



Universidad de las Ciencias Informáticas
Facultad 6
Trabajo de Diploma para optar por el título de
Ingeniero en Ciencias Informáticas



**Título: Informatización del proceso de Gestión de la Calidad en el Centro
para el Control Estatal de Medicamentos, Equipos y Dispositivos Médicos.**

Autor:

Kenia Macías González

Tutor:

Ing. Jackson Mendoza Romero

Ciudad de la Habana, 8 de julio del 2016.

“Año 58 del Triunfo de la Revolución”

Declaración de Autoría

Declaro que Kenia Macías González es la única autora de la presente tesis que tiene por título: Informatización del proceso de Gestión de la Calidad en el Centro para el Control Estatal de Medicamentos, Equipos y Dispositivos Médicos. El producto desarrollado, las fuentes y cualquier resultado de la investigación son propiedad del Centro para el Control Estatal de Medicamentos, Equipos y Dispositivos Médicos (CECMED).

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

Firma del Autor

Kenia Macías González

Firma del Tutor

Ing. Jackson Mendoza Romero

DATOS DE CONTACTO

Nombre: Ing. Jackson Mendoza Romero

Graduado de Ingeniero en Ciencias Informáticas en el año 2007. Se desempeña como profesor de Algoritmo, Introducción a la Programación 1, Programación 2: Estructuras de Datos y Práctica Profesional 1 y 2 en la Facultad Regional de la UCI “Mártires de Artemisa”. Obtuvo la categoría docente de Profesor Instructor en la Universidad de Pinar del Río. Se ha desempeñado como programador, analista, probador y actualmente Jefe de Grupo de Desarrollo en la Empresa SOFTEL.

Correo Electrónico: yackson@softel.cu

Agradecimientos

Si hoy logré convertirme en toda una profesional, es gracias a muchas personas que han estado conmigo y me han apoyado siempre, dándome fuerzas y convenciéndome de que si se podía y hoy es el día para agradecerle a todos y cada uno de ellos por haber creído en mí.

Quiero agradecerle a:

Primero que todo quiero agradecerles mis padres por haberme guiado siempre, por tanto sacrificio y porque en mi hay un pedacito de ellos, este título es para ustedes.

A mi tutor y a los que no lo son pero que me ayudaron como si lo fueran. Gracias a Jesús y a Regla por su apoyo.

A mis tías Mayra y Hada que desde donde quiera que me estén mirando hoy, se que están orgullosas de mi.

A mis familiares, a los que están y a los que no.

A mis amigas y amigos que lo han sido sin condiciones a Fefa, Abduly, Garnache, Daimir, Lisset, Alena, Carmen, Vania, Yameray, Hiram a todos gracias por poder contar con ustedes.

A mis amistades de Softel que aunque nos conocimos hace poco se ha ganado mi cariño a Jackson, Arianna, Ernesto, el Dany, Víctor y Perriman gracias por todo.

A José A. García Ramírez, Manuel López Pupo, Ninoska por su comprensión y haber confiado en mí.

A mis compañeros de trabajo por el apoyo incondicional a Sánchez, Mones, Julio Cesar, Espinosa, Adina, Yudelkis, Anay, Naranjo, Morris, Miralba, Ana, Ivet, Teresa, Yander, Arce, Quiroga, Pantoja, Núñez, David, Fleites, Miguel e Isalgue.

Dedicatoria

A mis padres, a mi familia en general.

RESUMEN

El Departamento de Gestión de la Calidad del Centro para el Control Estatal de Medicamentos, Equipos y Dispositivos Médicos es el encargado de orientar, organizar y participar en las acciones dirigidas al mejoramiento de la Calidad en la institución. El objetivo de esta investigación es desarrollar una aplicación para gestionar gran parte de las actividades que se realizan en el departamento de forma manual, entre las que se encuentran el registro de las incidencias y la planificación de las auditorías. Para ello se desarrolló un módulo capaz de integrarse al sistema de Gestión de la Información, desarrollado por la empresa Softel, para esta institución. Este debe ser capaz de gestionar y controlar la información relacionada con los Planes de Auditoría y la atención a las Incidencias (quejas, reclamaciones y reconsideraciones) presentadas por los clientes al Departamento de Gestión de la Calidad. Para darle cumplimiento a lo anteriormente planteado se desarrolló una aplicación Web en lenguaje de programación Groovy con framework Grails 2.3.7, Gestor de Base de Datos PostgreSQL 9.2.4 y utilizando RUP como Metodología para guiar el desarrollo de la aplicación. Con el desarrollo de la misma se propone mejorar el desempeño de los especialistas en la planificación de auditorías y en la recepción y tratamiento de Incidencias. Se obtuvo como resultado una aplicación capaz de gestionar las incidencias registradas por los especialistas en ella, que muestra la información necesaria a los especialistas, facilitando el seguimiento de las incidencias disminuyendo el gasto de recursos y mejorando la calidad del proceso.

PALABRAS CLAVES

Sistemas de Gestión de la Calidad, Plan de Auditoría, Incidencias, Calidad.

Abstract

The Department of Management Quality Center for State Control of Drugs, Medical Devices is responsible for guiding, organizing and participating in actions aimed at improving the quality of the institution. The objective of this research is to develop an application to manage most of the activities carried out in the department manually, among which are the recording of incidents and audit planning. This requires a module capable of integrating the system of Information Management, developed by the company Softel, for this institution was developed. This should be able to manage and control information related to the audit plans and attention to incidents (complaints and reconsiderations) submitted by customers to the Department of Quality Management. To comply with the above stated a web application programming language Groovy framework Grails 2.3.7, Database Manager 9.2.4 PostgreSQL and using RUP as a methodology to guide the development of the application was developed. With the development of it aims to improve the performance of specialists in audit planning and the receipt and processing of incidents. An application capable of managing registered by specialists in it, showing the necessary information to specialists, facilitating the monitoring of incidents decreasing spending resources and improving process quality incidents was obtained as a result.

KEYWORDS

Systems Quality Management, Audit Plan, Incident, Quality.

Índice

INTRODUCCIÓN	1
Capítulo 1.....	5
Fundamentación Teórica.....	5
1.1 Conceptos básicos relacionados con la gestión de procesos	5
1.2 Actualidad y tendencia a nivel mundial de los sistemas de Gestión de la Calidad.....	6
1.2.1. Herramientas empleadas en la atención a las quejas a nivel mundial.....	7
1.3. Actualidad y tendencias de los sistemas de gestión de la calidad en Cuba	10
1.4 Necesidad de crear un módulo para la informatización del proceso de Gestión de la Calidad del CECMED	11
1.5 Metodologías, Herramientas y Lenguaje de programación	12
1.5.1 Metodologías de software.....	13
1.5.2 Herramientas CASE	17
1.5.3 Marco de Trabajo o Framework.....	20
1.5.4 Lenguajes de programación utilizados.....	22
1.6 Ambiente de Desarrollo Integrado (IDE)	24
1.7 Sistema Gestor de Base de Datos (SGBD).....	25
1.8 Conclusiones parciales	28
Capítulo 2.....	29
Descripción del Sistema	29
2.1. Objeto de estudio	29
2.2. Modelo de negocio	30
2.2.1. Actores del negocio.....	30
2.2.2 Trabajadores del negocio	31
2.2.3 Diagrama de casos de uso del negocio.....	31
2.2.4 Realización de los casos de uso del negocio.....	31
2.2.5 Diagrama de Clases del Modelo de Objetos	34
2.2.6 Reglas del negocio.....	35
2.3 Requerimientos. Modelo del sistema	35
2.3.1 Especificación de requisitos.....	36
2.3.2 Justificación de los actores del sistema	41
2.3.3 Diagrama casos de uso del sistema.....	42
2.3.4 Descripción textual Caso de Uso del Sistema Gestionar Incidencias.	43
2.4 Conclusiones parciales	48
Capítulo 3.....	49
Diseño, implementación y prueba del sistema	49
3.1 Arquitectura de Grails.....	49
3.2 Arquitectura del sistema	49
3.3 Patrones utilizados.....	50
3.4 Diagrama de clases del diseño	53
3.5 Diagrama de clases persistentes	53
3.6 Diagrama entidad-relación.....	54
3.7 Diagrama de componentes.....	54

3.8 Diagrama de despliegue	56
3.9 Pruebas del Sistema	57
CONCLUSIONES GENERALES	63
RECOMENDACIONES	64
REFERENCIAS BIBLIOGRÁFICAS	65
ANEXO 1: PATRONES DE CASOS DE USO	68
ANEXO 2: DIAGRAMAS DE CLASES DEL DISEÑO	68
ANEXO 3: CLASIFICACIÓN DE LAS INCIDENCIAS	70
ANEXO 4: DESCRIPCIÓN DE LAS VARIABLES	70
ANEXO 5: DESCRIPCIÓN AMPLIADA DE LOS CUN Y LOS CUS	72
ANEXO 6: DISEÑO DE CASOS DE PRUEBA	76
GLOSARIO DE TÉRMINOS	80

ÍNDICE DE FIGURAS

Fig. 1 Organigrama del CECMED	2
Fig. 2 Figura de RUP en dos dimensiones. Se representa el proceso en el que se grafican los flujos de trabajo y las fases y muestra la dinámica expresada en iteraciones y puntos de control.....	14
Fig. 3 Diagrama de casos de uso del negocio.....	31
Fig. 4 Diagrama de actividades CUN Presentar Incidencia.	34
Fig. 5 Diagrama de Clases del Modelo de Objetos para el trabajador del negocio Especialista DGC.	35
Fig. 6 Diagrama de Clases del Modelo de Objetos para el trabajador del negocio Jefe DGC.	35
Fig. 7 Diagrama de Clases del Modelo de Objetos para el trabajador del negocio Jefe DGC.	43
Fig. 8 Arquitectura de Grails (27)	49
Fig. 9 Modelo-Vista-Controlador	52
Fig. 10 Diagrama de clase del diseño CU Nueva Incidencia	53
Fig. 11 Diagrama de clases persistentes.....	54
Fig. 12 Diagrama Entidad-Relación.....	54
Fig. 13 Paquete de Componentes.....	55
Fig. 14 Paquete Vistas.....	55
Fig. 15 Paquete Controladores.	55
Fig. 16 Paquete Modelo.....	56
Fig. 17 Paquete Conf.	56
Fig. 18 Diagrama de Despliegue.....	56
Fig. 19 Patrón CRUD.....	68
Fig. 20 Patrón Login.....	68
Fig. 21 Diagrama de clases del diseño Asignar Incidencia.....	68
Fig. 22 Diagrama de clases del diseño Editar Incidencia	69
Fig. 23 Diagrama de clases del diseño Evaluar Incidencia.....	69
Fig. 24 Diagrama de clases del diseño Reportar Incidencia.....	70

ÍNDICE DE TABLAS

Tabla 1 Descripción del actor del negocio.....	30
Tabla 2 Descripción de los trabajadores del negocio.	31
Tabla 3 Especificación del caso de uso Presentar Incidencia.	31
Tabla 4 Justificación de los actores del sistema.....	41
Tabla 5 Descripción textual del Caso de Uso del sistema Gestionar Incidencias.	43
Tabla 6 Diseño de casos de prueba Autenticar Usuario.....	59
Tabla 7 Diseño del caso de prueba Nueva Incidencia.....	59
Tabla 8 Descripción textual del CUS Reportar Incidencias.	72
Tabla 9 Descripción textual del CUS Evaluar Incidencias	73
Tabla 10 Descripción textual del CUS Enviar Notificaciones.....	75
Tabla 11 Descripción textual del CUS Mostrar Gráfica del Total de Incidencias VS Clasificación por año actual.....	75
Tabla 12 Descripción textual del CUS Mostrar Gráfica del Total de Incidencias por Clasificación vs Estado.	76
Tabla 13 Caso de prueba Cancelar Incidencia.....	76
Tabla 14 Caso de prueba Buscar Incidencia.....	77
Tabla 15 Caso de prueba Editar Incidencia.....	77

Introducción

El CECMED es una Unidad Presupuestada, subordinada al Ministerio de Salud Pública, creada por la Resolución No. 263, de 11 de mayo de 2011 dictada por el Ministro Salud Pública. Se creó por la fusión de las Unidades Presupuestadas denominadas Buró Regulatorio para la Protección de la Salud, el Centro para el Control Estatal de Calidad de los Medicamentos y el Centro de Control Estatal de Equipos Médicos; todas subordinadas al Ministerio de Salud Pública como Autoridad Reguladora de Medicamentos en Cuba (ARM). (1)

Su misión es promover y proteger la salud de la población mediante un sistema de regulación, fiscalización y vigilancia sanitaria eficaz y transparente, que asegure medicamentos, equipos, dispositivos médicos, servicios y otros productos para la salud y consolidarse como una Autoridad Nacional Reguladora de Medicamentos, Equipos y Dispositivos Médicos reconocida por sus homólogos internacionales.

