amplísima biblioteca que posee simplifica el trabajo de los desarrolladores.

1.3.5. Framework

Los *frameworks* o marcos de trabajo son aplicaciones compuestas por componentes personalizables que permiten el desarrollo de aplicaciones. Las funcionalidades principales de estas aplicaciones es aligerar de carga a los desarrolladores realizando de manera automática e interactiva tareas comunes dentro de la programación, lo que permite reducir considerablemente el tiempo de desarrollo de aplicaciones complejas.

Framework Symfony 2.3.9

Symfony es un poderoso *Framework* orientado para la optimización del desarrollo de aplicaciones web basado en el patrón de diseño MVC. El diseño del mismo separa la lógica del negocio, del servidor y la presentación; además proporciona herramientas para reducir el tiempo de desarrollo de aplicaciones complejas. Realiza de forma automatizada las tareas básicas durante el desarrollo permitiendo al desarrollador centrarse en las funciones más específicas de la aplicación. El *Framework* está implementado parcialmente en PHP 5.3, es compatible con los Sistemas Gestores de Base de Datos más reconocidos del mundo y es independiente al sistema operativo del servidor, brindando así características muy poderosas para el desarrollo de aplicaciones web (Potencier, 2009). Representa una arquitectura de software que modela las relaciones generales de las entidades del dominio (Framework, 2014).

Framework seleccionado

Symfony es un *framework* que se escoge para la realización del sistema porque presenta múltiples posibilidades con el trabajo con las validaciones, integración con otras herramientas y son muy factibles para realizar pruebas y depuración del código.

1.3.6. IDE de desarrollo

Para desarrollar, normalmente solo es necesario un editor de texto, un intérprete o compilador y una terminal de líneas de comando, un Entorno de Desarrollo Integrado (IDE). Esto simplifica el trabajo y ahorra tiempo de desarrollo.

NetBeans 8.0

NetBeans es una aplicación de código abierto diseñada para el desarrollo de aplicaciones portables entre las distintas plataformas. Dispone de soporte para crear interfaces gráficas de forma visual, desarrollo de aplicaciones web, control de versiones, entre otras. Es un programa con Licencia Común de Desarrollo y Distribución (CDDL) que permite su uso libremente, brinda el código del mismo y es totalmente gratuita. Permite escribir, compilar, hacer un debut, ensamblar y desplegar aplicaciones y aunque está escrito en Java, brinda soporte para toda clase de lenguajes de programación. Funciona en sistemas operativos compatibles con la máquina virtual Java (Windows XP, Vista, Windows 7, Ubuntu 9.10, Solaris, Mac OS X 10.5 o superior).

IDE de desarrollo seleccionado

Para el desarrollo de la aplicación informática se seleccionó NetBeans, por permitir escribir, compilar, ensamblar y desplegar aplicaciones, además de brindar soporte para varios lenguajes de programación mediante el empleo de plugins. Facilita el completamiento para los lenguajes de programación HTML, CSS, JavaScript y PHP permitiendo tracear este último.

1.3.7. Sistemas Gestores de Base de Datos (SGBD)

Un SGBD es un conjunto de programas que permiten crear y mantener una Base de datos, asegurando su integridad, confidencialidad y seguridad. Permite almacenar y posteriormente acceder a los datos de forma rápida y estructurada. Las aplicaciones más usuales son para la gestión de empresas e instituciones públicas, además de ser ampliamente utilizado en entornos científicos con el objeto de almacenar la información experimental (Días Roberto, 2010).

PostgreSQL 9.3

PostgreSQL es un SGBD Objeto-Relacional basado en el proyecto Postgres, de la Universidad de Berkeley. Es una derivación libre de este proyecto y utiliza el lenguaje SQL. Fue pionero en muchos de los conceptos del sistema objeto-relacional actual. Este proyecto lleva más de una década de desarrollo, siendo hoy día, el sistema libre más avanzado, soportando la gran mayoría de las transacciones SQL y control concurrente. Entre sus principales ventajas se destacan las siguientes:

- Permite la gestión de diferentes usuarios, como también los permisos asignados a cada uno de ellos.

- Posee una gran escalabilidad, haciéndolo idóneo para su uso en sitios Web que atienden un gran número de solicitudes.
- Puede ser instalado un número ilimitado de veces sin temor de sobrepasar la licencia.
- Posee estabilidad y confiabilidad legendarias.
- Es extensible a través del código fuente disponible sin costos adicionales.
- Es multiplataforma, disponible en la mayoría de los sistemas operativos.
- Permite implementar reglas, vistas, disparadores, subconsultas y funciones.
- Posee herramientas para generar SQL portable para compartir con otros sistemas compatibles con SQL.

MySQL

MySQL es un sistema de gestión de bases de datos relacional multiusuario, cuya idea originaria es de la empresa OpenSource MySQL AB. El objetivo que persigue esta empresa consiste en que MySQL cumpla el estándar SQL, pero sin sacrificar velocidad, fiabilidad o usabilidad. Existen varias interfaces de programación de aplicaciones que permiten, a aplicaciones escritas en diversos lenguajes de programación, acceder a las bases de datos MySQL, incluyendo C, C++, C#, Pascal y Delphi. Es muy utilizado en aplicaciones web, como Drupal o phpBB, en plataformas Linux, Windows, PHP, Perl y Python y por herramientas de seguimiento de errores como Bugzilla. Su popularidad como aplicación web está muy ligada a PHP, que a menudo aparece en combinación con MySQL.

Es una base de datos muy rápida en la lectura cuando utiliza el motor no transaccional MyISAM, pero puede provocar problemas de integridad en entornos de alta concurrencia en la modificación. En aplicaciones web hay baja concurrencia en la modificación de datos y en cambio el entorno es intensivo en lectura de datos, lo que hace a MySQL ideal para este tipo de aplicaciones. Sea cual sea el entorno en el que va a utilizar, es importante monitorizar de antemano el rendimiento para detectar y corregir errores tanto de SQL como de programación (MySQL-Ecured).

Gestor de Base de datos seleccionado

Después de analizar las características y prestaciones de los SGBD mencionados anteriormente, se aprecian las ventajas que brinda PostgreSQL. Este sistema provee de gran capacidad de almacenamiento, consistencia, escalabilidad y rendimiento bajo grandes cargas de trabajo. Es un SGBD objeto-relacional, distribuido bajo la licencia BSD13 y su código fuente se encuentra disponible libremente.

Por lo tanto se decide desarrollar solución propuesta en esta investigación haciendo uso del SGBD PostgreSQL en su versión 9.3

1.3.8. Servidor Web

Los servicios web tal como lo define el consorcio World Wide Web (WC3) son un conjunto de aplicaciones o de tecnologías con capacidad para inter-operar en la web (World Wide Web, 2014) estas aplicaciones o tecnologías intercambian datos entre sí, con el objetivo de ofrecer servicios.

Servidor Web Apache

El servidor Apache es desarrollado por la *Apache Software Foundation*, presenta entre otras características mensajes de error altamente configurables, bases de datos de autenticación y negociado de contenido, entre sus principales ventajas se encuentran: posee gran cantidad de extensiones para diversas tecnologías, además de una amplia documentación, es libre, modular y multiplataforma. Se señala como desventaja que no posee interfaz gráfica que facilite su configuración.

Este servidor Web es otro de los más ligeros que hay en el mercado. Está especialmente pensado para hacer cargas pesadas sin perder balance, utilizando poca RAM y poca de CPU. Algunas páginas populares que lo usan son Youtube, Wikipedia y otras que soportan gran tráfico diariamente. Es gratuito y se distribuye bajo Licencia BSD.

Servidor Web seleccionado

Se seleccionó el servidor web Apache debido a que consume pocos recursos, es gratuito, es uno de los más usados y presenta una amplia documentación.

Conclusiones del capítulo

Una vez realizado el estudio de la literatura científica, se pudo identificar la ineficiencia de la herramienta MEYCOR-COSO, ya que no se adecua a la necesidad de gestión del control interno del Vicedecanato Administrativo. Por lo que se plantea un sistema informático que garantice las necesidades requeridas, utilizando la metodología de desarrollo de software Open Up y las herramientas y técnicas necesarias para guiar el proceso de desarrollo de software.

Capítulo 2: Descripción del sistema

Introducción

En el presente capítulo se detallan las características del sistema, a partir de la descripción general de la propuesta de solución. Se modela el mapa de procesos del negocio e identifican los requisitos funcionales y no funcionales del sistema con la respectiva descripción de cada uno de ellos y se traza la línea base de la arquitectura del sistema en cuestión.

2.1. Propuesta de solución

A partir de las deficiencias identificadas anteriormente la Facultad 6 de la UCI para la gestión del Sistema del Control Interno y para dar solución al problema planteado en la investigación, se propone desarrollar una aplicación que permita realizar un seguimiento y análisis de los indicadores defectuosos, realizando así una trazabilidad de los datos de manera real de manera trimestral que esté compuesta por los siguientes módulos. Es válido aclarar que la propuesta de solución no puede funcionar correctamente sin la presencia de uno de sus módulos.

- **Módulo Recursos Humanos:** Permite la gestión de usuarios en el sistema, a partir de las distintas interfaces de las funcionalidades, el sistema mostrará un mensaje confirmando la correcta ejecución de la acción realizada; de igual manera al elegir un elemento para eliminar, adicionar o actualizar se mostrará un mensaje de confirmación para la acción que se va a ejecutar.
- **Módulo Administración:** Permite la gestión de criterios y de componentes en el sistema, a partir de las distintas interfaces de las funcionalidades, el sistema mostrará un mensaje confirmando la correcta ejecución de la acción realizada; de igual manera al elegir un elemento para eliminar, adicionar o actualizar se mostrará un mensaje de confirmación para la acción que se va a ejecutar.
- **Módulo Índice de Control:** Permite realizar los cálculos para determinar el índice de control interno en el sistema, a partir de las distintas interfaces.
- **Módulo Reporte:** Permite generar los reportes de los criterios que el usuario desee a partir de las distintas interfaces en el sistema.



Figura 1: Propuesta de solución.

2.2. Modelo de Negocio

Un Proceso de Negocio es una colección de actividades que, tomando una o varias clases de entradas, crean una salida que tiene valor para un cliente. Estos procesos del negocio se representan a través de los mapas de procesos, los cuales ofrecen una visión general del sistema. Con el Modelo del Negocio se pretende llegar a un mejor entendimiento de la organización donde se va a implantar el producto. Los objetivos principales de esta actividad son entender la estructura y la dinámica de la organización para la cual el sistema va a ser desarrollado (organización y objetivo) y de esta manera poder derivar los requerimientos del sistema necesarios para apoyar la organización objetivo (Días Roberto, 2010).

2.2.1. Actores del negocio

Los actores del negocio son roles que juegan los software, personas, grupos u organizaciones que interactúan con un negocio determinado, en este caso el negocio se desarrolla en la Facultad 6 de la UCI y en esta organización los actores del negocio se identifican a continuación:

Tabla 2: Actores del negocio.

Actor	Descripción
Vicedecano administrativo	Encargado de evaluar el cumplimiento de los indicadores del Sistema de Control Interno en la organización.
Departamento docente	Encargado de ejecutar el cumplimiento de los indicadores del Sistema de Control Interno en la organización.

2.2.2. Trabajadores del negocio

Los trabajadores del negocio son una representación abstracta de los seres humanos o de un sistema que representa los roles del desarrollo de las actividades dentro de un caso de uso del negocio. Siendo la abstracción que ayuda a identificar todas las mejoras en los procesos de negocio.

Permite analizar los efectos que se pueden obtener debido a la automatización de estos procesos para la organización. En la tabla siguiente se muestra la descripción de los trabajadores del negocio para el modelado del sistema.

Tabla 3: Trabajadores del negocio.

Trabajador	Descripción
Jefe de departamento.	Encargado de ejecutar la Guía de Autocontrol de la República.
Vicedecano Administrativo.	Encargado de evaluar los procesos del control interno en la Facultad 6 (visualizar reportes, gestionar componentes y criterios), en correspondencia con lo establecido en la Guía de Autocontrol de la República.
Experto.	Personas encargadas de determinar la importancia de criterios evaluativos en la guía de autocontrol.
Técnicos en economía de la Facultad 6.	Encargados de verificar los resultados de la implementación del Sistema de Control Interno en la organización.

2.3. Diagrama de casos de uso del negocio

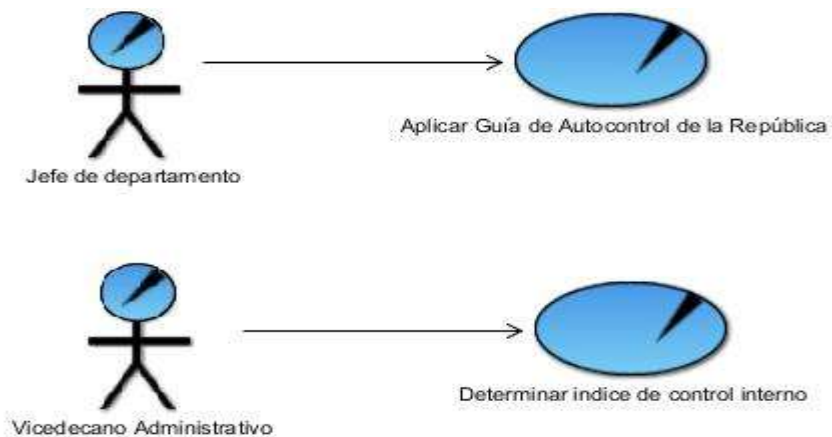


Figura 2: Caso de uso del negocio.

2.4. Diagrama de actividades del modelo de negocio

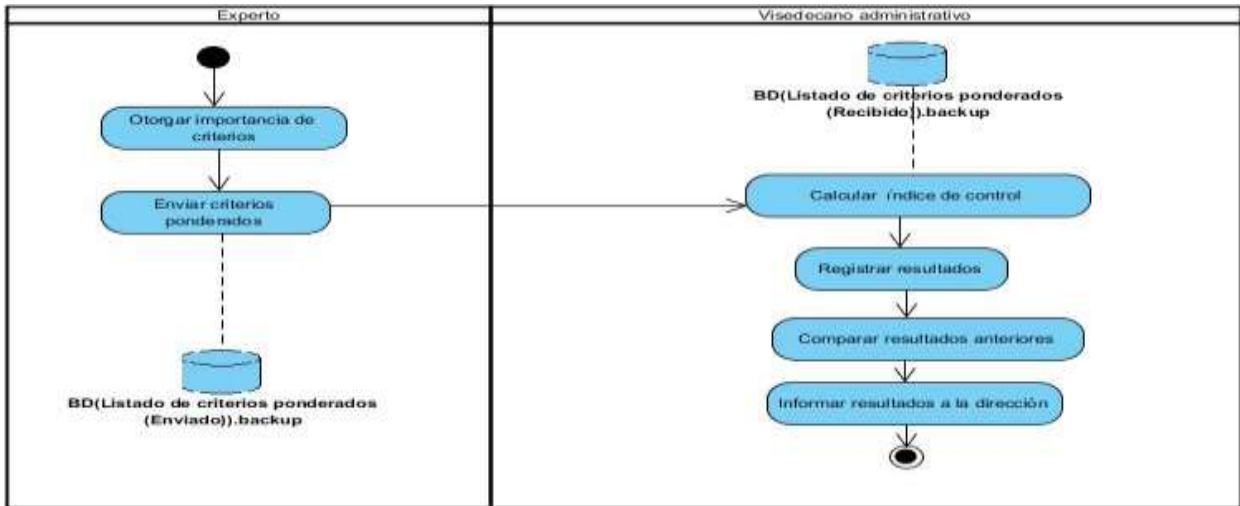


Figura 3: Diagrama de actividades: Determinar índice de control interno.

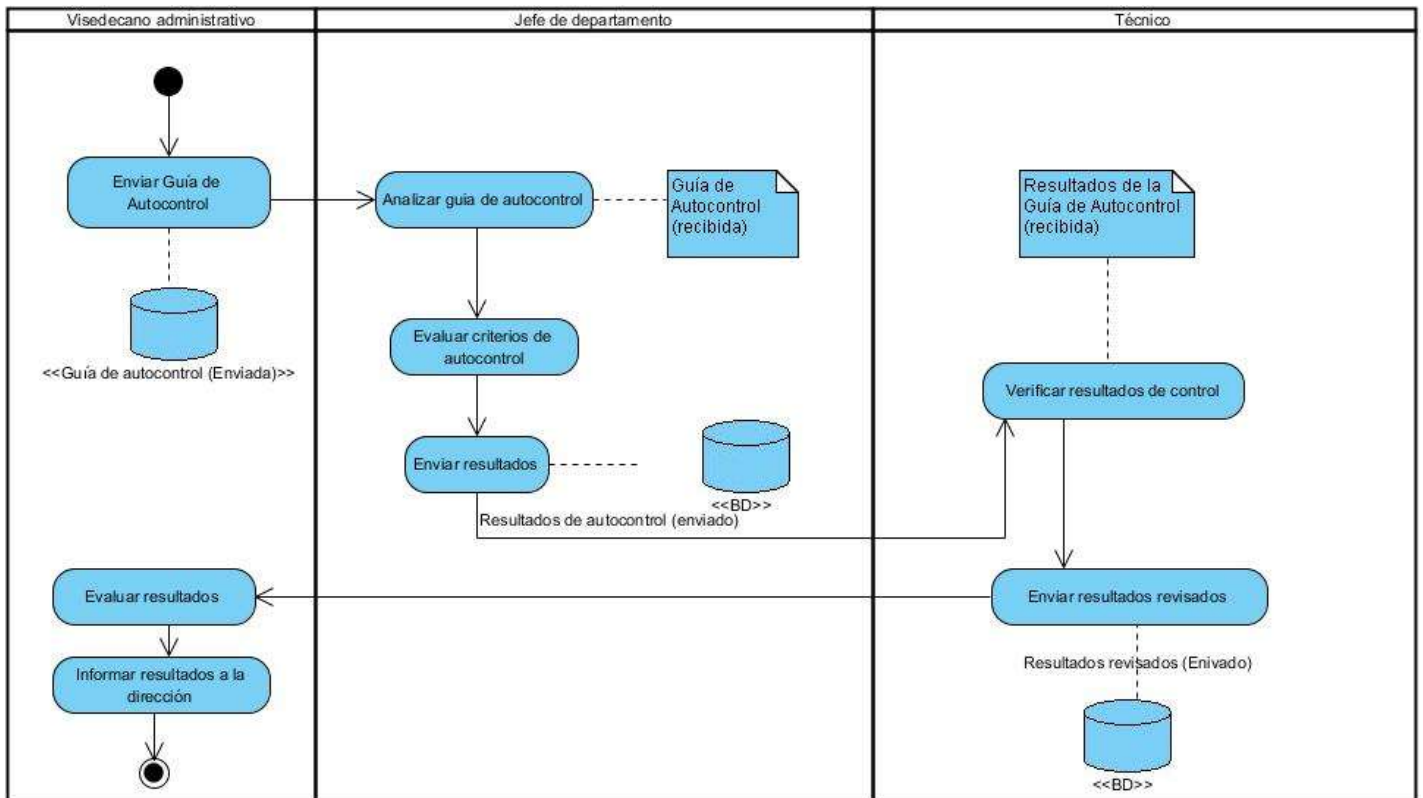


Figura 4: Diagrama de actividades: Aplicar Guía de Autocontrol de la República.

2.5. Descripción de los casos de usos del negocio

Tabla 4: Caso de uso del negocio: Aplicar Guía de Autocontrol de la República.

CUNegocio	Aplicar Guía de Autocontrol de la República.	
Actores	Jefe de departamento.	
Resumen	El caso de uso muestra como el jefe de un departamento determinado de la Facultad 6 de la UCI, evalúa los criterios de autocontrol y envían los resultados del mismo al Técnico medio en economía.	
Complejidad	Alta	
Prioridad	Alta	
Precondiciones	NA.	
Postcondiciones	Se ejecuta lo establecido en la Guía de Autocontrol de la República.	
Flujo de eventos		
Flujo básico		
	Actor	Respuesta del Negocio
1.	El caso de uso inicia cuando el jefe de un departamento de la Facultad 6 de la UCI evalúa cada uno de los criterios establecidos en la Guía de Autocontrol de la República.	
2.		Se realiza una evaluación de los criterios establecidos en la guía de autocontrol en los distintos departamentos de la Facultad.
3.		El departamento envía los datos evaluados al Técnico medio en economía.

4.		Se discuten los resultados obtenidos y se elabora un informe general de la evaluación.
5.	El jefe del departamento toma las medidas pertinentes para la próxima evaluación.	
6.		Finaliza el caso de uso del negocio.
	Prioridad	Crítica.

Tabla 5: Caso de uso del negocio: Determinar el índice de control interno.

CUNegocio	Determinar el índice de control interno.	
Actores	Vicedecano Administrativo.	
Resumen	El caso de uso expone el proceso establecido que realiza el Vicedecano Administrativo para determinar el índice de control interno, realiza los procedimientos matemáticos y cálculos necesarios, en correspondencia con los datos registrados en el vicedecanato administrativo de la organización.	
Complejidad	Alta	
Prioridad	Alta	
Precondiciones	Los criterios para determinar el índice organizacional tienen que estar ponderados.	
Postcondiciones	Se determina el índice de control interno.	
Flujo de eventos		
Flujo básico		
	Actor	Respuesta del Negocio

1.	El caso de uso inicia cuando el Vicedecano Administrativo calcula el índice de control interno en la organización.	
2.		Se realizan los procedimientos matemáticos necesarios para determinar el índice de control.
3.		Los indicadores evaluados se analizan con la Facultad 6 para realizar una valoración del control interno en la organización.
4.		Se archivan las evaluaciones obtenidas, junto con el índice de control interno de la Facultad 6.
5.	Se informan los resultados pertinentes a la dirección.	
6.		Finaliza el caso de uso del negocio.
	Prioridad	Crítica.

2.6. Extracción de Requisitos

La especificación de los requisitos de software es una descripción completa del comportamiento del sistema que se va a desarrollar. Contiene requisitos funcionales que son las capacidades o condiciones que el sistema debe cumplir. En el que se incluye un conjunto de casos de uso que describen las interacciones que tendrán los usuarios con el software y contiene además, requisitos no funcionales que son las propiedades o cualidades que el producto debe tener, que imponen restricciones en el diseño o la implementación. En esta etapa se debe generar una información clara y precisa de los aspectos más relevantes del producto, ya que esta actividad es el hilo conductor de todo el desarrollo del software.

2.6.1. Requisitos funcionales

Los Requisitos Funcionales (RF) definen las condiciones y funciones que el sistema será capaz de realizar. Luego de efectuar la modelación del dominio se obtuvieron los siguientes requisitos funcionales:

Autenticar usuario

RF 1. Autenticar usuario.

Gestionar usuarios

RF 2. Adicionar usuario.

RF 3. Eliminar usuario.

RF 4. Actualizar usuario.

RF 5. Buscar usuario.

RF 6. Búsqueda avanzada del usuario.

Gestionar criterios

RF 7. Actualizar criterio.

RF 8. Buscar criterio.

RF 9. Insertar criterio.

RF 10. Eliminar criterio.

Gestionar componentes

RF 11. Buscar componente.

RF 12. Actualizar componente.

RF 13. Asociar criterio al componente.

RF 14. Eliminar asociación del criterio al componente.

Otorgar importancia a los componentes

RF 15. Ponderar componente.

Evaluar criterio

RF 16. Evaluar criterio.

Determinar índice de control

RF 17. Determinar índice de control.

Visualizar reportes

RF 18. Generar reportes de criterios evaluados de mal.

RF 19. Generar reportes de criterios evaluados de bien.

RF 20. Generar reportes de la Tabla de aplicación de la Guía de Autocontrol de la República.

RF 21. Generar reportes de criterios.

2.6.2. Requisitos no funcionales

Los Requisitos No Funcionales (RNF) son propiedades o cualidades, que pueden usarse para juzgar la operación de un sistema en lugar de su funcionalidad. El sistema debe satisfacer las siguientes condiciones o cualidades: que el producto debe tener.

Requerimientos de Software

El Software se debe ejecutar sobre el sistema operativo Linux o Windows y la PC cliente con acceso al servidor web (apache 2.0), debe contar con al menos un navegador Web, por ejemplo (Mozilla Firefox desde su versión 30.0 hasta 38.0).

Requerimientos de Hardware

Estaciones de Trabajo:

- Se requiere tarjeta de red.
- Se requiere que tenga al menos 1 GB de memoria RAM.
- Procesador Dual Core como mínimo.

Servidores:

- Se requiere tarjeta de red.
- Se requiere que tenga al menos 1 GB de RAM.
- Procesador Dual Core como mínimo.