Este centro cuenta con varias direcciones y departamentos entre los que se encuentra el DGC donde la gran parte del flujo de trabajo es realizado de forma manual y algunos procesos son gestionados con herramientas informáticas que no cubren todas las necesidades del CECMED, lo que ha generado deficiencias en el correcto funcionamiento del mismo, por lo cual, se hace imprescindible la informatización de este departamento, estableciéndose con dicho propósito las relaciones de trabajo y de colaboración con la empresa cubana de desarrollo de software “SOFTEL”.

Como se muestra en la figura #1 la estructura del CECMED se divide en:

- Direcciones.
- Departamentos.
- Departamentos Independientes.
- Grupos.

Los servicios que oferta la entidad son realizados en las siguientes Subdirecciones:

- General.
- De Medicamentos y Diagnosticadores.
- De Fiscalización.
- De Equipos y Dispositivos Médicos.
- Departamento Independiente de Economía.

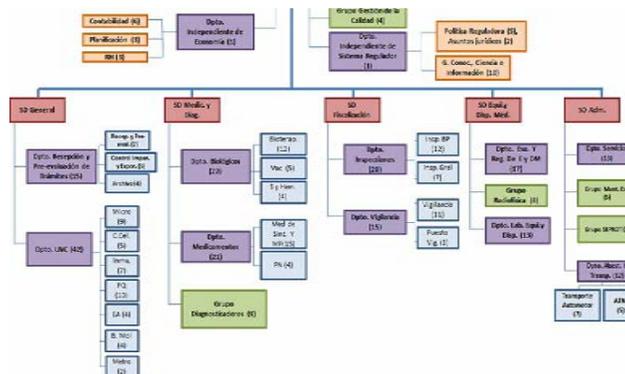


Fig. 1 Organigrama del CECMED

Esta institución cuenta con varios departamentos entre los que se encuentra el DGC que tiene como función orientar, organizar, coordinar y participar, en las acciones dirigidas a la implantación y mejoramiento de la Calidad del CECMED.

En este departamento hay una gran cantidad de procesos que no se le da seguimiento y no se auditan según las normas establecidas lo que provoca que no se corrijan las no conformidades detectadas durante las auditorías realizadas. Además, a los especialistas se les dificulta poder prevenir las decisiones incorrectas o inconsistentes y lograr optimizar la toma de decisiones en tiempo real.

El DGC es el encargado de recepcionar todas las Incidencias (quejas, reclamaciones y reconsideraciones) presentadas por los clientes al CECMED, estas son archivadas en los registros del DGC en correspondencia con su clasificación para asignarlas a los especialistas encargados de la investigación de las mismas. Luego se procede a entregar la incidencia al Jefe del Departamento involucrado en el comunicado, quien lo asigna al especialista evaluador encargado de realizar la investigación y darle respuesta al DGC, el cual se responsabiliza de comunicarle al cliente los resultados obtenidos en la investigación. El CECMED dará respuesta a las quejas en un período no mayor de 31 días naturales, 45 días para las reclamaciones y 60 para las reconsideraciones.

Otras de las funciones del DGC es la planificación anual de las auditorías las cuales son realizadas en Excel por el Jefe del DGC, y luego son enviadas por correo electrónico al Director del CECMED quien se encarga de aprobarlos. Posterior a su aprobación los planes son enviados a los auditores involucrados, los cuales en ocasiones no reciben la información a tiempo, corriéndose el riesgo de que no se cumpla eficazmente con el plan de auditoría.

Una vez concluidas las auditorías no son elaborados según los documentos rectores los informes correspondientes y no se reflejan adecuadamente las no conformidades y observaciones detectadas, lo que conlleva que no se formulen estrategias o acciones correctivas para mitigar las mismas. Lo cual afecta la calidad de los servicios que se ofrecen y las funciones del centro, generándose las incidencias (quejas, reclamaciones y reconsideraciones) por parte de los clientes de la institución.

De ahí, la necesidad imperiosa, de la implementación de un módulo para Informatización del proceso de Gestión de la Calidad en el CECMED, mediante el cual se elabore, controle y concilie la planificación anual de todas las auditorías que se deben implementar en los departamentos y direcciones de la entidad. Además de gestionar la información referente a las incidencias presentadas, por los clientes del CECMED.

A raíz de esta problemática, se plantea para la presente investigación como **problema a resolver**: ¿Cómo contribuir a mejorar el control del proceso de Gestión de la Calidad en el CECMED?

Como **objetivo general** se propone informatizar el proceso de Gestión de Incidencias como parte del proceso de Gestión de la Calidad en el CECMED.

Definiendo como **objeto de estudio** el proceso de gestión de la calidad enmarcado en el **campo de acción** la informatización del proceso de Gestión de la Calidad en el CECMED.

Para desglosar el objetivo general, se definieron los **objetivos específicos**:

- Realizar el análisis y diseño de la solución propuesta.
- Implementar la solución, de acuerdo a la arquitectura definida.
- Realizar las pruebas necesarias para encontrar errores y verificar que se implementaron los requisitos funcionales acordados.

Para dar cumplimiento a los objetivos específicos se definieron **las siguientes tareas de la investigación**:

- Detectar los principales problemas existentes en el Departamento de Gestión de la Calidad. (Este diagnóstico se hace a través de la entrevista al cliente).
- Investigar sobre los sistemas similares a la solución y los conceptos teóricos relacionados con el tema.
- Definir las metodologías, herramientas y tecnologías a utilizar para desarrollar el sistema.
- Definir los requisitos funcionales y no funcionales para la aplicación a desarrollar.

- Elaborar del diseño de la propuesta de solución y sus características.
- Realizar del prototipo de interfaz de usuario para facilitar la implementación de la aplicación a desarrollar.
- Implementar las funcionalidades definidas para la solución.

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

Capítulo 1: “Fundamentación teórica”

Donde se incluyen los resultados del estudio sobre el estado del arte de las auditorías a nivel internacional y nacional; así como el análisis de las principales metodologías, tecnologías y herramientas a utilizar en el desarrollo de la aplicación y los conceptos técnicos que se van a tratar para una comprensión fácil de este trabajo de diploma.

Capítulo 2: “Características del Sistema”

Breve descripción del problema. Estudio de los principales procesos que se llevan a cabo en el Departamento de Gestión de la Calidad a través del Modelo del negocio, principalmente realizaciones de casos de uso del negocio. Definición de requisitos funcionales y no funcionales descritos a través de diagramas de casos de uso del sistema y descripciones textuales de casos de uso del sistema.

Capítulo 3: “Diseño, implementación y prueba del sistema.”

Aborda las actividades definidas para el diseño del sistema siguiendo la metodología RUP, se elaboran los diagramas de clases del diseño teniendo presente los patrones a utilizar y se especifica la arquitectura del sistema. Además se definen los diagramas de componentes y despliegue, se presentan los resultados de las pruebas realizadas al sistema implementado, para verificar que cumple con los requisitos y así comprobar sus funcionalidades.

Capítulo 1

Fundamentación Teórica

Introducción

En el presente capítulo se realiza el estudio teórico que fundamenta la investigación. Se realiza un estudio de la evolución hasta la actualidad de los procesos de gestión de calidad en el mundo y las tendencias que han surgido durante su desarrollo. Este estudio se enfoca a en el ámbito mundial y nacional. Se incluye además el análisis de las tecnologías, herramientas y metodologías empleadas en la solución.

1.1 Conceptos básicos relacionados con la gestión de procesos

El término **gestión** es ampliamente empleado en este capítulo por su importancia en la investigación. Por **gestión** se referirá a la acción y al efecto de administrar o gestionar un negocio. A través de una gestión se llevarán a cabo diversas diligencias, trámites, los cuales, conducirán al logro de un objetivo determinado o de un negocio.

Otros conceptos que vienen a complementar lo anteriormente mencionado son **procesos y gestión de procesos**. La dirección de una empresa o institución debe dotar a la organización de una estructura que permita cumplir con la misión y la visión establecidas. Cualquier actividad o conjunto de actividades ligadas entre sí, que utiliza recursos y controles para transformar elementos de entrada (especificaciones, recursos, información, servicios, etc.) en resultados (otras informaciones, servicios, etc.) puede considerarse como un **proceso**. (2)

La implantación de la **gestión de procesos** se ha revelado como una de las herramientas de mejora de la gestión más efectivas para todos los tipos de organizaciones. Se basa en el control constante de la capacidad de cada proceso, la mejora continua, la flexibilidad estructural y la orientación de las actividades hacia la plena satisfacción de todas las partes interesadas (clientes, accionistas, personal, proveedores, sociedad en general) y sus necesidades. Su propósito final es asegurar que todos los procesos de una organización se desarrollan de forma coordinada, mejorando así su efectividad. Es uno de los mecanismos más efectivos para que la organización alcance altos niveles de eficiencia. (2)

Cuando se habla de la satisfacción del cliente, siempre va estar incluida la **calidad** ya sea de un producto o un servicio. La **calidad** se refiere al conjunto de propiedades inherentes a un objeto que le confieren capacidad para satisfacer necesidades implícitas o explícitas. Por otro lado, **la calidad de un producto o**

servicio es la percepción que el cliente tiene del mismo y su capacidad para satisfacer sus necesidades. (3)

Actualmente existen numerosos **sistemas** que se encargan de la **gestión de la calidad**. Se puede definir como **sistema** al conjunto de elementos o componentes que interactúan entre sí para cumplir ciertas y determinadas metas. Los **sistemas de gestión de la calidad** son un conjunto de normas y estándares internacionales que se interrelacionan entre sí para hacer cumplir los requisitos de calidad que una empresa requiere para satisfacer los requerimientos acordados con sus clientes a través de una mejora continua, una manera ordenada y sistemática. (4)

1.2 Actualidad y tendencia a nivel mundial de los sistemas de Gestión de la Calidad

Los primeros estudios sobre la calidad se hicieron en los años 30 antes de la Segunda Guerra Mundial en Estados Unidos. Para lograr elevar la calidad se crearon las primeras normas de calidad del mundo mediante el concepto moderno del aseguramiento de la calidad, para lograr un verdadero control de calidad se ideó un sistema de certificación de la calidad que el ejército de Estados Unidos inició desde antes de la guerra. Las primeras normas de calidad norteamericanas funcionaron precisamente en la industria militar y fueron llamadas las normas Z1. Las normas Z1 fueron de gran éxito para la industria norteamericana y permitieron elevar los estándares de calidad dramáticamente, evitando así el derroche de vidas humanas trazándose así el camino hacia la ISO. (5)

ISO (Organización Internacional para la Estandarización)