Requerimientos de Restricciones en el diseño y la implementación

- El sistema se desarrollará con tecnología PHP versión 5.5.11 o superior.
- El análisis y diseño del sistema estará basado en la metodología de desarrollo Open Up con el uso del lenguaje de modelado UML 2.0.
- Se utilizará Visual Paradigm 8.0 como herramienta CASE para el modelado de los artefactos que se generan con cada flujo de trabajo.
- Para el servidor web se utilizará tecnología Apache versión 2.0. o superior.
- La base de datos se implementará en Postgres SQL versión 9.3 o superior.

Requerimientos de Apariencia o interfaz externa

- Diseño perfectamente encuadrado para resoluciones de 800x600, pero preparado para verse en otras resoluciones.
- Para la realización del diseño se tendrá en cuenta el control y la transparencia como eje principal en torno al cual girará el sistema, además debe permitir el uso de colores claros o legibles como azul y blanco.

Requerimientos de Usabilidad

- Los usuarios deben tener un conocimiento básico sobre informática para poder trabajar con el software a desarrollar.

Requerimientos de Portabilidad

- El sistema debe ser multiplataforma, es decir la propuesta de solución puede ser manejable en el sistema operativo Linux o Windows .

Requerimientos de Seguridad Confidencialidad

Confidencialidad:

- La información manejada por el sistema deberá estar protegida de acceso no autorizado y divulgación.
- Los usuarios accederán a la información correspondiente a cada uno, ya que el sistema brinda un mecanismo de autenticación que garantiza lo antes planteado.

Integridad:

- La información manejada por el sistema será objeto de cuidadosa protección contra estados inconsistentes y corrupción; a través de roles y permisos de los usuarios autenticados (R-Back).

Disponibilidad:

- A los usuarios autorizados se les deberá garantizar el acceso a la información solicitada en todo momento, mientras que los no autorizados no presentaran los mismo privilegios (no tendrán acceso a la información).

Encriptación:

La contraseña de la base de datos se encuentra protegida, por el algoritmo de encriptación md5.

2.7. Modelo de casos de uso

El modelo de casos de uso del sistema permite que los desarrolladores de software y los clientes lleguen a un acuerdo sobre los requisitos, es decir sobre las condiciones y posibilidades que debe cumplir el sistema. Además proporciona la entrada fundamental para el análisis, diseño y pruebas.

2.7.1. Actores del sistema

Para el acceso a las funcionalidades que se desarrollarán se cuenta con varios actores, que se definen a continuación.

Tabla 6: Actores del sistema.

Actor	Descripción
Vicedecano Administrativo.	Encargado de la gestión de los usuarios del sistema y de gestionar la información referente a las unidades organizativas, el mismo puede realizar el cálculo del índice organizacional, gestionar criterios y componentes.

Jefe de departamento.	Usuario del sistema encargado de evaluar los criterios registrados.
Técnicos en economía de la Facultad 6.	Usuarios del sistema que solo pueden visualizar la información existente en el mismo, de acuerdo al nivel de acceso que posean.
Expertos.	Usuario del sistema encargado de determinar la importancia de los componentes.

2.8. Diagrama de casos de uso

Un diagrama de casos de uso del sistema (DCU) representa gráficamente a los procesos y su interacción con los actores. Además facilita el entendimiento de los procesos realizados por el sistema para el desarrollador.

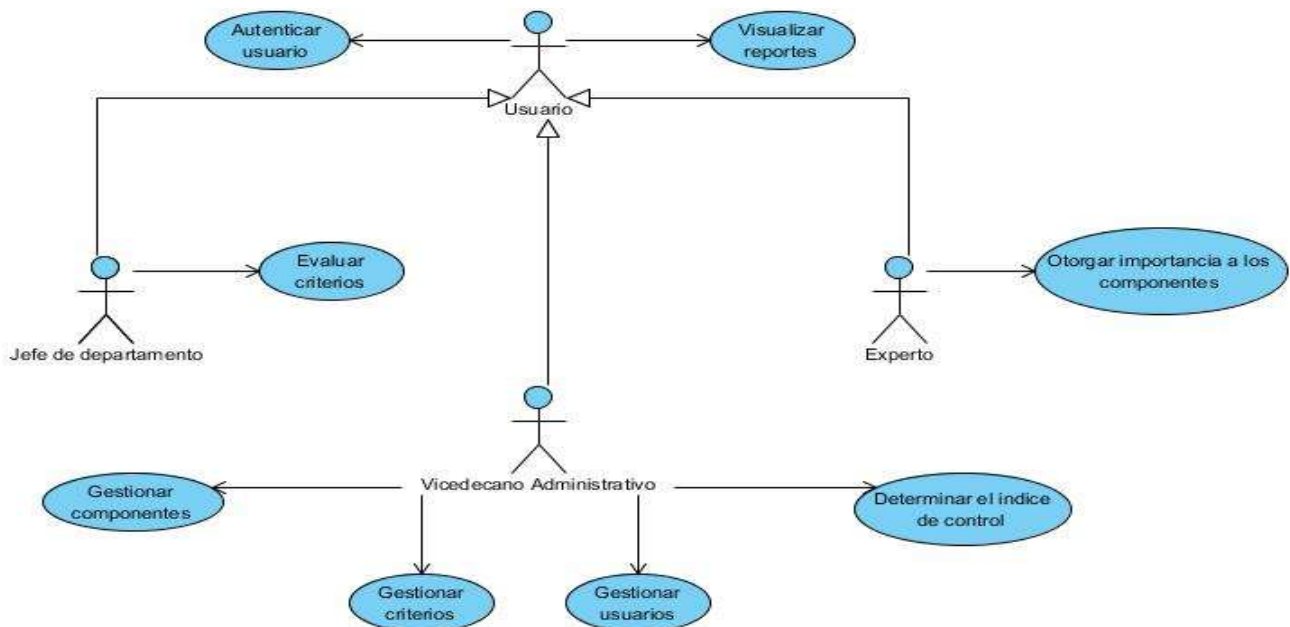


Figura 5: Diagrama de casos de uso del sistema.

2.9. Descripción de casos de uso

Tabla 7: Descripción de caso de uso: Gestionar criterios.

Objetivo	Gestionar criterio.
-----------------	---------------------

Actores	Vicedecano Administrativo (inicia).	
Resumen	El Vicedecano Administrativo solicita la gestión de criterios, el sistema muestra una opción en la parte izquierda del menú para la gestión de criterios y se muestra la lista de criterios, con la posibilidad de adicionar, insertar, actualizar y eliminar los criterios.	
Complejidad	Alta.	
Prioridad	Crítica.	
Precondiciones	-	
Postcondiciones	Se inserta, actualiza, adiciona, elimina o busca criterios.	
Flujo de eventos		
Flujo básico		
	Actor	Sistema
1.	Selecciona la opción "Criterio" en el menú del lateral izquierdo del sistema.	
2.		Muestra un listado con los criterios almacenados y posibilita al Vicedecano Administrativo ejecuta las siguientes acciones: Adicionar criterio. Ver Sección 1 Eliminar criterio. Ver Sección 2 Actualizar criterio. Ver sección 3. Buscar criterio. Ver sección 4.
3.	El Vicedecano Administrativo escoge una de las cuatro opciones.	
4.		Finaliza el caso de uso.

Sección 1: “Solicita la adición del criterio”		
Flujo básico		
	Actor	Sistema
1	Selecciona la opción “Adicionar” de la página principal de los criterios.	
2		Muestra una ventana con un formulario de registro con los siguientes campos: (nombre y descripción).
3	Adiciona los datos y da clic en el botón “Aceptar”.	
4		Se muestra un mensaje “Criterio adicionado correctamente”.
Flujo alterno		
“Escoge la opción cancelar.”		
	Actor	Sistema
1	Selecciona la opción “Cancelar”, que se encuentra en el borde inferior derecho, de la página principal de la sección “Adicionar criterio”.	
2		Finaliza el flujo alterno.
Sección 2: “Solicita la eliminación del criterio”		
Flujo básico		
	Actor	Sistema
1	Selecciona la opción “Eliminar” de la página principal de las evidencias.	

2		Se muestra una ventana con la lista de criterios.
3	Selecciona el criterio que desee y da clic en el botón "Eliminar".	
4		Se muestra un mensaje "Desea eliminar el criterio"
5	El Vicedecano Administrativo da clic en el botón "si".	
6		Se muestra un mensaje de notificación "Criterio eliminado correctamente".
Flujo alterno		
"Escoge la opción cancelar."		
	Actor	Sistema
1	Selecciona la opción "Cancelar", que se encuentra en el borde inferior derecho, de la página principal de la sección "Eliminar criterio".	
2		Finaliza el flujo alterno.
Sección 3: "Solicita la actualización del criterio"		
Flujo básico		
	Actor	Sistema
1	Selecciona la opción "Actualizar" de la página principal de los criterios.	
2		Se muestra un formulario (nombre y descripción).

3	El Vicedecano Administrativo modifica los campos del formulario de criterios y da clic en el botón "Aceptar".	
4		Se muestra un mensaje de notificación "Criterio actualizado correctamente".
Flujo alterno		
"Escoge la opción cancelar."		
1	Selecciona la opción "Cancelar", que se encuentra en el borde inferior derecho, de la página principal de la sección "Actualizar criterio".	
2		Finaliza el flujo alterno.
Sección 4: "Solicita la búsqueda del criterio"		
Flujo básico		
	Actor	Sistema
1	Selecciona la opción "Buscar" de la página principal de los criterios.	
2		Permite la búsqueda del criterio por el nombre del mismo (nombre).
3	El Vicedecano Administrativo realiza la búsqueda del criterio y da clic en el botón "Buscar".	
4		Muestra el criterio buscado en caso de existir, sino muestra el mensaje "El criterio no se encuentra registrado".
Requisitos funcionales		no Seguridad.

Conclusiones del capítulo

La utilización de un modelo de negocio bien definido y la generación de los artefactos del mismo, permitieron conocer, el funcionamiento de todos los procesos que se llevan a cabo en la organización, correspondientes al área del Vicedecanato Administrativo para aplicar la Guía de Autocontrol de la República; a partir de los datos procesados. Así como la objetiva descripción de actores y trabajadores que intervienen en el proceso, permitiendo definir los requisitos funcionales y no funcionales del sistema. Presentando la propuesta de solución mayor organización estructural y organizativa.

Capítulo 3: Análisis y diseño del sistema

Introducción

Está relacionado con el diseño del sistema; se describen los patrones de diseño empleados. Se presentan los diagramas de clases del diseño, diagramas de interacción y diagrama de despliegue del sistema con el objetivo de tener una idea de cómo quedará.

3.1. Modelo del diseño

El Modelo de Diseño es una disciplina que no se puede obviar en el proceso de desarrollo del software. Es imprescindible para comprender la forma en que va a funcionar el sistema en conjunto con los requisitos, lenguajes de programación, componentes reutilizables y tecnologías de interfaz de usuario que se eligieron para el desarrollo del mismo. Es una representación gráfica, mediante varios diagramas muy explícitos, de la implementación del sistema.

3.2. Descripción de patrones arquitectónicos

Los patrones ayudan al arquitecto a definir la composición y el comportamiento del sistema de software, y una combinación adecuada de ellos permite alcanzar los requerimientos de calidad. A continuación se describen los utilizados en el desarrollo de la propuesta de solución.

3.2.1. Patrones arquitectónicos

Los patrones arquitectónicos expresan el esquema de organización estructural fundamental para sistemas de software. Provee un conjunto de subsistemas predefinidos, especifica sus responsabilidades e incluye reglas y pautas para la organización de las relaciones entre ellos. Propone que son plantillas para arquitecturas de software concretas, que especifican las propiedades estructurales de una y tienen un impacto en la arquitectura de subsistemas. La selección de un patrón arquitectónico es, por lo tanto, una decisión fundamental de diseño en el desarrollo de un sistema de software. (Buschmann F et al. 1996).

Los patrones arquitectónicos:

- Definen la estructura básica de una aplicación.
- Pueden contener o estar contenidos en otros patrones.
- Proveen un subconjunto de subsistemas predefinidos, incluyendo reglas y pautas para su organización.

- Son una plantilla de construcción.

3.2.2. Patrón arquitectónico Modelo – Vista – Controlador

La Arquitectura de Software es la organización fundamental de un sistema encarnada en sus componentes, las relaciones entre ellos y el ambiente y los principios que orientan su diseño y evolución.