La ISO (Organización Internacional de Normalización) es la fuente de las familias de normas de gestión de calidad y ambiental ISO 9000 e ISO 14000, así como de alrededor 15000 Normas Internacionales para los negocios, el gobierno y la sociedad. La ISO es una red de institutos de normas nacionales de 146 países que trabajan en sociedad con organismos internacionales, en los cuales están representados los gobiernos, la industria, los negocios y los consumidores. (6)

La ISO como organización no otorga directamente la certificación, sino que lo hacen organismos de acreditación certificados. El proceso de certificación se lleva a cabo mediante un proceso de auditorías, llevadas a cabo por auditores externos e internos a la compañía en cuestión; los procesos de auditorías están normados asimismo, por la ISO 19011. La empresa se debe preparar para su certificación, asegurando de que todos sus procesos se ajusten a los requerimientos de la norma. (7)

Todos los SGC se encuentran normados bajo el organismo internacional no gubernamental llamado **ISO** International Organization for Standardization (Organización Internacional para la Estandarización). Estos contribuyen a la mejora de las organizaciones mediante el asesoramiento en materia de calidad, que se basa, entre otros aspectos en: la definición de políticas y estrategia centrada en los clientes y grupos de interés, análisis y comprensión de las necesidades y expectativas de los clientes, soluciones en la gestión de los recursos humanos, gestión de procesos, establecimiento y entrenamiento de equipos de trabajo, de mejora y de procesos, mediciones del progreso y de la calidad a través de Sistemas de Indicadores de Gestión y Cuadro de mando. (8)

Se han identificado ocho **Principios de gestión de la calidad** que pueden ser utilizados por la Dirección de una empresa, con el fin de conducir a la organización hacia una mejora en el desempeño. Estos ocho principios se derivan de la experiencia colectiva y el conocimiento de los expertos internacionales (que participan en el Comité Técnico responsable de desarrollar y mantener actualizadas las normas) y constituyen la base de las normas de SGC de la familia ISO 9000 ellos son: enfoque al cliente, liderazgo, compromiso del personal, enfoque a procesos, enfoque a la gestión, mejora continua, toma de decisiones basada en hechos, relaciones mutuamente beneficiosas con los proveedores. (9)

La variedad de software, para la gestión de la calidad, en el mundo actual es difícil de especificar. Cada empresa o negocio tiene características y procesos determinados dentro de sus flujos de trabajo que requieren funcionalidades diferentes en cada sistema empleado. (10) Existen herramientas especializadas en las necesidades de varias ramas de la economía, sin embargo, en ocasiones resultan demasiado genéricas para ser implantadas en una empresa con procesos específicos, sumándole a esto que la mayoría son consideradas software propietario por lo que habría que tener en cuenta el costo de obtención.

Surge entonces la necesidad de continuar la producción de estos sistemas a ser implantados en las empresas que se animen a la automatización, para lo que se debe estudiar a fondo el flujo de los procesos que la misma sustenta y a partir de los resultados arrojados por estos estudios elaborar una propuesta para el sistema a desarrollar que se ajuste a las necesidades y expectativas del cliente.

1.2.1. Herramientas empleadas en la atención a las quejas a nivel mundial

El estudio de herramientas para la atención a las incidencias, existentes facilita la decisión a la hora de seleccionar cuáles emplear en un caso determinado o cuáles se ajustan a las necesidades de un cliente en específico. Algunas de estas herramientas son empleadas actualmente por grandes empresas y

organizaciones con una amplia aceptación dada, principalmente, por los servicios que brinda y las características que implementa.

QACTION

Controla las quejas de clientes, productos o servicios no conformes, acciones correctivas y preventivas. Automatiza la metodología de las ocho disciplinas facilitando la resolución de problemas y el ciclo de mejora continua, las funciones que lleva a cabo son: planificar las auditorías al sistema de calidad, generar de manera automática hojas de verificación, emitir carta predeterminada de notificación de auditoría, registrar problemas o no conformidades encontradas, planificar acciones correctivas y preventivas para cada problema, incorporar planes de auditoría escritos con cualquier procesador de textos, verificar y registrar la implantación y efectividad de las acciones correctivas efectuadas, emitir solicitud de acción correctiva y preventiva, controlar las quejas de clientes.

Este software cumple con los requisitos de las normas ISO 9000 y QS 9000 y tiene como desventaja ser un software propietario. Para su adquisición es necesaria la compra de varias licencias como la principal, que tiene un costo de \$900,00 (USD), la empresarial, cuyo monto asciende a \$3 470,00 (USD) y la corporativa asciende a \$9000,00 (USD). (11)

SE Audit

SE Audit es un software planificado para ayudar a las organizaciones a administrar la amplia gama de actividades, datos y procesos relacionados con las auditorías en un ambiente único y amplio. Proporciona la flexibilidad de soportar todos los tipos de auditoría, incluyendo auditorías internas, operacionales, de proveedores, de riesgos/controles y auditorías de calidad. El sistema también ofrece funcionalidades para administrar el ciclo de vida completo de una auditoría incluyendo planificación, programación, preparación, desarrollo de planes de auditoría estándar, checklists, colecta de datos, ejecución, informes, monitoreo y el acompañamiento de las acciones correctivas y/o preventivas implementadas.

La solución de SoftExpert para gestión de auditorías también ofrece recursos avanzados, como workflows flexibles para no conformidades, monitoreo del tiempo de ejecución, notificaciones y alertas a través de e-mail y metodologías de gestión de riesgos, permitiendo que las organizaciones implementen las mejores prácticas para la ejecución de auditorías eficientes. Garantiza la integración del proceso de auditoría con el sistema de gestión de riesgos y conformidad. SE Audit asegura los controles necesarios para atender

los requisitos establecidos en normas internacionales de la calidad como: ISO 9000, QS 9000, ISO 14000, etc. (12)

WILSOFT

Software diseñado para asegurar la calidad y resolver problemas críticos de las empresas realizando funciones como: control de documentos, acciones correctivas y preventivas, auditorías, quejas de cliente, informes de producto no conforme, indicadores gerenciales, control de registros y procesos, satisfacción de clientes. (13)

Herramienta de Software Libre (gratuitas) para los SGC.

Administrar un **Sistema de Gestión de Calidad** puede ser una tarea complicada. Sobre todo para pequeñas y medianas empresas (conocida también por el acrónimo PYME) las cuales muchas veces no cuentan con los recursos necesarios. (14)

KMKey Quality

Es el software de gestión de calidad ideal para la implantación y la gestión de la calidad en una empresa u organización. Mediante su uso sencillo e intuitivo, su acceso universal, y su entorno colaborativo, hará más fácil la tarea de gestionar la calidad de una forma eficiente. Una excelente herramienta que nos facilita la gestión de nuestro sistema de calidad es **KMKey Quality**. Esta herramienta realiza funciones tales como: mapa de procesos: Permite definir la organización por procesos y dibujar de forma automática un diagrama (Mapa de procesos), integra un sistema de notificaciones internas o por e-mail que facilita la comunicación a los diferentes usuarios avisando de las acciones que debe realizar respecto a cada documento, auditorías: Generar planes de auditoría, lanzar programación de las auditorías, gestionar y controlar su realización, relacionar con los hallazgos y no Conformidades que se detecten, no conformidades: Permite registrar y realizar el tratamiento, seguimiento y cierre de las no conformidades recibidas en el SGC, quejas, reclamaciones y sugerencias: permite registrar y realizar el tratamiento, seguimiento y cierre de las reclamaciones y quejas recibidas. (15)

Existen herramientas de gestión de información especializadas en las necesidades de varias ramas de la economía, sin embargo, en ocasiones resultan demasiado genéricas para ser implantadas en una empresa con procesos específicos, sumándole a esto que la mayoría son consideradas software propietario por lo que habría que tener en cuenta el costo de obtención. Surge entonces la necesidad de emprender el desarrollo de sistemas que serán desplegadas en las empresas que se animen a la automatización de sus procesos. Por tanto se necesita estudiar a fondo el flujo de las actividades que se

llevan a cabo en el DGC del CECMED y a partir de los resultados arrojados elaborar una propuesta que se integre con el sistema desplegado en dicha institución, desarrollado por la empresa Softel quien además se encarga del mantenimiento del mismo

1.3. Actualidad y tendencias de los sistemas de gestión de la calidad en Cuba

Actualmente existen varios factores que propician la implantación de los sistemas de Gestión de la Calidad en Cuba, en función del desarrollo social y económico que emprende el país. Se destacan los siguientes:

Oficina Nacional de Normalización

La Oficina Nacional de Normalización, reconocida como NC, es el Organismo Nacional de Normalización de la República de Cuba que representa al país ante las Organizaciones Internacionales y Regionales de Normalización. La NC está dirigida a coadyuvar a la organización y fortalecimiento de las actividades de normalización, metrología y calidad en correspondencia con las necesidades de la economía nacional y de las tendencias y requisitos internacionales actuales.

Principales resultados alcanzados en el país

Como miembro pleno de la Organización Internacional de Normalización (ISO) la NC participa activamente en sus distintos comités técnicos y órganos de política, así como en los trabajos técnicos a través de los expertos propuestos por los Comités Técnicos de Normalización. Dicho organismo representa a Cuba ante la Organización Internacional de Normalización (ISO), la Organización Internacional de Metrología Legal (OIML), la Comisión del Codex Alimentarius (CODEX) donde la NC es el punto de contacto ante esta organización gubernamental lo que nos permite promover la adopción de medidas que garanticen la calidad de los alimentos para el consumo humano, la Conferencia General de Pesas y Medidas (CGPM) entre otras. (16)

Los trabajos de normalización nacional, los procedimientos de evaluación de la conformidad (certificación, inspección de la calidad, ensayos, entre otros) y el Servicio Nacional de Metrología (SENAMET) se desarrollan a partir de los principios y recomendaciones de las organizaciones internacionales y regionales referidas, de manera particular de las derivadas de la OMC (Organización Mundial del Comercio). (17)

El ONARC (Órgano Nacional de Acreditación de la República de Cuba) es el encargado de desarrollar y aprobar la evaluación y reconocimiento de la competencia de las entidades que llevan a cabo la evaluación de la conformidad, entre las que se incluyen los laboratorios de ensayo y de calibración, así como los órganos de inspección, con el objetivo de coadyuvar al reconocimiento nacional e internacional de las actividades que se desarrollan en materia de evaluación de la conformidad. (18)

1.4 Necesidad de crear un módulo para la informatización del proceso de Gestión de la Calidad del CECMED

El estudio presentado hasta el momento ha dejado ver que actualmente en el mundo existen varios sistemas de gestión de la calidad que cuentan con una amplia gama de funcionalidades y servicios. Además, se ha podido concluir que por su gran variedad pueden ser empleados en diversas ramas de la economía incluso pueden ser modificados de acuerdo a las necesidades de la empresa. Se adicionan los resultados del estudio de las condiciones satisfactorias existentes en el país para la implantación de sistemas de gestión de la calidad.

Entonces cabe la interrogante: ¿por qué se plantea la necesidad de desarrollar un módulo para la informatización del proceso de Gestión de la Calidad del CECMED? La respuesta está dada por diversos factores.

Primeramente, se debe mencionar que el flujo de procesos que se llevan a cabo actualmente en el DGC del CECMED posee características específicas que no son satisfechas por ninguno de los sistemas de gestión de la calidad existentes. Esto, en gran medida, se debe al uso de planillas y registros definidos por la empresa para el desarrollo de las actividades y procesos del departamento que se han venido definiendo en base a las necesidades y experiencias que han surgido en el transcurso de los años. Estas planillas constan de un formato específico consecuente con la información que se maneja en el departamento.

A esto se adiciona que el sistema a implementar deberá contar con gestión de avisos, así como otras funciones que son definidas únicamente por las características y necesidades del departamento. Lo anteriormente mencionado está relacionado con las funcionalidades que debe presentar el sistema. Además el CECMED requiere de un software que sea capaz de integrarse con el sistema que le está desarrollando SOFTEL para la informatización de sus servicios y funciones y necesitan tener disponible un buen servicio de soporte.

Sin embargo, también existen razones relacionadas con las propiedades o cualidades que debe presentar el producto final y que ya están predefinidas por la arquitectura general que rige el proyecto al cual pertenece el módulo a desarrollar. Esta arquitectura ha establecido los patrones y estilos arquitectónicos, tecnologías y herramientas, a emplear en la construcción del producto. En la siguiente sección se mencionan varias tecnologías incluyendo las empleadas en la solución del problema junto con sus principales características.

1.5 Metodologías, Herramientas y Lenguaje de programación

La arquitectura de un software puede concebirse como aquella estructura del programa que cohesiona las funcionalidades más críticas y relevantes (necesarias para el sistema), y que sirve de soporte al resto de funcionalidades finales (necesarias para el usuario). Una definición más formal: “La arquitectura del software es la organización fundamental de un sistema formada por sus componentes, las relaciones entre ellos y el contexto en el que se implantarán, y los principios que orientan su diseño y evolución”. (19)

Se puede incluir que aporta una visión abstracta de alto nivel, posponiendo el detalle de cada uno de los módulos definidos a pasos posteriores del diseño. Es una vista estructural de alto nivel; define un estilo o combinación de estilos para una solución; centra su atención en los requerimientos no funcionales, ya que los requerimientos funcionales se satisfacen mediante el modelado y diseño de la aplicación. Para que la arquitectura se convierta en una herramienta útil dentro del desarrollo y mantenimiento de los sistemas de software es necesario que se cuente con una manera precisa de representarla.

Para la solución se sigue la definición de la arquitectura de software para la Informatización del proceso de Gestión de la Calidad en el CECMED. Una definición de arquitectura, abarca la estructura del programa en términos de los estilos que guiarán el desarrollo de la aplicación, las tecnologías y herramientas en las que se apoyará el desarrollo.

Los estilos arquitectónicos guían a la organización del sistema de software. Estos incluyen reglas y líneas a seguir para la organización de un sistema. El **estilo llamada y retorno** es el más utilizado en los grandes sistemas informáticos. Esta familia de estilos enfatiza la escalabilidad. Son los estilos más generalizados en sistemas en gran escala.

Los patrones de arquitectura relacionados a la interacción de objetos dentro o entre los niveles arquitectónicos, se aplican durante la fase de diseño inicial para resolver problemas arquitectónicos,

adaptabilidad a requerimientos cambiantes, performance, modularidad y acoplamiento. El patrón **Modelo Vista Controlador (MVC)**, perteneciente al estilo anterior, es muy recomendado para sistemas interactivos.

Es un patrón fácil de explicar y entender que reduce dependencias y potencia la reutilización. También se aprovecha la ventaja que brinda al permitir dividir el trabajo en base a distintos roles (el diseñador gráfico no necesita saber cómo se ha implementado el resto de los componentes). (20)

Cuando se lleva a cabo un proceso de desarrollo se define **quién** está haciendo **qué, cuándo, y cómo** alcanzar un determinado objetivo. Se define también como el conjunto de actividades necesarias para convertir los requisitos de los usuarios en un producto final. Durante el período de desarrollo de un proyecto de software es necesario un proceso que sirva como guía a todos los interesados en el proyecto, clientes, usuarios y desarrolladores. Un Proceso de Desarrollo de Software es la definición del conjunto de actividades que guían los esfuerzos de las personas implicadas en el proyecto, a modo de plantilla que explica los pasos necesarios para terminar el proyecto. (21)

Las piedras angulares del proceso de desarrollo del software son: el proyecto, las personas y el producto; siendo las características del cliente, el entorno de desarrollo y las condiciones del negocio, elementos que influyen en el proceso. Existe una estrecha relación entre personas, proyecto, producto y proceso. Estos términos son conocidos como las cuatro “P” en el desarrollo de software.

1.5.1 Metodologías de software

La metodología de software se desarrolla con el objetivo de dar solución a los problemas existentes en la producción de software, que cada vez son más complejos. Estas engloban procedimientos, técnicas, documentación y herramientas que se utilizan en la creación de un producto de software. Una metodología es un proceso.

Se debe mencionar que no existe una metodología de software universal. Las características de cada proyecto (equipo de desarrollo, recursos, etc.) exigen que el proceso sea configurable y las diversas metodologías existentes son aplicables a diferentes proyectos según sus características. Para dar una idea de que metodología se puede usar y cuál será la más adaptable a nuestro sistema, se hará una breve descripción de las siguientes: RUP, XP y MSF.

Proceso Unificado de Desarrollo (RUP)

Una de las metodologías de desarrollo más usadas actualmente es RUP. Su desarrollo como producto sigue un camino que comienza con la publicación en 1987 de su primera versión, Proceso Objeto¹, evolucionando luego al Proceso Objeto de Rational en 1997 hasta el Proceso Unificado de Rational en 1998. Su desarrollo ha recibido influencias de muchas fuentes, algunas desconocidas hasta para sus principales creadores (Jacobson, Boch, y Rumbaugh). Como RUP es un proceso, en su modelación define como sus principales elementos (21):

Trabajadores (“quién”): Define el comportamiento y responsabilidades (rol) de un individuo, grupo de individuos, sistema automatizado o máquina.

Actividades (“cómo”): Es una tarea que tiene un propósito claro, es realizada por un trabajador y manipula elementos.

Artefactos (“qué”): Productos tangibles del proyecto que son producidos, modificados y usados por las actividades.

Flujo de actividades (“cuándo”): Secuencia de actividades realizadas por trabajadores y que produce un resultado de valor observable.

En RUP se han agrupado las actividades en grupos lógicos definiéndose 9 flujos de trabajo principales. Los 6 primeros son conocidos como flujos de trabajo de ingeniería y los tres últimos como flujos de trabajo de apoyo.

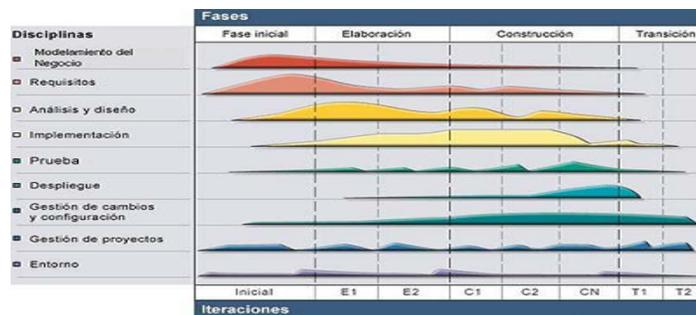


Fig. 2 Figura de RUP en dos dimensiones. Se representa el proceso en el que se grafican los flujos de trabajo y las fases y muestra la dinámica expresada en iteraciones y puntos de control.

El ciclo de vida de RUP se caracteriza por ser:

¹ “Objectory” es una abreviatura de “Object Factory”, compañía fundada por Ivar Jacobson en 1987.

Dirigido por casos de uso: Los casos de uso reflejan lo que los usuarios futuros necesitan y desean, lo cual se capta cuando se modela el negocio y se representa a través de los requerimientos.

Centrado en la arquitectura: La arquitectura muestra la visión común del sistema completo en la que el equipo de proyecto y los usuarios deben estar de acuerdo, por lo que describe los elementos del modelo que son más importantes para su construcción.

Iterativo e Incremental: RUP propone que cada fase se desarrolle en iteraciones. Una iteración involucra actividades de todos los flujos de trabajo, aunque desarrolla fundamentalmente algunos más que otros. Por ejemplo, una iteración de elaboración centra su atención en el análisis y diseño, aunque refina los requerimientos y obtiene un producto con un determinado nivel, pero que irá creciendo incrementalmente en cada iteración.

RUP como metodología de desarrollo se integra con el Lenguaje Unificado de Modelado (UML) permitiendo realizar el análisis, documentación e implementación de sistemas orientados a objetos.

Lenguaje Unificado de Modelado (Unified Modeling Language, UML)

El Lenguaje Unificado de Modelado (UML) es un lenguaje para visualizar, especificar, construir y documentar los artefactos de un sistema. Es la especificación de OMG (Grupo de gestión de objetos, en inglés Object Management Group) y no solo es la forma mundial de representar la estructura de las aplicaciones, comportamiento y arquitectura sino que también representa procesos de negocio y estructura de datos. (22)

Se debe mencionar que UML no es un proceso. Este lenguaje tiene elementos que conforman su vocabulario. En los diagramas se relacionan estos conjuntos de elementos y son los encargados de visualizar un sistema desde diferentes perspectivas. Además permite el modelado de sistemas con tecnología orientada a objetos, los principales diagramas de UML son: diagramas de estructura estática, diagramas de comportamiento, diagramas de implementación

Ventajas

- Ampliamente utilizado por la industria desde su adopción por OMG.
- Modela estructuras complejas.
- Las estructuras más importantes que soportan tienen su fundamento en las tecnologías orientadas a objetos, tales como objetos, clase, componentes y nodos. (23)

Extreme Programming (XP)

Es una de las metodologías de desarrollo de software más exitosa en la actualidad utilizada para proyectos de corto plazo, pequeño equipo y cuyo plazo de entrega era ayer. La metodología consiste en una programación rápida o extrema, cuya particularidad es tener como parte del equipo, al usuario final, pues es uno de los requisitos para llegar al éxito del proyecto. La metodología se basa en:

- **Pruebas Unitarias:** Se basa en las pruebas realizadas a los principales procesos, de tal manera que adelantándonos en algo hacia el futuro, podamos hacer pruebas de las fallas que pudieran ocurrir. Es como si nos adelantáramos a obtener los posibles errores.
- **Refabricación:** Se basa en la reutilización de código, para lo cual se crean patrones o modelos estándares, siendo más flexible al cambio.
- **Programación en pares:** Una particularidad de esta metodología es que propone la programación en pares, la cual consiste en que dos desarrolladores participen en un proyecto en una misma estación de trabajo. Cada miembro lleva a cabo la acción que el otro no está haciendo en ese momento. Es como el chofer y el copiloto: mientras uno conduce, el otro consulta el mapa.

¿Qué es lo que propone XP?

- Empieza en pequeño y añade funcionalidad con retroalimentación continua.
- El manejo del cambio se convierte en una parte sustantiva del proceso.
- El costo del cambio no depende de la fase o etapa.
- No introduce funcionalidades antes que sean necesarias
- El cliente o el usuario se convierten en miembros del equipo. (24)

Microsoft Solution Framework (MSF)

Esta es una metodología flexible e interrelacionada con una serie de conceptos, modelos y prácticas de uso, que controlan la planificación, el desarrollo y la gestión de proyectos tecnológicos. MSF se centra en los modelos de proceso y de equipo dejando en un segundo plano las elecciones tecnológicas.

MSF tiene las siguientes características:

- **Adaptable:** Es parecido a un compás, usado en cualquier parte como un mapa, del cual su uso es limitado a un específico lugar.
- **Escalable:** Puede organizar equipos tan pequeños entre 3 o 4 personas, así como también, proyectos que requieren 50 personas a más.
- **Flexible:** Es utilizada en el ambiente de desarrollo de cualquier cliente.
- **Tecnología Agnóstica:** Porque puede ser usada para desarrollar soluciones basadas sobre cualquier tecnología.(24)

Está basado en desarrollo con tecnología Microsoft lo que limita las opciones del cliente en lo que se refiere a herramientas de desarrollo. Esta metodología hace un análisis de riesgo demasiado exhaustivo que puede frenar el avance del proyecto y además solicita demasiada documentación en todas las fases resultando muy engorroso el trabajo y por consiguiente afecta el tiempo de entrega del producto.