La necesidad del manejo de la arquitectura de un sistema de software nace con los sistemas de mediana o gran envergadura, que se proponen como solución para un problema determinado. En la medida que los sistemas de software crecen en complejidad, dado por el número de requerimientos o por el impacto de los mismos, se hace necesario establecer medios para el manejo de esta complejidad. En general, la técnica es descomponer el sistema en componentes que agrupan aspectos específicos del mismo y que al organizarse, de cierta manera constituyen la base de la solución de un problema en particular.

MVC son las siglas de Modelo Vista Controlador, que es un patrón de arquitectura de software cuya función es subdividir una aplicación en tres módulos que corresponden a la vista del usuario (la interfaz a la que accede el usuario), una lógica de control para captar los eventos que el usuario ha generado a través de la interfaz, y un modelo que gestiona los datos según le indique la lógica de control.

Modelo: Esta es la representación específica de la información con la cual el sistema opera y se compone por el Sistema de Gestión de Base de Datos y la lógica de negocio. La lógica de negocio asegura la integridad de estos y permite derivar nuevos datos. El Sistema de Gestión de Base de Datos (SGBD) será el encargado de almacenar los cambios en los datos (agregar datos, editarlos o borrarlos) producidos por la lógica de negocio; ejemplos de SGBD son MySQL, Oracle... Es recomendable una capa de abstracción extra denominada Data Access Object (DAO), que es un componente de software que suministra una interfaz común entre la lógica de negocio y el SGBD.

Vista: Este presenta el modelo en un formato adecuado para interactuar, usualmente la interfaz de usuario. Por lo tanto, la vista es la encargada de presentar los datos al usuario y la interfaz necesaria para modificarlos. Un ejemplo de tecnología podría ser las JSP que, mediante el servidor, genera HTML que interpreta el navegador del usuario mostrándole los datos y los formularios que constituyen la vista para que pueda interactuar con la aplicación.

Controlador: Este responde a eventos, usualmente acciones del usuario e invoca cambios en el modelo y probablemente en la vista. Por lo general, el controlador sería la unidad central que comunica la vista con

el modelo y viceversa, asociando los eventos del usuario con los cambios que se producirán en el modelo y devolviendo los datos resultantes que genere el modelo a la vista que corresponda.

El patrón arquitectónico Modelo – Vista – Controlador (MVC) divide una aplicación interactiva en tres componentes. El “modelo” contiene la información central y los datos. Las “vistas” despliegan información al usuario. Los “controladores” capturan la entrada del usuario. Es el patrón arquitectónico utilizado por Symfony, separando la vista de las aplicaciones de la lógica del negocio y del controlador.

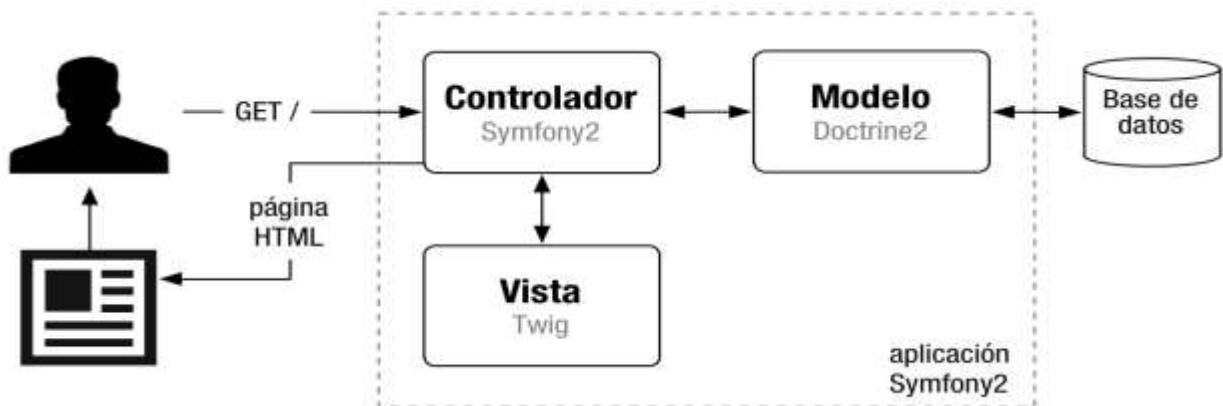


Figura 6: Patrón arquitectónico Modelo Vista Controlador.

Aunque se pueden encontrar diferentes implementaciones de MVC, el flujo que sigue el control generalmente es el siguiente:

- El usuario interactúa con la interfaz de usuario.
- El controlador recibe (a través de la interfaz) la notificación de la acción solicitada por el usuario. Es decir, el controlador gestiona el evento que llega desde la vista producida por un usuario.
- El controlador accede al modelo, ya sea con el fin de consultar datos o actualizarlos, posiblemente modificándolo de forma adecuada a la acción solicitada por el usuario.
- El controlador delega a los objetos de la vista la tarea de desplegar la interfaz de usuario. La vista obtiene sus datos del modelo para generar la interfaz apropiada para el usuario donde se refleja los cambios en el modelo. El modelo no debe tener conocimiento directo sobre la vista. Por lo general, el controlador no pasa objetos de dominio (el modelo) a la vista, aunque puede dar la orden a la vista para que se actualice. Sin embargo, en algunas implementaciones, la vista no tiene acceso directo al modelo, dejando que el controlador envíe los datos del modelo a la vista.
- La interfaz de usuario espera nuevas interacciones del usuario, comenzando el ciclo nuevamente.

3.3. Patrones de diseño

Un patrón de diseño provee un esquema para refinar los subsistemas o componentes de un sistema de software, o las relaciones entre ellos. Describe la estructura comúnmente recurrente de los componentes en comunicación, que resuelve un problema general de diseño en un contexto particular. (Buschmann F et al. 1996).

3.3.1. Patrones GRASP

Los Patrones de Principios Generales para Asignar Responsabilidades (GRASP por sus siglas en inglés) describen los principios fundamentales del diseño de objetos y la asignación de responsabilidades, expresados como patrones (Cardoso E, 2004).

- **Creador:** Asignarle a una clase la responsabilidad de crear una instancia de otra. Dentro del sistema este patrón se evidencia en las acciones de los controladores, las cuales crean objetos del modelo o los formularios que representan las entidades.
- **Bajo Acoplamiento:** Determina el nivel de dependencia de una clase con respecto a otras. Una clase con bajo acoplamiento no depende de muchas otras. Este patrón es utilizado por el *framework* Symfony, y por ende en el sistema, al no asociar las clases del modelo con las de la vista o el controlador, la dependencia entre las clases, en este caso, se mantiene baja.
- **Controlador:** Es el encargado de asignar la responsabilidad del manejo de un mensaje de los eventos de un sistema a una clase que represente una de las siguientes opciones. Se evidencia el uso de este patrón, ya que para cada petición o evento que se genere en el mismo, existe un controlador con la responsabilidad de obtenerla y devolver una respuesta. La respuesta puede ser mostrar una vista, ejecutar un método, devolver un mensaje, etc.

3.3.2. Patrones Gof

Los patrones GoF (Gang of Four o “Pandilla de los Cuatro” en español), describen las formas comunes en que diferentes tipos de objetos pueden ser organizados para trabajar unos con otros. Tratan la relación Entre clases, la combinación clases y la formación de estructuras de mayor complejidad. Nos permiten crear grupos de objetos para ayudarnos a realizar tareas complejas.

- **Observador** (*Observer*): Patrón de comportamiento. Se utiliza para mantener “informados” a objetos desacoplados entre sí y que presentan dependencia de un objeto, notificando a los primeros ante cualquier cambio que ocurra en el último. Este patrón es utilizado tradicionalmente en la capa de presentación en el diseño de componentes para las interfaces de usuario de las aplicaciones.
- **Decorador** (*Decorator*): Patrón de tipo estructura. A nivel de objetos añade responsabilidades adicionales a un objeto dinámicamente. Se utiliza este patrón para la vista y el *layout* o plantilla global que decora el contenido de la misma.

3.4. Diagrama entidad relación

El diagrama entidad relación muestra las relaciones existentes entre las clases del sistema, es una representación de las entidades de la base de datos.

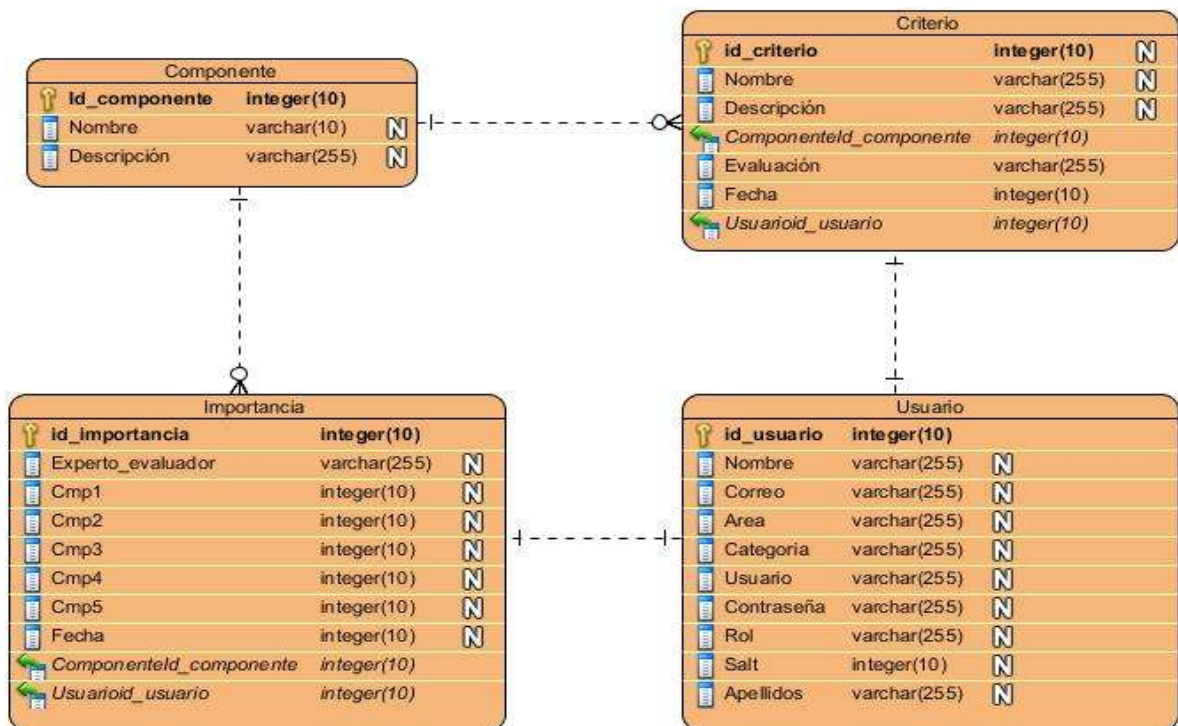


Figura 7: Diagrama Entidad Relación.

3.5. Diagrama de clases del diseño

Los diagramas de clases muestran cómo se lleva a cabo la colaboración entre las clases para dar cumplimiento a un requisito determinado. Para la elaboración de estos diagramas se hace uso de estereotipos que ayudan a representar de manera más fácil la función y el carácter de las clases dentro de la realización del requisito. A continuación se muestra el diagrama de clases del diseño del caso de uso del sistema gestionar usuario y gestionar componentes.

3.5.1. Diagrama de clases del diseño

A continuación se muestra el diagrama de clases para el CU Gestionar criterios.