Justificación de la metodología seleccionada

Para guiar el desarrollo de esta aplicación se va a utilizar la metodología RUP pues cuenta con una plataforma flexible de procesos de desarrollo de software que ayuda brindando guías consistentes y personalizadas de procesos para todo el equipo de proyecto. RUP describe cómo utilizar de forma efectiva reglas de negocio y procedimientos comerciales probados en el desarrollo de software para equipos de desarrollo de software, conocidos como “mejores prácticas”. Es una guía de cómo utilizar de manera efectiva UML, provee a cada miembro del equipo fácil acceso a una base de conocimiento con guías, plantillas y herramientas para todas las actividades críticas de desarrollo.

Permite trabajar con precisión y calidad, perfeccionado el software en cualquier momento del desarrollo, logrando así, un sistema con la robustez necesaria, independientemente del tiempo disponible. Si se maneja bien el conjunto de procesos, es posible realizar todos los artefactos que propone RUP independientemente del tamaño del equipo de desarrollo. Además de integrarse con el Lenguaje Unificado de Modelado (UML) permitiendo realizar el análisis, la documentación e implementación de sistemas orientados a objetos, siendo la metodología recomendada en proyectos grandes y ser la utilizada por la empresa con la cual se va a desarrollar la aplicación.

1.5.2 Herramientas CASE

Las herramientas CASE (Computer Aided Software Engineering, Ingeniería de Software Asistida por Ordenador) son diversas aplicaciones informáticas destinadas a aumentar la productividad en el desarrollo

de software, mejorar la calidad de los sistemas realizados y reducir el coste de las mismas en términos de tiempo y dinero. La tecnología CASE permite partir los proyectos en múltiples piezas que se pueden trabajar individualmente y que luego pueden ser integradas en una única aplicación que funciona perfectamente.

Beneficios de las Herramientas CASE

Entre los beneficios más significativos de las herramientas CASE se enumeran los siguientes:

- **Facilidad para la revisión de aplicaciones:** La experiencia muestra que una vez que las aplicaciones se implementan, se emplean por mucho tiempo. Las herramientas CASE proporcionan un beneficio substancial para las organizaciones al facilitar la revisión de las aplicaciones.
- **Generación de código:** La ventaja más visible de esta característica es la disminución del tiempo necesario para preparar un programa. Sin embargo, la generación del código también asegura una estructura estándar y consistente para el programa (lo que tiene gran influencia en el mantenimiento) y disminuye la ocurrencia de varios tipos de errores, mejorando de esta manera la calidad.
- **Mejora en la habilidad para satisfacer los requerimientos del usuario:** Es bien conocida la importancia de satisfacer los requerimientos del usuario, ya que esto guarda relación con el éxito del sistema. De manera similar, tener los requerimientos correctos mejora la calidad de las prácticas de desarrollo. Las herramientas CASE disminuyen el tiempo de desarrollo, una característica que es importante para los usuarios.
- **Soporte interactivo para el proceso de desarrollo:** Las herramientas CASE soportan pasos interactivos al eliminar el tedio manual de dibujar diagramas, elaborar catálogos y clasificar.

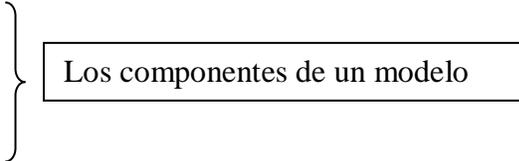
Magic Draw

Magic Draw es una herramienta de modelaje con completas características UML. Es desarrollada por No Magic, Inc. Implementada totalmente en JAVA. Diseñada para los analistas del negocio, los analistas del software, los programadores, los ingenieros de software, y los escritores de la documentación, esta herramienta de desarrollo dinámica y versátil facilita análisis y el diseño de los sistemas y de las bases de datos orientados objeto. Sus características principales son: ayudas en el diseño con auto completamiento y corrección automática en tiempo real, permite visualizar el proyecto de diferentes formas, posible derivación de modelos UML a través de códigos fuente escritos anteriormente, facilidad y rapidez para el cambio del dominio del modelado, generador automático de informes, disponible para un gran número de plataformas y sistemas operativos. (19)

Rational Rose

Rational Rose es una herramienta de producción y comercialización establecidas por Rational Software Corporation (actualmente parte de IBM). Rose es un instrumento operativo que utiliza el Lenguaje Unificado (UML) como medio para facilitar la captura de dominio de la semántica, la arquitectura y el diseño.

Este software tiene la capacidad de:

- Crear
 - Ver
 - Modificar
 - Manipular
- 
- Los componentes de un modelo

Sus características principales: no es gratuito, se debe hacer un previo pago para poder adquirir el producto, la ingeniería de código (directa e inversa) es posible para ANSI C++, Visual C++, Visual Basic 6, Java, J2EE/EJB, CORBA, Ada 83, Ada 95, Bases de datos: DB2, Oracle, SQL 92, SQL Server, Sybase, Aplicaciones WEB, Rational Rose habilita asistentes para crear clases y provee plantillas de código que pueden aumentar significativamente la cantidad de código fuente generado. Adicionalmente, se pueden aplicar los patrones de diseño, Rational Rose ha provisto 20 de los patrones de diseño GOF para Java, además admite la integración con otras herramientas de desarrollo (Ides). (19)

Visual Paradigm for UML 8.0

Es una herramienta CASE, desarrollada por la compañía Visual Paradigm International, que utiliza UML como lenguaje de modelado. Soporta el ciclo de vida completo del desarrollo de software: análisis y diseño orientados a objetos, construcción, pruebas y despliegue. Permite dibujar todos los tipos de diagramas de clases, código inverso, generar código desde diagramas y generar documentación. También proporciona abundantes tutoriales de UML, demostraciones interactivas de UML y proyectos UML. Presenta licencia gratuita y comercial. Es fácil de instalar, actualizar y es compatible entre ediciones. Presenta las siguientes características: soporte de UML versión 2.1, diagramas de Procesos de Negocio. (Actor de Negocio, Caso de Uso del Negocio), diagramas de Clases de Diseño, Diagrama de Despliegue, ingeniería inversa - Código a modelo, código a diagrama, generación de bases de datos - Transformación de diagramas de Entidad-Relación en tablas de base de datos, Se integra con herramientas como: Eclipse, Net Beans, IntelliJ IDEA. (19)

Justificación de la Herramienta CASE seleccionada

Para el análisis, el diseño y el desarrollo del software se decidió utilizar la Herramienta CASE Visual Paradigm en su versión 8.0, permitiendo modelar todos los diagramas necesarios para representar gráficamente el ciclo de vida del desarrollo de software: análisis y diseño orientados a objetos. Esta herramienta propicia un conjunto de ayudas para el desarrollo de programas informáticos, desde la planificación, pasando por el análisis y el diseño, hasta la generación del código fuente de los programas y la documentación. Esta herramienta CASE posee numerosas ventajas que determinaron su elección para el desarrollo de la aplicación. A continuación se listan las principales ventajas con las que cuenta la herramienta: disponibilidad en múltiples plataformas (Windows, Linux), uso de un lenguaje estándar común a todo el equipo de desarrollo que facilita la comunicación, licencia: gratuita y comercial, generación de código para Java, permite la integración con distintos Entornos de Desarrollo Integrados, entre ellos IntelliJ IDEA.

1.5.3 Marco de Trabajo o Framework

Un framework es una estructura de soporte definida en la cual otro proyecto de software puede ser organizado y desarrollado. Proporciona una estructura que fuerza al desarrollo de código más legible. Representa una arquitectura de software que modela las relaciones generales de las entidades del dominio y permite separar en capas la aplicación.

Symfony Framework

Symfony es un completo framework diseñado para optimizar, gracias a sus características, el desarrollo de las aplicaciones web. Para empezar, separa la lógica de negocio, la lógica de servidor y la presentación de la aplicación web. Proporciona varias herramientas y clases encaminadas a reducir el tiempo de desarrollo de una aplicación web compleja. Además, automatiza las tareas más comunes, permitiendo al desarrollador dedicarse por completo a los aspectos específicos de cada aplicación. Symfony se diseñó para que se ajustara a los siguientes requisitos: fácil de instalar y configurar en la mayoría de plataformas (y con la garantía de que funciona correctamente en el sistema operativo Windows), independiente del sistema gestor de bases de datos, sencillo de usar en la mayoría de casos, pero lo suficientemente flexible como para adaptarse a los casos más complejos, basado en la premisa de "convenir en vez de configurar", en la que el desarrollador solo debe configurar aquello que no es convencional, sigue la mayoría de mejores prácticas y patrones de diseño para la web, código fácil de leer y documentar, además de que permite un mantenimiento muy sencillo. (25)

Ext JS Framework

Ext JS es una librería de Java Script ligera, de alto rendimiento para el desarrollo rápido de aplicaciones web *cross-browser*. Presenta una interfaz de usuario personalizable, bien diseñada y contiene un modelo de componentes extensibles. Tiene disponible licencias comerciales y de código abierto. Es compatible con la mayoría de los navegadores y su familia de productos es utilizada por miles de compañías en el mundo. (26)

Grails 2.3.7

Es un framework para aplicaciones web libres, desarrollado sobre el lenguaje de programación Groovy (el cual a su vez se basa en la plataforma Java). Grails pretende ser un marco de trabajo altamente productivo siguiendo paradigmas tales como convención sobre configuración o no te repitas (DRY), proporcionando un entorno de desarrollo estandarizado y ocultando gran parte de los detalles de configuración al programador. Grails fue conocido como 'GroovyonRails' (el nombre cambió en respuesta a la petición de David Heinemeier Hansson, fundador de Ruby onRails).

Grails se ha desarrollado con una serie de objetivos en mente: ofrecer un framework web de alta productividad para la plataforma Java, reutilizar tecnologías Java ya probadas como Hibernate y Spring bajo una interfaz simple y consistente, ofrecer un framework consistente que reduzca la confusión y que sea fácil de aprender, ofrecer documentación para las partes del framework relevantes para sus usuarios. Proporciona lo que los usuarios necesitan en áreas que a menudo son complejas e inconsistentes: framework de persistencia potente y consistente, patrones de visualización potente y fácil de usar con GSP (Groovy Server Pages), bibliotecas de etiquetas dinámicas para crear fácilmente componentes web, buen soporte de Ajax que sea fácil de extender y personalizar.

Grails se ha diseñado para ser fácil de aprender, fácil para desarrollar aplicaciones y extensible. Intenta ofrecer el equilibrio adecuado entre consistencia y funcionalidades potentes. Tiene tres características que intentan incrementar su productividad comparándolo con los framework Java tradicionales:

- Inexistencia de configuración XML.
- Entorno de desarrollo preparado para funcionar desde el primer momento.
- Funcionalidad disponible mediante métodos dinámicos.

Posee un entorno de desarrollo preparado para funcionar desde el primer momento mientras usamos herramientas Java tradicionales, es tarea del desarrollador ensamblar los componentes, lo cual puede ser tedioso. Grails tiene un servidor web integrado preparado para desplegar la aplicación desde el primer momento. Todas las librerías requeridas son parte de la distribución de Grails y están preparadas para ser desplegadas automáticamente.

El framework web de Grails se ha diseñado según el paradigma Modelo Vista Controlador. Grails utiliza la JVM, está construido sobre cinco pilares fundamentales:

- Groovy (lenguaje de programación) para la creación de propiedades y métodos dinámicos en los objetos de la aplicación.
- Spring (framework para el desarrollo de aplicaciones) para los flujos de trabajo e inyecciones de dependencias.
- Hibernate (framework de persistencia) para el Mapeo objeto-relacional.
- SiteMesh (framework de aplicación) que gestiona la creación de las vistas.
- Ant como framework para gestionar la compilación de un proyecto.(27)

Algunas de las características fundamentales de Grails son:

- **Productividad:** posee características fundamentales que intentan incrementar su productividad en comparación con los marcos de trabajo Java tradicionales, es un marco de trabajo preparado para funcionar desde el primer momento, y con funcionalidades disponibles a través de métodos dinámicos.
- **Persistencia:** GORM (Grails Object Relational Mapping) es el medio a través el cual los datos del modelo de dominio se hacen persistentes utilizando métodos dinámicos.

Justificación del Framework seleccionado

El framework seleccionado para el desarrollo del software fue Grails 2.3.7 pues permite crear un marco de desarrollo que favorece la productividad al crear aplicaciones web, integrando factores de configuración comunes a la mayoría de los escenarios siguiendo paradigmas tales como Convención sobre Configuración cuyo objetivo fundamental es disminuir el número de decisiones que el desarrollador debe tomar, ganando en sencillez pero no pierde por ello la flexibilidad y No te Repitas, ya que su finalidad es promover la disminución de la duplicación. (27)

1.5.4 Lenguajes de programación utilizados

Un lenguaje de programación es aquel elemento dentro de la informática que nos permite crear programas mediante un conjunto de instrucciones, operadores y reglas de sintaxis; que pone a disposición del programador para que este pueda comunicarse con los dispositivos hardware y software existentes.

Groovy

Es un lenguaje de programación orientado a objetos implementado sobre la plataforma Java que compila a código de bytes de Java y se ejecuta en la Máquina Virtual de Java (JVM), permitiendo su integración a la perfección con Java, haciendo el aprendizaje de los programadores más fácil. Como base para su confección se usaron lenguajes tales como Java, Python y otros. Sin embargo, su sintaxis es mucho más flexible y potente que Java pues posee una gran flexibilidad, dinamismo e integración.

Uno de los puntos más fuertes sobre Groovy y Grails es que son nativos de la JVM. Por esta razón, Groovy y Grails están destinados a ser un gran éxito en el mundo empresarial. Dada la universalidad de Java hoy en día, sería demasiado pedir a los desarrolladores implementar todas sus infraestructuras basadas en API (Interfaz de Programación de Aplicaciones), bibliotecas y marcos empezando desde el principio. (27)

HTML 5

El HTML 5 (Hyper Text Markup Language, versión 5) es la quinta revisión del lenguaje de programación “básico” de la World Wide Web. Esta nueva versión pretende reemplazar al actual XHTML, corrigiendo problemas con los que los desarrolladores web se encuentran, así como rediseñar el código actualizando las nuevas necesidades que demanda la web de hoy en día. (28)

Lleva a un nivel más alto que el código HTML 4 el lenguaje de marcado, ya que será capaz de controlar los eventos y las iteraciones con el usuario. La diferencia esencial no está en el lenguaje en sí mismo, se añaden las etiquetas nuevas en comparación con HTML 4 y no requiere un tipo de documento específico, el verdadero “cambio” es en el manejo de la tela, o las áreas de la página que se puede utilizar como pizarras interactivas (se puede diseñar y gestionar los eventos), todos a través de Java Script.

HTML 5 conduce a una fusión entre Java Script como lenguaje de programación, HTML como modelo semántico y CSS 3 que es la evolución del CSS como el lenguaje de los estilos, que se dedica a dar un mejor aspecto a nuestros proyectos. (29)

A continuación alguna de las reglas establecidas para HTML 5:

- Las nuevas características debe basarse en HTML, CSS, DOM y Java Script.
- Reducir la necesidad de plugins externos (como Flash).
- Mejor manejo de errores.
- Más marcado para reemplazar secuencias de comandos.
- HTML 5 debe ser independiente del dispositivo.
- El proceso de desarrollo debe ser visible para el público.

CSS 3

CSS 3 es la última versión hasta la fecha y presenta como principales características mayor control sobre el estilo de los elementos de nuestra página web y mayor número de efectos visuales. Mejora la accesibilidad del documento, reduce la complejidad de su mantenimiento y permite visualizar el mismo documento en infinidad de dispositivos diferentes. Si el lenguaje HTML/XHTML se utiliza para marcarlos contenidos, es decir, para designar lo que es un párrafo, lo que es un titular o lo que es una lista de elementos, el lenguaje CSS se utiliza para definir el aspecto de todos los contenidos, es decir, el color, tamaño y tipo de letra de los párrafos de texto, la separación entre titulares y párrafos, la tabulación con la que se muestran los elementos de una lista entre otros. (30)

Las ventajas de usar CSS 3 son:

- Código más simple para muchas tareas.
- Mayores opciones gráficas. (31)

1.6 Ambiente de Desarrollo Integrado (IDE)

IDE NetBeans

NetBeans es un entorno de desarrollo integrado, extensible para el desarrollo sobre varios lenguajes, aunque se realizó fundamentalmente para Java. Es un producto de código abierto desarrollado por la compañía Sun MicroSystem en el año 2000 que seguía siendo el patrocinador del proyecto hasta enero de 2010, cuando Sun Microsystems se convirtió en una subsidiaria de Oracle.

Se basa en una filosofía modular, lo que permite el desarrollo de múltiples proyectos con el uso de varias tecnologías. Entre sus principales ventajas se encuentran:

- Administración de interfaces de usuario.
- Integración a múltiples framework.

- Administración de almacenamiento.
- Gran cantidad de módulos y extensiones para múltiples lenguajes (PHP, Java, C / C ++, Java Script, etc.) y tecnologías.
- Fuerte comunidad de respaldo.(32)

Intelij IDEA

Es un Ambiente de Desarrollo Integrado o Entorno de Desarrollo Interactivo, que proporciona servicios integrales para facilitarle al desarrollador o programador el desarrollo de software. Uno de los propósitos de los IDE es reducir la configuración necesaria para reconstruir múltiples utilidades de desarrollo, en vez de proveer el mismo set de servicios como una entidad cohesiva. Existen muchos IDE de múltiples lenguajes tales como Eclipse, Active State Komodo, IntelliJ IDEA, Oracle JDeveloper, NetBeans, Codenvy y Microsoft Visual Studio. El IntelliJ IDEA está disponible en dos ediciones: Community Edition (Gratis) y Ultimate Edition (De pago). Ambas ediciones soportan varios lenguajes de programación entre ellos: Java, Groovy, Scala (vía plugin), XML/XSL, entre otros. Ambas ediciones pueden usar las tecnologías y framework siguientes: Ajax, Android, Django, Grails entre otras. La edición ultimate también apoya los servidores de aplicación Gerónimo, GlassFish, JBoss, Jetty, Tomcat, Weblogic, y WebSphere. (33)

Justificación del IDE seleccionado

Se selecciona IntelliJ IDEA pues presenta un único programa en el que se lleva a cabo todo el desarrollo ofreciendo varias características para la creación, modificación, compilación, implementación y depuración de software.

1.7 Sistema Gestor de Base de Datos (SGBD)

Un Sistema de Gestión de Bases de Datos es aquel que se encarga de controlar el acceso concurrente, evitar redundancia, que se cumplan las restricciones y reglas de integridad, usar elementos que aceleren el acceso físico a los datos (índices, agrupamientos, funciones de dispersión), distribuir los bloques del disco del modo más adecuado para el crecimiento y uso de los datos, controlar el acceso y los privilegios de los usuarios y recuperar ante fallos. (34)

Oracle

Oracle es un sistema de administración de base de datos, fabricado por Oracle Corporation, básicamente una herramienta cliente/servidor para la gestión de Bases de Datos. Es un producto vendido a nivel mundial, aunque la gran potencia que tiene y su elevado precio hacen que solo se vea en empresas muy grandes y multinacionales, por norma general. En el desarrollo de páginas Web pasa lo mismo: como es un sistema muy caro no está tan extendido como otras bases de datos, por ejemplo, Access, MySQL, SQL Server y otros. Oracle es sin duda una de las mejores bases de datos que tenemos en el mercado, es un sistema gestor de base de datos robusto, tiene muchas características que nos garantizan la seguridad e integridad de los datos; que las transacciones se ejecuten de forma correcta, sin causar inconsistencias; ayuda a administrar y almacenar grandes volúmenes de datos; estabilidad, escalabilidad y es multiplataforma. Aunque su dominio en el mercado de servidores empresariales ha sido casi total hasta hace poco, recientemente sufre la competencia de gestores de bases de datos comerciales y de la oferta de otros con licencia Software Libre como PostgreSQL o FireBird. (35)

MySQL

MySQL es un sistema de administración de bases de datos relacional (RDBMS). Se trata de un programa capaz de almacenar una enorme cantidad de datos de gran variedad y de distribuirlos para cubrir las necesidades de cualquier tipo de organización, desde pequeños establecimientos comerciales a grandes empresas y organismos administrativos. Puede desarrollar sus propias aplicaciones de bases de datos en la mayor parte de lenguajes de programación utilizados en la actualidad y ejecutarlos en casi todos los sistemas operativos, incluyendo algunos de los que probablemente no ha oído hablar nunca, utiliza el lenguaje de consulta estructurado (SQL), es rápido es una solución accesible para administrar correctamente los datos de una empresa. MySQLAB es la compañía responsable del desarrollo de MySQL, dispone de un sistema de asistencia eficiente y a un precio razonable, y como ocurre con la mayor parte de las comunidades de código abierto, se puede encontrar una gran cantidad de ayuda en la Web. Como características fundamentales se pueden mencionar:

- Asistencia: MySQLAB ofrece contratos de asistencia a precios razonables y existe una nutrida y activa comunidad MySQL.
- Funcionalidad: MySQL dispone de muchas de las funciones que exigen los desarrolladores profesionales, como compatibilidad completa con ACID, volcados online, duplicación, funciones SSL e integración con la mayor parte de los entornos de programación.
- Portabilidad: MySQL se ejecuta en la inmensa mayoría de sistemas operativos y la mayor parte de los casos, los datos se pueden transferir de un sistema a otro sin dificultad.

- Facilidad de uso: MySQL resulta fácil de utilizar y de administrar. Las herramientas de MySQL son potentes y flexibles, sin sacrificar su capacidad de uso.(35)

PostgreSQL 9.2.4

PostgreSQL es un Sistema de Gestión de Bases de Datos Objeto-Relacionales (ORDBMS) que ha sido desarrollado de varias formas desde 1977. Comenzó como un proyecto denominado Ingres en la Universidad Berkeley de California. Está considerado como la base de datos de código abierto más avanzada del mundo, proporciona un gran número de características que normalmente solo se encontraban en las bases de datos comerciales tales como DB2 u Oracle. A continuación se muestra brevemente una lista de algunas de esas características:

- Soporte SQL Comprensivo: PostgreSQL soporta la especificación SQL99 e incluye características avanzadas tales como las uniones (joins) SQL92.
- Integridad Referencial: PostgreSQL soporta integridad referencial, la cual es utilizada para garantizar la validez de los datos de la base de datos.
- DBMS Objeto-Relacional: PostgreSQL aproxima los datos a un modelo objeto-relacional y es capaz de manejar complejas rutinas y reglas. Ejemplos de su avanzada funcionalidad son consultas SQL declarativas, control de concurrencia multi-versión, soporte multi-usuario, optimización de consultas, herencia y arrays.
- Altamente Extensible: PostgreSQL soporta operadores, funciones, métodos de acceso y tipos de datos definidos por el usuario.
- Cliente/Servidor: PostgreSQL usa una arquitectura proceso-por-usuario cliente/servidor.(36)

Justificación del SGB seleccionado

Se escoge el PostgreSQL en su versión 9.2 para la solución propuesta por ser uno de los SGBD de código abierto más potentes del mercado y ser multiplataforma. Por ser rápido, confiable y fácil de usar, ofrece hoy una rica variedad de funciones, su conectividad, velocidad y seguridad lo hacen altamente satisfactorio para acceder bases de datos además de ser la base de datos de código abierto más avanzada del mundo. Además por las características del módulo a desarrollar y la importancia de un SGBD robusto para una correcta manipulación de los datos, incluso en situaciones críticas además la seguridad que deben tener los mismos. (37)

Servidor de Aplicaciones Web

Tomcat v7

Tomcat es una implementación completamente funcional de los estándares de JSP (Java Server Pages) y Java Servlet, es el servidor Web más utilizado a la hora de trabajar con el lenguaje Java en aplicaciones web, puede especificarse también como el manejador de las peticiones de JSP recibidas por servidores Web populares, como el servidor Apache HTTP de la Fundación de software de Apache o el servidor Microsoft Internet Information Server (IIS).