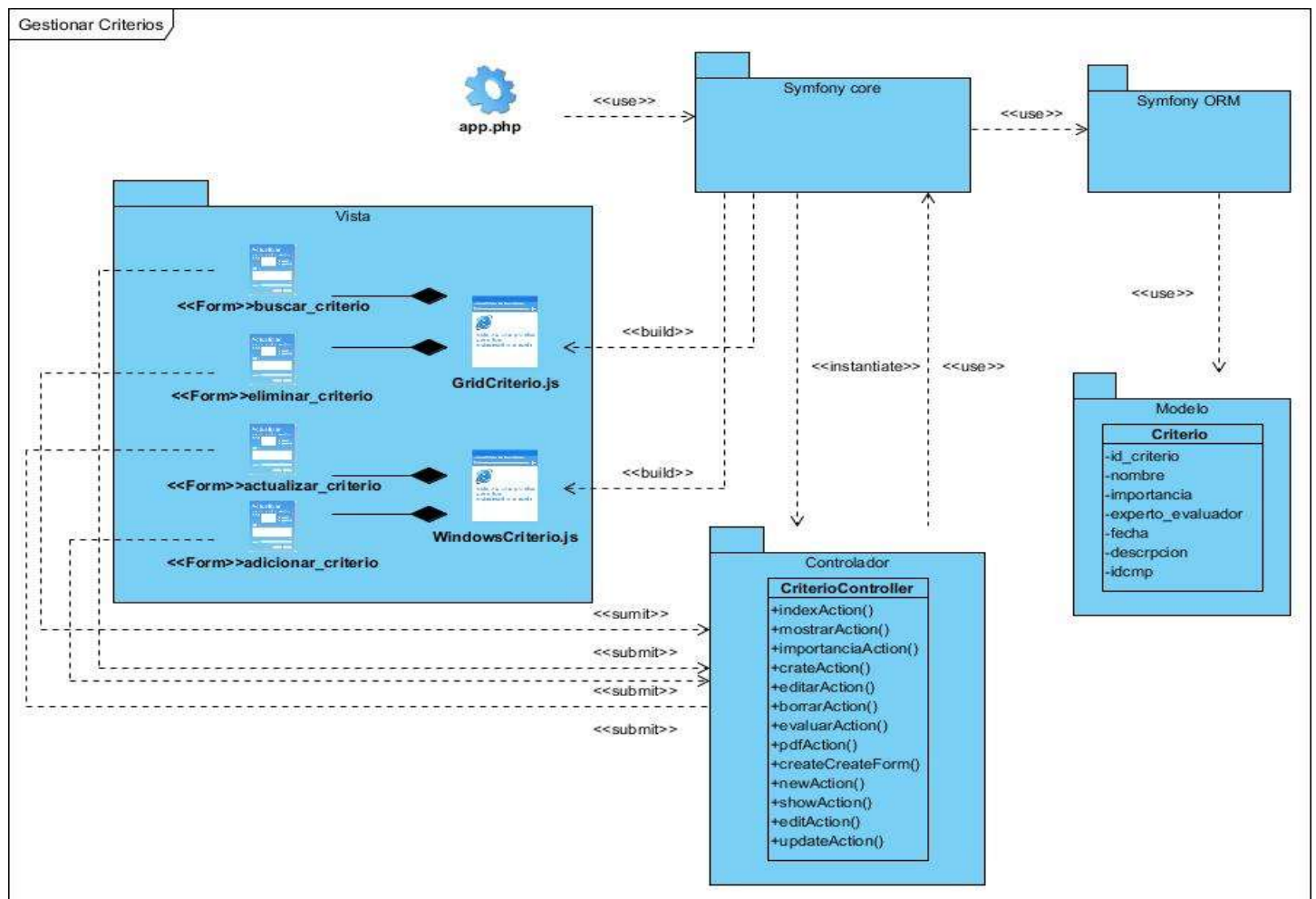


Figura 8: Diagrama de clases del diseño: Gestionar criterios.

3.6. Diagramas de interacción

Los diagramas de interacción representan la forma en cómo un cliente u objeto se comunican entre sí en petición a un evento determinado. Existen dos tipos de diagramas de interacción: diagrama de colaboración y diagrama de secuencia, este último es el empleado para diseñar cómo será el flujo de mensajes entre los objetos. Muestra la interacción de un conjunto de objetos en una aplicación a través del tiempo. Este tipo de diagrama contiene detalles de implementación del escenario, incluyendo los objetos y clases que se usan para implementar el mismo, y los mensajes intercambiados entre los objetos.

A continuación se muestra el diagrama de secuencia para el CU Otorgar importancia.

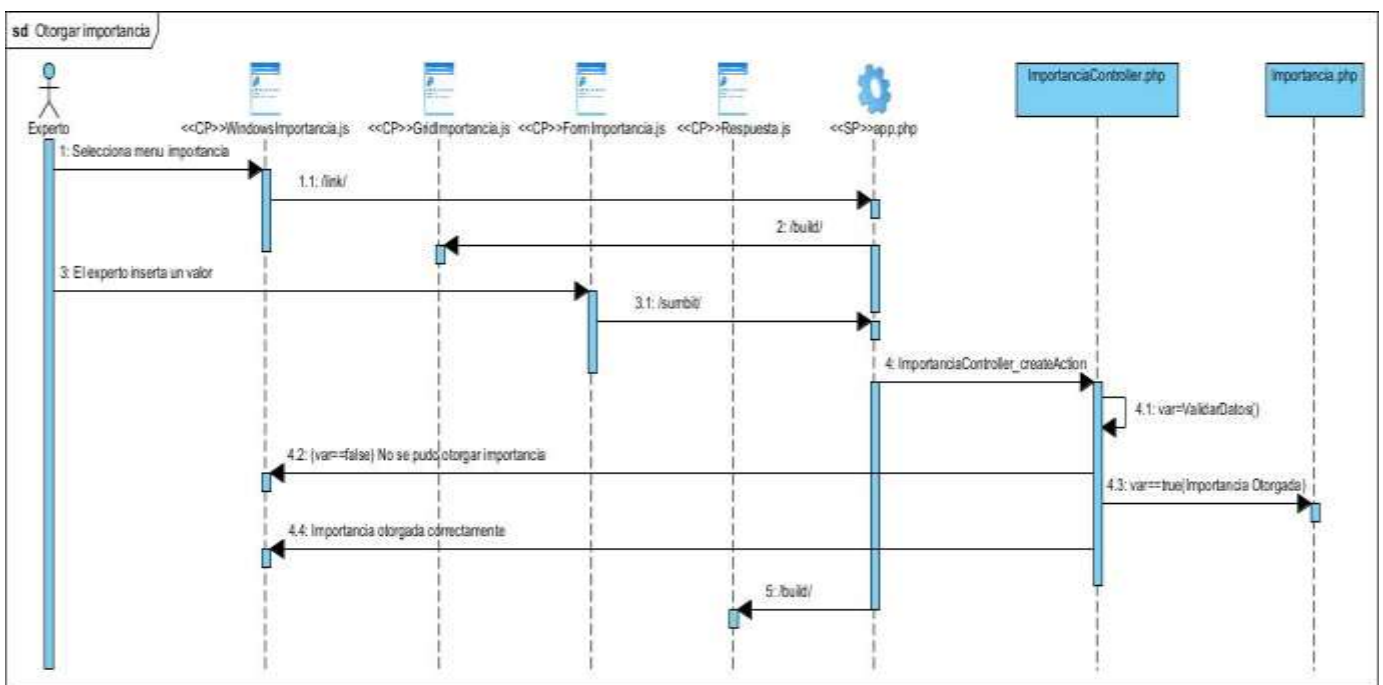


Figura 9: Diagrama de secuencia: Otorgar importancia.

3.7. Diagrama de despliegue

El diagrama de despliegue muestra la configuración de los nodos de procesamiento en tiempo de ejecución, los vínculos de comunicación entre ellos y las instancias de los componentes y objetos que residen en ellos. Está compuesto por nodos, dispositivos y conectores. El propósito del modelo de despliegue es capturar la configuración de los elementos de procesamiento y las conexiones entre estos elementos en el sistema.

PC Cliente: Representa las computadoras clientes que se conectan al servidor de aplicaciones, las mismas se comunican con el servidor a través del protocolo seguro HTTP.

Servidor web: Representa el servidor donde se encuentra instalada la aplicación web. Este accede al servidor de Base de Datos para el manejo de la información mediante el protocolo ADO.

Servidor de Base de datos: Es donde se almacena toda la información de la aplicación.

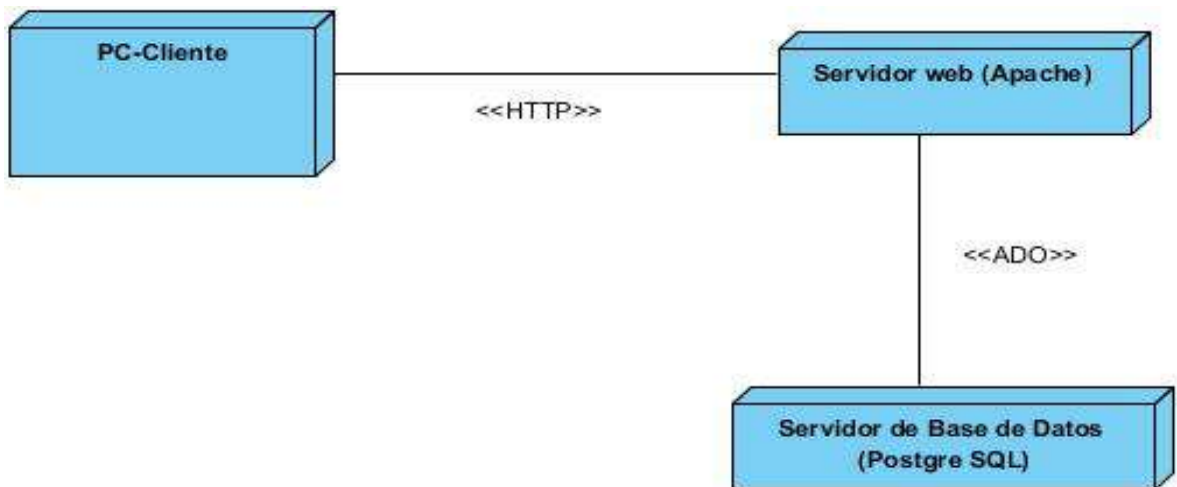


Figura 10: Diagrama de despliegue.

3.8. Diagrama de componentes

Los diagramas de componentes describen los elementos físicos del sistema y sus relaciones. Se realizan con el objetivo de poseer una vista de forma general del sistema a partir de las dependencias e integraciones de los componentes y módulos.

- En el paquete controlador se encuentran las clases controladoras, encargadas de manejar las peticiones de los usuarios a través de los métodos que tienen implementados.
- El paquete modelo agrupa las entidades del sistema, a través de las cuales se realiza el acceso a la base de datos y el paquete vista, agrupa los archivos que permiten visualizar las respuestas que devuelven los controladores al usuario.

A continuación se muestra el diagrama de componentes para el MainBundle, donde se encuentran los principales componentes del sistema a desarrollar.

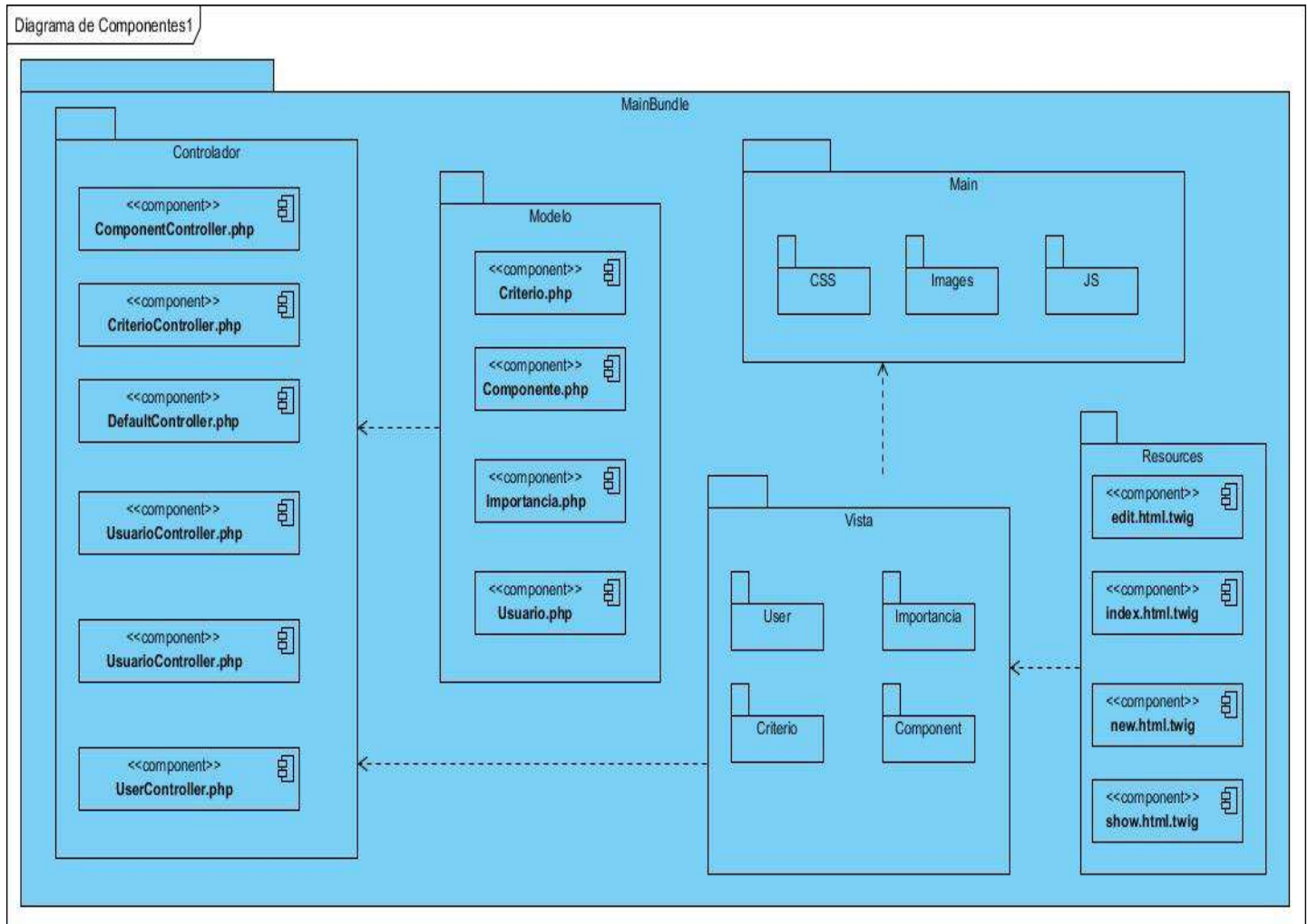


Figura 11: Diagrama de componentes: MainBundle.