Tomcat puede utilizarse como un contenedor solitario (principalmente para desarrollo y depuración) o como plugin para un servidor web existente (en este caso está integrado como un plugin al framework Grails). (38) Dado que Tomcat fue escrito en Java, funciona en cualquier sistema operativo que disponga de la máquina virtual Java. Se seleccionó como Servidor Web pues el servidor Web más utilizado a la hora de trabajar con el lenguaje Java en aplicaciones web.

1.8 Conclusiones parciales

En este capítulo se detallaron los conceptos y definiciones que rodean al objeto de estudio. Se formalizó un estudio del estado del arte sobre la planificación de auditorías, la atención a las incidencias y las soluciones informáticas que actualmente se enfocan en esa área. El estudio arrojó como resultado que los sistemas estudiados a pesar de satisfacer las necesidades del DGC del CECMED en cuanto a la gestión de información, no pueden ser adquiridos por su valor adquisitivo y no integrarse al sistema desarrollado por SOFTEL el cual se encuentra desplegado en esta institución. Por tanto se decidió implementar una aplicación web siguiendo las características principales de los sistemas estudiados que apliquen a la institución a automatizar para la gestión de información del DGC del CECMED.

Por último se fundamentó la selección de las metodologías, tecnologías y herramientas necesarias para desarrollar la aplicación. Definiéndose para la realización de la aplicación propuesta como metodología de desarrollo de software a RUP y como lenguaje unificado de modelado UML, tomando en consideración que la integración de ambas permite el análisis, documentación e implementación de sistemas orientado a objetos.

Capítulo 2

Descripción del Sistema

Introducción

En el actual capítulo se muestran los distintos procesos que tienen lugar en el DGC del CECMED, enmarcándose en la recepción y tratamiento de las incidencias. Son realizados los artefactos correspondientes al flujo de trabajo Modelación del Negocio donde son enunciadas las reglas del negocio, confeccionado el diagrama de casos de uso del negocio con los diagramas de actividades y el modelo de objetos. Se identifican los requisitos funcionales y no funcionales propiciando las condiciones y capacidades del sistema para satisfacer las necesidades del cliente. Se establecerán los vínculos entre los actores y casos de uso del sistema, así como la descripción detallada de cada uno de ellos.

2.1. Objeto de estudio

Objetivos estratégicos de la organización

Los especialistas del DGC son los encargados de garantizar el cumplimiento de los requisitos regulatorios establecidos en la ISO 10002:2004 Gestión de la calidad (39), para ello realizan las auditorías de calidad a cada departamento o dirección de la institución, prestándole especial atención a las no conformidades detectadas y a las incidencias presentadas a la institución.

Los procesos que serán objeto de informatización se encuentran:

Tratamiento de Incidencias: Son entregadas por los clientes al especialista del DGC, quien registra todos los datos del comunicado en el “Registro de Quejas, “Registro de Reclamaciones” y “Registro de Reclamaciones” y le envía un acuse de recibo al cliente con el nombre, el cargo y la fecha del especialista que se encargará de la investigación.

Al informatizar estos procesos se logrará mantener informado a todo el personal implicado y conocer el estado de la calidad de los servicios brindados por el centro. Además de tener acceso por parte de los especialistas y el personal autorizado a la información de los planes de auditoría y las incidencias, logrando tener un control de las actividades relacionadas con el funcionamiento del DGC.

Propuesta de sistema

Después de haber realizado un estudio y análisis de los problemas existentes en el DGC del CECMED; en los niveles correspondientes con el manejo y la gestión de toda la información que se necesita controlar, buscar y obtener; y teniendo en cuenta de que no existe un sistema anterior que dé cumplimiento total a lo requerido y que cubra en su totalidad todas las necesidades, se propone desarrollar dos módulos que aporten soluciones reales, palpables y satisfactorias a las necesidades planteadas. Estos se integrarán al Sistema de Información del CECMED y su principal administrador será el Jefe del DGC máxima autoridad del departamento.

El módulo ofrecerá la posibilidad de gestionar las incidencias reportadas. El sistema quedará elaborado de forma tal que muestre al usuario solo aquellas opciones a las que está autorizado a acceder. Debe ser fácil de usar y debe mantener todos los datos archivados con la mayor seguridad disponible, así como la integridad de los mismos.

2.2. Modelo de negocio

El proceso de modelamiento del negocio permite obtener una visión de la organización con el objetivo de definir los procesos, roles y responsabilidades de la organización en los modelos de casos de uso del negocio y de objetos.

2.2.1. Actores del negocio

Un actor del negocio es cualquier individuo, grupo, entidad, organización, máquina o sistema de información externos con los que el negocio interactúa y desempeña un rol determinado dentro del negocio para beneficiarse de sus resultados. Representa un tipo particular de usuario del negocio más que un usuario físico, ya que varios usuarios físicos pueden realizar el mismo papel en relación al negocio, por otro lado un mismo usuario puede actuar como diferentes actores.(40)

Tabla 1 Descripción del actor del negocio.

Actores del Negocio	Justificación
Cliente	Es la persona que presenta una Queja, Reclamación y Reconsideración al Departamento de Gestión de la Calidad del CECMED, por la insatisfacción de un servicio recibido con el objetivo de recibir una respuesta a su solicitud.
CECMED	Como organización al realizarse las auditorias por el DGC obtiene como beneficio la posibilidad de detectar no conformidades en los procesos llevados a cabo por cada departamento y así poder realizar acciones correctivas que le permitan mejorar la calidad de los servicios que la institución ofrece.

2.2.2 Trabajadores del negocio

Un trabajador del negocio representa a personas o sistemas dentro del negocio que son los que realizan las actividades que están comprendidas dentro de un caso de uso. Estos trabajadores están dentro de la frontera del negocio, son los candidatos a convertirse en un futuro en usuarios del sistema que se quiere construir.

Tabla 2 Descripción de los trabajadores del negocio.

Trabajadores del Negocio	Justificación
Especialista DGC	Es la persona encargada de registrar las Incidencia presentadas por los clientes al DGC, además de darle seguimiento a las mismas.
Evaluador	Encargado de evaluar las incidencias reportadas en el DGC.

2.2.3 Diagrama de casos de uso del negocio

El diagrama de casos de uso representa la interacción entre actores del negocio y los casos de uso del negocio, incluye las relaciones entre actores y entre casos de uso.

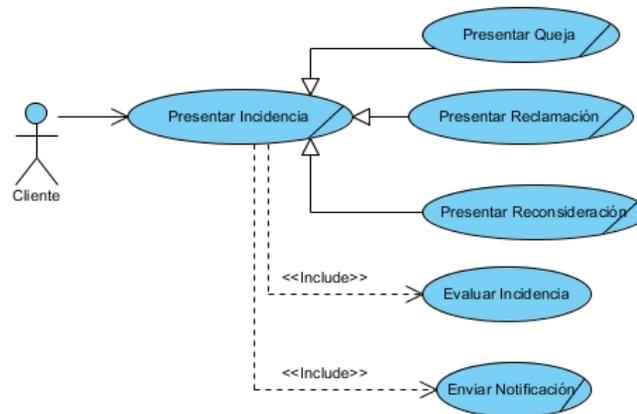


Fig. 3 Diagrama de casos de uso del negocio.

2.2.4 Realización de los casos de uso del negocio

Los casos de uso del negocio consisten en secuencias de actividades que, en conjunto, producen algo para el actor del negocio. La realización de los casos de uso ayuda a entender en detalle cómo se suceden las actividades a partir de la descripción de los flujos de procesos. Brinda una noción de cómo funcionan estos procesos dentro del negocio.

Descripción en formato extendido del CUN Presentar Incidencia.

Tabla 3 Especificación del caso de uso Presentar Incidencia.

Caso de Uso:	Presentar Incidencia
Actores:	Cliente
Trabajadores:	Especialista del DGC
Resumen:	<p>El cliente presenta la comunicación con su Incidencia al Departamento de Gestión de la Calidad del CECMED las cuales se clasifican en: quejas, reclamaciones y reconsideraciones.</p> <p>El cliente puede presentar una incidencia debido a: trato inadecuado, espera prolongada y falta de información o dificultad para acceder a ella siendo estos, motivos para presentar una queja CECMED.</p> <p>Si el cliente presenta una Reclamación, esta debe estar originada por errores en las certificaciones, falta de legibilidad, fallas mecanográficas y errores en algunos de los campos de los documentos emitidos por el CECMED.</p> <p>Si el cliente se encuentra inconforme, con decisiones tomadas por el centro, en relación, a solicitudes realizadas por este, vinculadas a Trámites de Registro, Licencias de Establecimientos, Ensayos Clínicos, Liberación de Lotes entre otros que han sido rechazados después de la evaluación integral, períodos de validez aprobados por menor tiempo que el solicitado, medidas sanitarias aplicadas como resultados de controles o investigaciones entre otros, puede presentar una Reconsideración de la decisión reguladora.</p>
Posibles mejoras:	Se recomienda informatizar el proceso de acceso a la información requerida para agilizar todas las actividades relacionadas.
Flujo Normal de Eventos	
Acción del Actor	Respuesta del Negocio
1- El Cliente se presenta al DGC del CECMED y le entrega al Especialista del DGC el documento con su Incidencia.	2- El Especialista del DGC revisa el documento verificando que todos los datos necesarios estén reflejados revisando los motivos de la Incidencia para registrarla en el registro correspondiente, si los motivos indican que es una queja este lo registra en el "Registro de Quejas" con los siguientes datos: Id de la Queja, Organización, Dirección, Código Postal, Ciudad, País, Número de Teléfono, Número de Fax, Correo electrónico, Datos del representante del reclamante (si aplica), Persona de Contacto (si es diferente del reclamante), Descripción del producto o servicio, Número de referencia del producto o pedido, Motivo de la Queja, Fecha de ocurrencia, Descripción, Fecha en que se emite la Queja, Adjuntos (si procede): Lista de documentos que se adjuntan.