Conclusiones del capítulo

La arquitectura definida permitió realizar un diseño robusto y complejo, conforme a las buenas prácticas seguidas por los patrones seleccionados, donde se evidenció la integración de los componentes Modelo Vista Controlador y dichos patrones, mostrando el alto grado de flexibilidad que posee el sistema. Se arribó a que la propuesta de solución, Sistema para determinar el índice de control organizacional, es factible y brinda información valiosa relacionada al control interno de la organización; que permitió consolidar las bases y principios que guiarán el proceso de implementación y estructuración del sistema.

Capítulo 4: Implementación y Análisis de los Resultados

Introducción

La fase de implementación en el desarrollo de un producto de software, es el mecanismo donde se ponen en práctica todas las descripciones y arquitecturas propuestas en las fases de análisis y diseño, es el complemento del trabajo de las fases que lo preceden dentro del proceso de desarrollo de software. La implementación ofrece una materialización precisa de los requisitos.

Una de las últimas fases del ciclo de vida antes de entregar un software para su explotación es la fase de pruebas, cuyo objetivo es comprobar si este cumple sus requisitos. Dentro de la fase de pruebas pueden desarrollarse varios tipos de pruebas en función de los objetivos de las mismas y de su importancia.

4.1. Código fuente

Para obtener una versión funcional de la aplicación se deben implementar los componentes que se han definido, como resultado se obtienen archivos que contienen el código fuente de la aplicación. El código fuente de un software es un conjunto de líneas de texto que son las instrucciones que debe seguir la computadora para ejecutar dicho programa. Por tanto en el código fuente de un programa está escrito su funcionamiento. Estas instrucciones son escritas en un lenguaje de programación que consiste en un conjunto de símbolos, reglas sintácticas y semánticas que definen su estructura y el significado de sus elementos (Lasso, Iván 2008).

4.1.1. Estándares de codificación

Un estándar de codificación completo comprende todos los aspectos de la generación de código. Si bien los programadores deben implementar un estándar de forma prudente, éste debe tender siempre a lo práctico. Un código fuente completo debe reflejar un estilo armonioso, como si un único programador hubiera escrito todo el código de una sola vez.

Para facilitar el entendimiento del código y fijar un modelo a seguir, se establecieron estándares de codificación. A continuación se muestran algunos de estos estándares para el lenguaje PHP utilizados en Symfony2.

4.1.2. Nombres de clases y métodos

Para la definición de las clases y métodos en el código del sistema informático fueron utilizados los estándares de codificación CamelCase y UpperCamelCase. Estos son estilos de escritura que se aplica a frases o palabras compuestas.

```
public function indexAction() { ...9 lines }
public function pdfAction() { ...14 lines }
public function mostrarAction() {
    try { ...12 lines } catch (\Exception $exc) { ...7 lines }
}
public function createAction(Request $request) { ...21 lines }
public function editarAction(Request $request) { ...24 lines }
public function borrarAction() { ...16 lines }
public function evaluarAction() {
```

Figura 12: Estándar de codificación CamelCase.

```
class CriterioController extends Controller {
```

Figura 13: Estándar de codificación UpperCamelCase.

4.1.3. Estructura

- El código debe usar cuatro espacios en vez de usar el tabulado. Esto minimiza problemas con otras herramientas de desarrollo.
- Las líneas deben tener 80 caracteres o menos, evitando tener más de 120 caracteres.
- Las llaves de apertura en las clases deben ir en la siguiente línea y la llave de cierre debe ir en la siguiente línea después del cuerpo.
- Las llaves de apertura, en los métodos de las estructuras de control debe ir en la misma línea y las llaves de cierre deben de ir después del cuerpo.
- Los paréntesis en las estructuras de control no deben usar espacios antes o después.
- Añadir un solo espacio después de cada limitador de coma.
- Añadir un solo espacio alrededor de los operadores (==, &&, ...).
- Usa llaves para indicar el control de la estructura sin tener en cuenta el número de declaraciones que el grupo pueda contener.

- Definir una clase por fichero.
- Declarar las propiedades de clase antes que los propios métodos de clase.

4.2. Pantallas principales de la aplicación

La interfaz de una aplicación permite el flujo de información entre el usuario y el sistema. A continuación se muestran algunos ejemplos de las interfaces del sistema desarrollado:

Nombre	Apellidos	Usuario	Correo	Categoría	Area	Rol
admin	admin	admin	admin@uci.cu	Profesor	Area	Administrador
Jefe	Jefe	jefe	asd@uci.cu	asd	asd	JefeDpto
sergio	morales	smorales	smorales@uci.cu	ing	geysed	Experto
User	User	user	asd	asd	asd	Técnico
Super	ass	super	super@uci.cu	asd	Asd	Administrador
Experto	Experto	experto	asd@uci.cu	asd	asd	Experto
a	a	a	a@uci.cu	a	a	Experto

Figura 14: Página principal de administración: Usuarios.



Figura 15: Página principal de la administración: Componentes.

Validación del sistema

Una vez terminada la implementación del producto que se requiere es necesario realizarle pruebas con el objetivo de detectar errores en la aplicación y la documentación; este proceso resulta de gran importancia ya que da una medida de la calidad del mismo, siempre que se lleve a cabo de la forma correcta. A continuación se muestran las pruebas realizadas al sistema y los resultados obtenidos por cada una.

El proceso de pruebas se centra en los procesos lógicos internos del software, asegurando que todas las sentencias se han comprobado, y en los procesos externos funcionales, es decir, la realización de las prueba para la detección de errores. Además son utilizadas para identificar posibles fallos de implementación, calidad o usabilidad de un programa (Pressman, 2010).

4.3. Niveles de Prueba

Los niveles de prueba son diferentes ángulos de verificar y validar en determinados momentos el ciclo de vida del software. Existen diferentes niveles de pruebas como: pruebas funcionales, de sistema y de aceptación. En el desarrollo de la fase de pruebas del sistema para determinar el índice de control organizacional se aplicarán las pruebas que abarcan el siguiente nivel:

- **Desarrollador:** Se realiza con el objetivo de detectar errores en la implementación de requerimientos.

4.4. Tipos de Prueba

Existen diferentes tipos de pruebas que se pueden aplicar para comprobar el correcto funcionamiento de la aplicación, se seleccionaron los que a continuación se muestran:

- **Pruebas de Funcionalidad:** Se asegura el trabajo apropiado de los requisitos funcionales, incluyendo la navegación, entrada de datos, procesamiento y obtención de resultados.

4.5. Métodos de Prueba

4.5.1. Pruebas funcionales de caja negra

Las pruebas de caja negra, también denominadas pruebas funcionales se centran en los requisitos funcionales del software, es decir, la prueba de caja negra permite al ingeniero de software obtener los conjuntos de condiciones de entrada, que ejerciten completamente todos los requisitos funcionales de un programa.

Por ello se denominan pruebas funcionales, y el probador se limita a suministrarle datos como entrada y estudiar la salida, sin preocuparse de lo que pueda estar haciendo el módulo por dentro.

- Las pruebas funcionales que se realizarán a la solución, estarán enfocadas o dirigidas a los casos de uso del sistema para verificar su correcto funcionamiento.
- En este tipo de pruebas se ejecutarán los distintos servicios prestados con datos correctos e incorrectos; en caso de que los datos sean incorrectos se verificará que los mensajes de error sean los deseados y en el caso opuesto que los resultados sean los esperados (Manas,1994).

A continuación se muestran los casos de prueba Otorgar importancia y Evaluar criterio.

Tabla 8: Caso de prueba: Otorgar importancia.

SC 1 Otorgar importancia

Escenario	Descripción	Experto	Cmp1	Cmp2	Cmp3	Cmp4	Cmp5	Respuesta del sistema	Flujo central
EC 1.1 Otorgar importancia correctamente.	Permite otorgar un valor entero para emitir una importancia.	V	V(20)	V(10)	V(20)	V(20)	V(30)	Se otorga una importancia para poder calcular el índice de control interno y se actualiza el listado de componentes e importancia.	1- Se selecciona la opción "Otorgar importancia". 2- Se llenan todos los campos. 3-Se presiona la opción "Aceptar".
EC 1.2 Cancelar operación: Otorgar importancia.	Se cancela la operación de otorgar importancia.	V	V(10)	V(20)	V(40)	V(20)	V(10)	Se cancela la operación de otorgar una importancia para poder calcular el índice de control interno y se actualiza el listado de componentes e importancia.	1- Se selecciona la opción "Otorgar importancia". 2- Se llenan todos los campos. 3-Se presiona la opción "Cancelar".
EC 1.3 Dejar campos vacíos.	Se inserta una importancia, dejando campos vacíos.	V	I	I	V(20)	V(20)	V(20)	Se muestra un mensaje "Existen campos vacíos".	1- Se selecciona la opción "Otorgar importancia". 2- Se llenan los campos. 3- Se presiona la opción "Aceptar". 4- Se muestra un mensaje de error "Existen campos vacíos".
EC 1.4 Otorgar importancia con un valor 0.	El valor de un campo no puede tener un valor de 0.	V	I(0)	V(10)	V(20)	V(10)	V(10)	Se muestra un mensaje "Existen campos vacíos".	1- Se selecciona la opción "Otorgar importancia". 2- Se llenan los campos. 3- Se presiona la opción "Aceptar". 4- Se muestra un mensaje de error "Existen campos vacíos".
EC 1.5 Otorgar importancia sin que la suma de las importancias sea correcta.	Verifica si la suma de todas las importancias asociadas a cada componente presenta como valor 100.	V	V(10)	V(5)	V(5)	V(40)	V(10)	No se activa la opción "Aceptar", evidenciando que la suma no es la correcta.	1- Se selecciona la opción "Otorgar importancia". 2- Se llenan los campos. 3- Se presiona la opción "Aceptar". 4- Se muestra un mensaje de error "La suma de las importancias tiene que dar 100".

Tabla 9: Caso de prueba: Evaluar criterio.

SC 1 Evaluar criterio

Escenario	Descripción	Evaluación	Respuesta del sistema	Flujo central
EC 1.1 Evaluar criterio correctamente.	Permite evaluar un criterio de la lista de criterios registrados.	V (Bien)	Se adiciona la zona de trabajo, muestra el mensaje "Criterio evaluado correctamente" y se actualiza el listado de criterios.	1- Se selecciona la opción "Evaluar". 2- Seleccionar opción "Evaluación". 3- Seleccionar opción "Aceptar".
EC 1.2 Cancelar operación evaluar criterio.	Se cancela la operación de evaluar criterio.	V (Bien)	Se cancela la operación de adicionar una nueva zona de trabajo y la información no es guardada en la base de datos.	1- Se selecciona la opción "Evaluar". 2- Seleccionar opción "Evaluación". 3- Seleccionar opción "Cancelar".

Tabla 10: Caso de prueba: Gestionar componentes.

SC 1 Actualizar componente

Escenario	Descripción	Nombre	Descripción	Respuesta del sistema	Flujo central
EC 2.1 Editar un componente.	Edita el comité especial en la base de datos.	V (Control)	V (Investigación)	Edita la información del componente, muestra el mensaje "La operación se realizó satisfactoriamente" y se actualiza el listado de componentes.	1- Se selecciona la opción "Componentes". 2- Seleccionar opción "Actualizar". 3- Se presiona la opción "Aceptar".
EC 2.2 Cancelar operación editar componente.	Se cancela la operación de editar comité especial.	V (Control)	V (Investigación)	Se cancela la operación de editar comité especial.	1- Se selecciona la opción "Componentes". 2- Seleccionar opción "Actualizar". 3- Se presiona la opción "Cancelar".
EC 2.3 Editar componentes de manera incorrecta	Verifica si se introdujeron todos los datos del componente a editar de manera valida.	I (Control)	I ()	No se activa la opción "Aceptar", evidenciando la entrada de datos de manera incorrecta.	1- Se selecciona la opción "Componentes". 2- Seleccionar opción "Actualizar". 3- Se presiona la opción "Aceptar". 4- Se muestra un mensaje de notificación.
EC 2.4 Dejar campos vacíos	Verifica que no existan campos vacíos.	I	I	No se activa la opción "Aceptar", evidenciando la falta de datos obligatorios del componente a editar.	1- Se selecciona la opción "Componentes". 2- Seleccionar opción "Actualizar". 3- Se presiona la opción "Aceptar". 4- Se muestra un mensaje de notificación "Campos vacíos".

SC 2 Buscar componente

Escenario	Descripción	Nombre	Descripción	Respuesta del sistema	Flujo central
EC 3.1 Eliminar componente correctamente	Elimina un comité especial previamente seleccionado de la base de datos.	V (Control)	NA (Investigación)	Se retorna el componente buscado.	1- Se selecciona la opción "Componentes". 2- Seleccionar opción "Buscar".
EC 3.2 La búsqueda no devolvió ningún resultado	Se cancela la operación de eliminar comité especial.	V (Control)	NA (Investigación)	Se devuelve la lista vacía.	1- Se selecciona la opción "Componentes". 2- Seleccionar opción "Buscar".

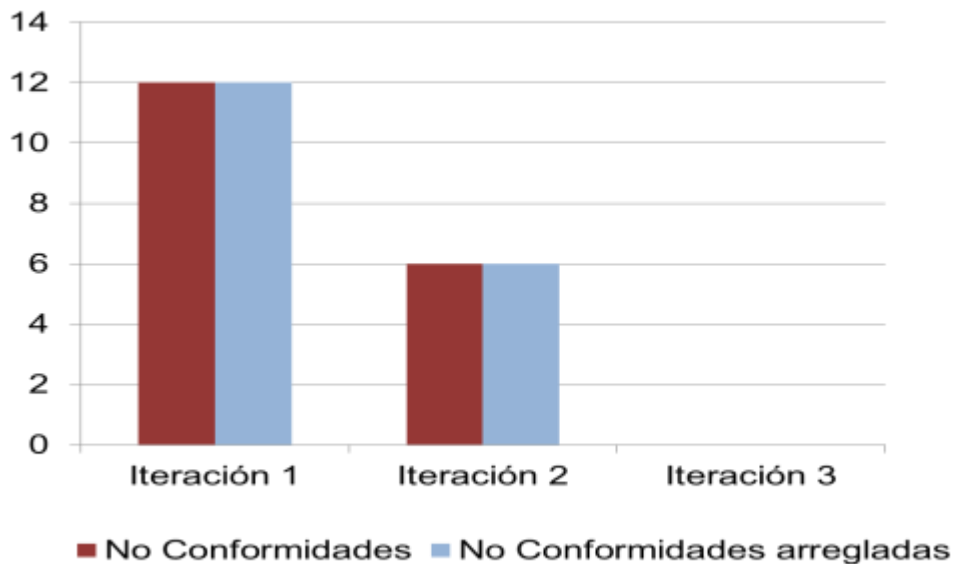


Figura 16: Iteraciones de las pruebas funcionales.

4.6. Pruebas de seguridad

Las pruebas de seguridad tienen como objetivo evaluar la confidencialidad, integridad y disponibilidad de los datos, por lo cual permiten validar que dichos datos o el sistema solamente sean accedidos por los

actores definidos y según sus niveles de acceso. De esta forma se logra aumentar el control de la información sensible del sistema.

Autorización: En esta sección se asegura que cada usuario solo puede realizar las operaciones que le corresponden. En el cumplimiento de las pruebas pertenecientes a esta sección se comprobó que los usuarios autenticados podía modificar su información y la de otros usuarios.

4.7. Demostración de la hipótesis

El presente trabajo de diploma defendió la hipótesis: Con el desarrollo de una aplicación informática, se contribuirá a la trazabilidad de los datos, se eliminaran errores de cálculo y se disminuirá el tiempo de cálculo del índice de control organizacional en la aplicación de la Guía de Autocontrol de la República en la Facultad 6 de la UCI.

En entrevista realizada a los trabajadores del vicedecanato de la Facultad 6 (Ver anexo 1), los cuales están inmersos directamente a la aplicación de la guía de autocontrol, se tuvieron en cuenta tres variables fundamentales para la identificación del problema existente en esta área. Estas variables fueron:

1. Trazabilidad de los datos resultantes de la Aplicación de la Guía de Autocontrol de la República.
2. Errores en el cálculo del índice de control.
3. Tiempo invertido en el cálculo del índice de control.

El sistema obtenido por el autor se centra básicamente en mejorar los indicadores vasados en estas tres variables. A continuación se muestra un estado comparativo de los resultados obtenidos en la entrevista realizada a los trabajadores antes de la implementación del sistema y luego del trabajo durante una semana del sistema implementado (Ver anexo 2).

Tabla 11: Validación de la hipótesis.

Preguntas	Respuestas sin utilizar el sistema	Respuestas utilizando el sistema
¿Se da seguimiento a los resultados obtenidos? ¿Qué seguimiento se le da los	El seguimiento que se le da a los resultados obtenidos es	El seguimiento que se le da a los resultados obtenidos está acorde con lo planteado por la

resultados obtenidos?	pobre.	Guía de Autocontrol de la República.
¿Cómo se identifican los criterios críticos para reforzar acciones en próximas aplicaciones de la guía?	Muy pocas veces se sabe cuáles son los criterios críticos.	Se conoce con exactitud los criterios evaluados de mal.
¿Se calcula el índice de control tras la Guía de Autocontrol de la República?	No se calcula.	Se calcula.
¿Qué tiempo demora el cálculo del índice de control?	Las ocasiones en que se ha intentado calcular, se ha utilizado más de tres jornadas laborales, equivalente a 24 horas de trabajo pero el resultado obtenido no ha sido real por lo que en la actualidad no se realiza.	El tiempo para realizar el cálculo es de aproximadamente un minuto.
¿Cuál es la principal causa de lo planteado anteriormente?	La dificultad de los cálculos matemáticos.	Ninguna.
¿Existen otras razones además de la antes planteada?	Si, los cálculos se realizan manualmente, es decir de manera tradicional y no se lleva una trazabilidad de los datos evaluados.	No.
¿Es confiable el resultado obtenido?	No.	Sí.

¿Se almacena los índices de control obtenidos en cada trimestre para su posterior análisis?	No.	Sí.
¿Los indicadores evaluados en iteraciones anteriores se corrigen de manera correcta?	No, puesto que muchas veces no se sabe cuáles son los indicadores críticos o por corregir.	Sí.

Como queda demostrado con la aplicación del sistema obtenido se logró:

1. **Trazabilidad de los datos:** Los resultados de la aplicación de la Guía de Autocontrol de la República están almacenados para ser consultados a partir de criterios y fechas específicas. Se tiene un reporte solo para los criterios críticos y se brindan datos porcentuales de los resultados obtenidos. Estos reportes pueden ser exportados en formato PDF para su posterior impresión.
2. **Errores de cálculo:** El cálculo del índice de control se realiza a través del modelo propuesto por (Mar Omar, 2014) que define cinco pasos o procedimientos para el cálculo del mismo, sin la presencia de errores.
3. **Tiempo de cálculo:** El tiempo de cálculo del índice de control disminuyó aproximadamente en 23 horas, con 59 minutos.

Componente	Cp	P	P x Cp
Ambiente de control	0.75	0.1418	0.1064
Gestión y prevención de riesgos	0.67	0.2518	0.1687
Actividad de control	0.33	0.2345	0.0774
Información y comunicación	0.5	0.1309	0.0655
Supervisión y monitoreo	0.67	0.2409	0.1614
Índice de Control			0.5794

Figura 17: Tiempo de cálculo del índice de control.

Conclusiones del capítulo

Tras el flujo de implementación y prueba, el sistema quedó desarrollado. Se demostró que la hipótesis investigativa, apoyada en la entrevista, realizada al departamento del Vicedecanato Administrativo de la Facultad 6 es la solución al problema de la investigación científica. Asegurando que la propuesta de solución es un mecanismo fiable para garantizar el control interno, acorde a las exigencias nacionales.

Conclusiones generales

Una vez concluida la investigación referida al proceso de aplicación de la Guía de Autocontrol de la República en la organización, se obtuvo que:

- Estaba marcada la necesidad de un sistema informático que garantice el correcto funcionamiento de las actividades laborales, utilizando la Guía de Autocontrol de la República como principal instrumento y que no era práctico utilizar la herramienta MEYCOR-COSO existente en el mercado, ya que no utilizaban la modelación matemática para garantizar el control interno.
- La generación de los artefactos establecidos por la metodología desarrollo de software Open Up, correspondientes a las fases: Modelado de Negocio, extracción de Requisitos, Análisis, Diseño e Implementación; permitió un mejor entendimiento de las funcionalidades a desarrollar, gracias a la gran documentación que presenta, sirviendo de base para la implementación del sistema informático.
- Luego de someter el software a pruebas funcionales, específicamente pruebas de caja negra, se pudieron identificar algunas no conformidades que poseía el sistema, lo que permitió que fueran solucionadas, en busca de un software más potente y de mayor calidad.
- El sistema informático permitió, en el cálculo del índice de control organizacional, reducir el margen de errores existentes y de tiempo en un 99% y en 23 horas y 59 minutos respectivamente, además la información que brinda el sistema es de vital importancia para desarrollar un seguimiento de los indicadores defectuosos detectados en iteraciones previas.

Recomendaciones

El sistema para determinar el índice de control organizacional, es un proyecto de envergadura e impacto social que requiere de seguimiento e incremento de los servicios que puede brindar. Se recomienda desarrollar la propuesta de solución utilizando otros procedimientos matemáticos basados en el consenso de expertos y las técnicas multicriterio.

Referencias Bibliográficas

BUSCHMANN F, MEUNIER, R., ROHNERT, H., SOMMERLAND, P. y STAL M 1996. Pattern – Oriented Software Architecture. A System of Patterns. Londres.

CARDOSO E, CAMACHO F y NUÑEZ G 2004. Arquitecturas de software. Guia de estudio. . S.I.

CASTRO RAÚL 2009. www.cubadebate.cu. www.cubadebate.cu [en línea]. [Consulta: 10 noviembre 2014]. Disponible en: <https://cubadebate.cu/>.

CONTRALORÍA GENERAL DE LA REPÚBLICA 2009a. Folleto SistemaCI, Artículo 3.

CONTRALORÍA GENERAL DE LA REPÚBLICA 2009b. Folleto Sistema CI, Artículo 6. .

CONTRALORÍA GENERAL DE LA REPÚBLICA 2009c. Folleto SistemaCI, Resolución No. 60/11. La Habana.

CONTRALORÍA GENERAL DE LA REPÚBLICA 2009d. Ley 107, Artículo 83.

DIAS ROBERTO 2010. Propuesta de Sistema Automatizado de Informacion Docente para Tecnologia en Villa Clara. [en línea]. [Consulta: 10 junio 2015]. Disponible en: <http://revedumecentro.sld.cu/index.php/edumc/article/view/70/143>.

ECLIPSE FOUNDATION. 2011. OpenUp Basic. S.I.

Framework-Ecured. 2014 [en línea]. [Consulta: 24 febrero 2015]. Disponible en: <http://www.ecured.cu/index.php/Framework>.

GOBIERNO DE LA REPÚBLICA DE CUBA 2009. Bases para la constitución de la República. [en línea]. S.I.[Consulta: 23 octubre 2014]. Disponible en: <http://www.contraloria.cu/>.

GONZALEZ ENRIQUE 2010. Aprender a Programar.

INFANTE A. 2013. Desarrollo de un prototipo de software para el requerimiento de solicitudes de creditos. Universidad EAN facultad de ingenieria. *Bogota.com*.

JACOBSON 2000. *El Proceso Unificado de Desarrollo de Software*. Madrid: Pearson Educación S.A. ISBN 8478290362.

LASSO IVAN 2008. Que es el código fuente. [en línea]. [Consulta: 17 abril 2015]. Disponible en: [http://www.proyectoautodidacta.com/comics/que es el código fuente](http://www.proyectoautodidacta.com/comics/que%20es%20el%20codigo%20fuente).

LEDO, V. 2012. Gestión de la información y el conocimiento. [en línea], vol. 26, no. 3. [Consulta: 20 diciembre 2014]. Disponible en: <https://EducaciónMédicaSuperior-Gestión-información-conocimiento>.

MANAS, J.A. 1994. Pruebas de programación. [en línea]. [Consulta: 17 abril 2015]. Disponible en: <http://www.it.uc3m.es/tsps/testinghtml>.

MAR OMAR 2014. Procedimiento para determinar el índice de control organizacional. , vol. 18, no. 2.

MARTÍN T. 2013. Modelación y manejo de bases de datos para el almacenamiento de la información sobre ortología genética. *Revista cuba ciencia informática*, vol. 7, no. 3. ISSN 2227-1899.

PRESSMAN, R.S. 2010. *Ingeniería del software, un enfoque práctico*. 5. S.l.: Mc Graw Hill.

SALINAS R 2004. *Enfoque Multicriterio Multiexperto para la Toma de Decisiones*. La Habana: s.n.

SOLÓRZANO P. 2007. Los cinco componentes del Control Interno. [en línea]. [Consulta: 18 noviembre 2014]. Disponible en: <https://www.google.es/#output=search&client=psy-ab&q=fiverdolly+26455>.

W3C 2014. World Wide Web. [en línea]. [Consulta: 4 diciembre 2014]. Disponible en: <http://www.w3c.es/Divulgacion/GuiasBreves/ServiciosWeb>.

Bibliografía

BUSCHMANN F, MEUNIER, R., ROHNERT, H., SOMMERLAND, P. y STAL M 1996. Pattern – Oriented Software Architecture. A System of Patterns. Londres.

CARDOSO E, CAMACHO F y NUÑEZ G 2004. Arquitecturas de software. Guía de estudio.S.I.

CASTRO RAÚL 2009. www.cubadebate.cu. www.cubadebate.cu [en línea]. [Consulta: 10 noviembre 2014]. Disponible en: <https://cubadebate.cu/>.

CONTRALORÍA GENERAL DE LA REPÚBLICA 2009a. Folleto SistemaCI, Artículo 3.

CONTRALORÍA GENERAL DE LA REPÚBLICA 2009b. Folleto Sistema CI, Artículo 6.

CONTRALORÍA GENERAL DE LA REPÚBLICA 2009c. Folleto SistemaCI, Resolución No. 60/11. La Habana.

CONTRALORÍA GENERAL DE LA REPÚBLICA 2009d. Ley 107, Artículo 83.

ECLIPSE FOUNDATION. 2011. OpenUp Basic. S.I.

EGUILUZ JAVIER 2013. Introducción a AJAX.

EGUILUZ JAVIER 2010. Desarrollo web ágil con Symfony 2.3.

FLORES C y ALFERÉZ, G. 2011. Establecimiento de una Metodología de Desarrollo de Software para la Universidad de Navojoa Usando Open UP. México.

Framework-Ecured. [en línea] 2014. [Consulta: 24 febrero 2015]. Disponible en: <http://www.ecured.cu/index.php/Framework>.

GOBIERNO DE LA REPÚBLICA DE CUBA 2009. Bases para la constitución de la República. [en línea]. S.I.: [Consulta: 23 octubre 2014]. Disponible en: <http://www.contraloria.cu/>.

GOLDSTEIN A, WEYL E y LAZIRIS L 2011. HTML5 & CSS3 for the Real World. . S.I.: SitePoint.

- GONZALEZ ENRIQUE 2010.** Aprender a Programar.
- HERNÁNDEZ R A y COELLO S 2002.** *El paradigma cuantitativo de la investigación científica*. La Habana: Editorial universitaria.
- JACOBSON 2000.** *El Proceso Unificado de Desarrollo de Software*. Madrid: Pearson Educación S.A. ISBN 8478290362.
- LARMAN, C. 2009.** *UML y patrones. Una introducción al análisis y diseño orientado a objetos*. 2. S.l.: s.n.
- LOBOS MARIA 2014.** Emagister. www.emagister.com [en línea]. [Consulta: 23 noviembre 2014]. Disponible en:<http://www.emagister.com/curso-aprende-programar/concepto-lenguaje-programacion>.
- LOPERA C, GUTIERREZ E y MARTÍN JC 2003.** *Indicadores: ciencia y tecnología en países de América Latina*. Lecturas de Economía. S.l.: s.n.
- MARIN ALEJANDRA 1998.** Nuevos conceptos del control interno. Valencia.
- MAR OMAR 2014.** Procedimiento para determinar el índice de control organizacional. , vol. 18, no. 2.
- MC GRAW, H. 2001.** *Ingeniería de software. Un enfoque práctico*. 5. España: s.n. ISBN 8448132149.
- MINISTERIO DE FINANZAS 2003.** Resolución del Ministerio de Finanzas y Precios. [en línea]. La Habana: [Consulta: 12 noviembre 2014]. Disponible en: <http://www.contraloria.cu/>.
- NEGRÍN SIMEÓN 1997.** La ciencia y la tecnología en Cuba, *Revista Cubana de Medicina Tropical*. , vol. 49, no. 3, pp. 153-160.
- R. I. DE I. DE CIENCIA 2003.** *Tecnología (Ricyt), Estado de la Ciencia: Principales indicadores de Ciencia y Tecnología Iberoamericanos/Interamericanos*. Buenos Aires.
- SALINAS R 2004.** *Enfoque Multicriterio Multiexperto para la Toma de Decisiones*. La Habana: s.n.
- SANCHEZ R 2002.** *Indicadores de los sistemas de ciencia, tecnología e innovación. Economía industrial*. S.l.: s.n. 343.

ULLMAN LARRY 2007. Php 5 advanced: visual quickpro guide. S.I.

W3C 2014. World Wide Web. [En línea]. [Consulta: 4 diciembre 2014]. Disponible en:
<http://www.w3c.es/Divulgacion/GuiasBreves/ServiciosWeb>.

Anexos

Anexo 1: Entrevista realizada para identificar errores en el proceso de aplicación de la Guía de Autocontrol de la República en la Facultad 6 de la Universidad de las Ciencias Informáticas.

Descripción: El autor realiza la entrevista al departamento del vicedecanato administrativo de la Facultad 6, específicamente a la técnico medio en economía Liudmila de la Osa y al Msc. Omar Mar Cornelio, para detallar el proceso de control interno en la Facultad 6 de la Universidad de las Ciencias Informáticas (UCI), de esta manera se extrae la información que se necesita a través del uso de las técnicas opináticas y la propia experimentación de los implicados (entrevistador y entrevistados), la información de la entrevista está presente en el siguiente trabajo respetando las respuesta de los entrevistados.

Preguntas	Respuestas
¿Se aplica la Guía de Autocontrol de la República en la Facultad 6?	Sí.
¿Con que frecuencia se realiza la aplicación de la Guía de Autocontrol de la República?	Trimestralmente.
¿Quiénes intervienen en la aplicación de la Guía de Autocontrol de la República en la organización?	Los jefes de departamentos, el jefe del vicedecanato administrativo, los técnicos medios en economía y los expertos designados por la Facultad.
¿Quién verifica los resultados plasmados por los jefes de departamento en la aplicación de la Guía de Autocontrol de la República?	El técnico medio de economía del departamento del vicedecanato administrativo de la Facultad 6.

¿Se da seguimiento a los resultados obtenidos? ¿Qué seguimiento se le da los resultados obtenidos?	El seguimiento que se le da a los resultados obtenidos es pobre.
¿Cómo se identifican los criterios críticos para reforzar acciones en próximas aplicaciones de la Guía de Autocontrol de la República?	Muy pocas veces se sabe cuáles son los criterios críticos.
¿Se calcula el índice de control tras la aplicación de la Guía de Autocontrol de la República?	No se calcula.
¿Qué tiempo demora el cálculo del índice de control?	Las ocasiones en que se ha intentado calcular, se ha utilizado más de tres jornadas laborales, equivalente a 24 horas de trabajo pero el resultado obtenido no ha sido real por lo que en la actualidad no se realiza.
¿Cuál es la principal causa de lo planteado anteriormente?	La dificultad de los cálculos matemáticos.
¿Existen otras razones además de la antes planteada?	Si, los cálculos se realizan manualmente, es decir de manera tradicional y no se lleva una trazabilidad de los datos evaluados.
¿Es confiable el resultado obtenido?	No.

¿Cuáles son los problemas más frecuentes para poder llegar a un resultado matemático?	La presencia de errores en los procedimientos matemáticos (margen de error elevado).
¿Se almacena los índices de control obtenidos en cada trimestre para su posterior análisis?	No.
¿Los indicadores evaluados en iteraciones anteriores se corrigen de manera correcta?	No, puesto que muchas veces no se sabe cuáles son los indicadores críticos o por corregir.
¿Qué herramienta auxiliar se utiliza para calcular el índice organizacional en la Facultad?	Ningún sistema, solo se utiliza el Excel y Acces como única ayuda.
¿Es necesario digitalizar estos procesos para un mejor funcionamiento de la aplicación de la Guía de Autocontrol de la República para un seguimiento y análisis de los resultados obtenidos?	Totalmente.

Anexo 2: Cuestionario de la entrevista para validar la hipótesis investigativa.

Descripción: El autor realiza la entrevista al departamento del vicedecanato administrativo de la Facultad 6, específicamente a la técnico medio en economía Liudmila de la Osa y al Msc Omar Mar Cornelio, para validar la hipótesis investigativa, de esta manera se extrae la información que se necesita a través del uso

de las técnicas opináticas y la propia experimentación de los implicados (entrevistador y entrevistados), la información de la entrevista está presente en el siguiente trabajo respetando las respuesta de los entrevistados.

Preguntas	Respuestas
¿Se aplica la Guía de Autocontrol de la República en la Facultad 6?	Sí.
¿Con que frecuencia se realiza la aplicación de la Guía de Autocontrol de la República?	Trimestralmente.
¿Quiénes intervienen en la aplicación de la Guía de Autocontrol de la República en la organización?	Los jefes de departamentos, el jefe del vicedecanato administrativo, los técnicos medios en economía y los expertos designados por la Facultad.
¿Quién verifica los resultados plasmados por los jefes de departamento en la aplicación de la Guía de Autocontrol de la República?	El técnico medio de economía del departamento del vicedecanato administrativo de la Facultad 6.
¿Se da seguimiento a los resultados obtenidos? ¿Qué seguimiento se le da los resultados obtenidos?	El seguimiento que se le da a los resultados obtenidos está acorde con lo planteado por la Guía de Autocontrol de la República.
¿Cómo se identifican los criterios críticos para reforzar acciones en próximas aplicaciones de la Guía de Autocontrol de la	Se conoce con exactitud los criterios evaluados de mal.

República?	
¿Se calcula el índice de control tras la aplicación de la Guía de Autocontrol de la República?	Se calcula.
¿Qué tiempo demora el cálculo del índice de control?	El tiempo para realizar el cálculo es de aproximadamente un minuto.
¿Cuál es la principal causa de lo planteado anteriormente?	Ninguna.
¿Existen otras razones además de la antes planteada?	No.
¿Es confiable el resultado obtenido?	
¿Cuáles son los problemas más frecuentes para poder llegar a un resultado matemático?	Por ahora no.
¿Se almacena los índices de control obtenidos en cada trimestre para su posterior análisis?	Sí.
¿Los indicadores evaluados en iteraciones anteriores se corrigen de manera correcta?	Sí.
¿Qué herramienta auxiliar se utiliza para calcular el índice organizacional en la	Sistema para determinar el índice de

Facultad?	control
¿Es necesario digitalizar estos procesos para un mejor funcionamiento de la aplicación de la Guía de Autocontrol de la República para un seguimiento y análisis de los resultados obtenidos?	Ya se digitalizaron.