	3- Le envía al cliente por correo electrónico o fax un acuse de recibo en el que constará: Nombre, Cargo y Fecha de quien recibió el comunicado con su Incidencia.
Flujos Alternos	
Flujo alternativo 1: Si la incidencia fue enviada por correo electrónico	
Acción del Actor	Respuesta del Negocio
	2- El Especialista del DGC revisa el correo electrónico y verifica que el cliente le haya adjuntado el documento formal descriptivo de las causas de la Incidencia y la copia digitalizada del documento con errores.
	3- Continúa en el Paso 2 del Flujo Normal de Eventos.
Flujo alternativo 2: Si los motivos de la Incidencia están relacionados con una reconsideración.	
Acción del Actor	Respuesta del Negocio
1- El Cliente se presenta al Área de Reconsideración de Disposiciones Regulatoras del CECMED y le entrega al Especialista DGC el documento con su Incidencia relacionada con la reconsideración.	2- El Especialista del DGC revisa el comunicado verificando que todos los datos necesarios estén reflejados; de ser así el especialista abre un expediente con los datos de la Reconsideración: Id de la reconsideración, Organización, Dirección, Código Postal, Ciudad, País, Número de Teléfono, Número de Fax, Correo electrónico, Datos del representante del reclamante (si aplica), Persona de Contacto (si es diferente del reclamante), Descripción del producto o servicio: Tipo de trámite, Nombre del producto, Nombre del establecimiento, Número de registro, Licencia, Autorización de ensayo clínico y Certificado, Objeto de la Solicitud de Reconsideración: Descripción completa, clara y resumida del(los) aspecto(s) que se discrepa(n), Aspecto técnico-científico, Aspecto del procedimiento o Ambos, Descripción de la solución a la que se aspira, Fundamentación: Argumentos, explicaciones y consideraciones que respaldan la discrepancia y que asisten al solicitante para pedir una reconsideración, Lista de elementos y documentos originales entregados al CECMED (con su localización en el expediente), que opina no fueron tenidos en cuenta o interpretados debidamente o Enumeración de las acciones previas con el CECMED para esclarecer el motivo de la discrepancia (fecha de reuniones, funcionarios contactados,

	asuntos tratados y resultados obtenidos), Fecha y firma del que solicita la Reconsideración, Adjuntos (si procede): Lista de documentos que se adjuntan.
	3. Continúa en el paso 3 del Flujo Normal de eventos.

Diagrama de Actividades

El diagrama de actividades es parte de la realización de un caso de uso, representa un flujo de actividades que describen cómo colaboran trabajadores, actores y entidades del negocio para alcanzar un resultado final. El proceso consiste de un flujo básico de una o más alternativas de flujos. La estructura del flujo se describe gráficamente con la ayuda del diagrama de actividades. A continuación se muestra el diagrama de actividades del CUN Presentar Incidencia.

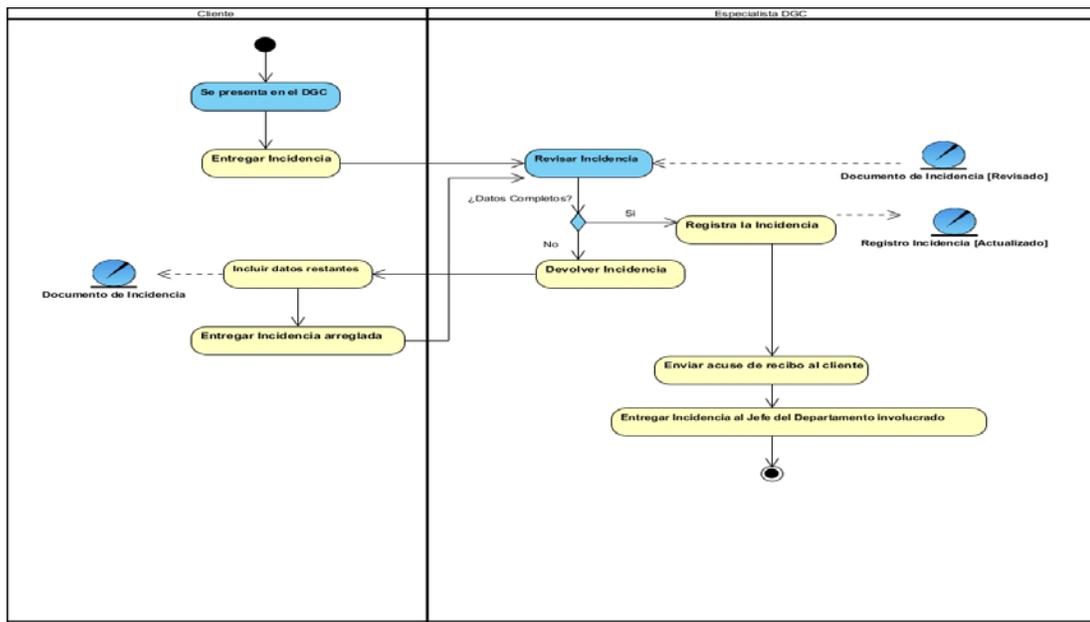


Fig. 4 Diagrama de actividades CUN Presentar Incidencia.

2.2.5 Diagrama de Clases del Modelo de Objetos

El diagrama de clases del modelo de objetos es el conjunto básico de entidades del negocio y los trabajadores del negocio que las manipulan.



Fig. 5 Diagrama de Clases del Modelo de Objetos para el trabajador del negocio Especialista DGC.

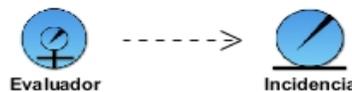


Fig. 6 Diagrama de Clases del Modelo de Objetos para el trabajador del negocio Jefe DGC.

2.2.6 Reglas del negocio

Las reglas de negocio describen políticas que deben cumplirse o condiciones que deben satisfacerse, por lo que regulan algún aspecto del negocio.

Reglas de acción

1. Restricciones de operaciones:

RN 2 Para dar por terminado el proceso de recepción de las Incidencias (quejas, reclamaciones y reconsideraciones) el Especialista DGC debe haber enviado una notificación al Cliente para que este conozca el nombre, el cargo y la fecha en la que se recibió su comunicación. (42)

RN 3 Para llevar a cabo la investigación de una Reclamación y una Reconsideración esta debe venir acompañada de las causas y el documento original emitido por el CECMED con errores. (43)

2. Estímulo y respuesta:

RN 4 Si se cumple el tiempo de evaluación de las Incidencias se procede a comunicarle al Cliente los resultados de la investigación. En caso que no asista a la institución, se procede a enviarle una notificación por correo electrónico para que se presente a recoger los resultados.

2.3 Requerimientos. Modelo del sistema

El flujo de trabajo de RUP Requerimientos, se desarrolla en la fase de Inicio. El objetivo fundamental de este flujo de trabajo es establecer un convenio entre cliente y desarrolladores sobre lo que el sistema debe hacer. A partir de la especificación de los requisitos y la modelación gráfica de estos, agrupados en casos de uso, se brinda una visión de lo que se debe implementar.

El artefacto fundamental en este flujo de trabajo es el Modelo del sistema que contiene actores y casos de uso. Este artefacto es un modelo de las funciones deseadas para el sistema y su entorno, y sirve como contrato entre el cliente y los desarrolladores. Se utiliza como entrada esencial para las actividades de análisis, diseño y prueba. La realización de los casos de uso del sistema se realiza en el flujo de trabajo de Análisis y Diseño.

2.3.1 Especificación de requisitos

Un requisito es:

- a) Condición o capacidad que necesita un usuario para resolver un problema o lograr un objetivo.
- b) Condición o capacidad que tiene que ser alcanzada o poseída por un sistema o componente de un sistema para satisfacer un contrato, estándar, u otro documento impuesto formalmente.
- c) Una representación documentada de una condición o capacidad como en las dos anteriores. (44)

Los requisitos se pueden clasificar en: funcionales y no funcionales. Los requisitos funcionales son capacidades o condiciones que el sistema debe cumplir. Los requisitos no funcionales son propiedades o cualidades que el producto debe tener, las características que hacen al producto atractivo, usable, rápido o confiable.

Técnicas de captura de requisitos

La captura de requisitos es la actividad mediante la cual el equipo de desarrollo de un sistema de software extrae, de cualquier fuente de información disponible, las necesidades que debe cubrir dicho sistema. Para la obtención de los requisitos del sistema se emplearon dos técnicas que permitió hacer este proceso de una forma más eficiente y precisa.

Entrevistas: Resultan una técnica muy aceptada dentro de la ingeniería de requisitos y su uso está ampliamente extendido. A través de las **entrevistas** realizadas al cliente, se logró tomar conocimiento del problema y comprender los objetivos de la solución buscada, lo que ayudó a definir lo que éste desea. Básicamente, la estructura de la entrevista abarca tres pasos: identificación de los entrevistados,

preparación de la entrevista, realización de la entrevista y documentación de los resultados (protocolo de la entrevista). (45)

Tormenta de ideas: Esta técnica de reuniones en grupo cuyo objetivo es que los participantes muestren sus ideas de forma libre, consiste en la acumulación de ideas e información sin evaluar las mismas. El grupo de personas que participa en estas reuniones no debe ser muy numeroso (máximo 10 personas), una de ellas debe asumir el rol de moderador de la sesión, pero sin carácter de controlador. (45) Su aplicación en el DGC perteneciente al CECMED como cliente ofreció una visión general de las necesidades del sistema, aunque no sirvió para obtener detalles concretos del mismo, por lo que se aplicó en los primeros encuentros.

Lista de Requisitos Funcionales

Gestionar Incidencias.

De cada Incidencia se captarán los siguientes datos:

RF 10 Registrar Incidencia.

Las incidencias se clasifican en Queja, Reclamación y Reconsideración y de las mismas se captarán los siguientes datos:

- Clasificación de Incidencia. (Se selecciona a partir de un nomenclador del sistema, obligatorio, ver Anexo 3.)
- No de la Incidencia. (Se genera automáticamente por el sistema y se conforma a partir de un consecutivo total + punto + año y cada año el consecutivo total se inicia en 1, obligatorio)
- Organización. (Se selecciona a partir de un nomenclador del sistema llamado Entidad, obligatorio)
- Dirección. (Se llena automáticamente a partir de la organización seleccionada, obligatorio)
- País. (Se llena automáticamente a partir de la organización seleccionada, obligatorio)
- Número de Teléfono. (Se llena automáticamente a partir de la organización seleccionada, no obligatorio)
- Número de Fax. (entrada de texto, no obligatorio)
- Correo electrónico. (Se llena automáticamente a partir de la organización seleccionada, no obligatorio)
- Datos del representante del reclamante (entrada de texto libre, se expresa el Nombre y Apellidos,

no obligatorio).

- Persona de Contacto (se toma el nombre y apellidos a partir del usuario autenticado en el sistema, solo para el caso en que el rol del usuario sea Cliente, obligatorio).
- Descripción del producto o servicio. (Se llena automáticamente a partir del nomenclador servicio configurado en el sistema, no obligatorio)
- Número de referencia del producto o pedido. (Hace referencia al código del trámite, no obligatorio)
- Motivo de la Incidencia. (Entrada de Texto libre, el campo debe permitir al menos una pantalla para describir el motivo)
- Fecha de ocurrencia. (Campo de selección, donde el formato de la fecha es dd/mm/aa)
- Fecha en que se emite la Incidencia. (Capta la fecha actual del servidor donde se encuentra desplegada la aplicación, la misma corresponde con la fecha de creación de la incidencia)
- Reportado por. (Se toma el nombre y apellidos a partir del usuario autenticado en el sistema, obligatorio).
- Estado. (Valor que se toma a partir de las acciones que se realizan en el sistema puede ser: Nueva, Reportada, Resuelta y Rechazada)
- Adjuntos: Lista de documentos que se adjuntan. (Se solicitan los documentos para las reclamaciones, permite subir múltiples documentos a la vez)

En el caso de las Reconsideraciones se captarán además el siguiente dato:

- Número de registro/ Licencia/ Ensayo clínico/ Certificado/ Otro.

RF 1.1 Editar Incidencia.

Corresponde con la modificación de los datos de una incidencia que haya sido creada y su estado aun sea nueva. Se podrán editar todos los datos menos el No de la Incidencia.

- Resultados de la Evaluación,
 - a. Dictamen: Procede, No procede, el dictamen es de selección simple, una vez se evalúe cambia al estado a procede o no procede.
 - b. Detalles de la Evaluación (entrada de texto libre, no obligatorio).

RF 1.2 Buscar Incidencia.

El usuario podrá realizar las búsquedas por el parámetro que desee, en función de los datos captados de las Incidencia, en el caso de la Fecha se podrá realizar para un día específico o en un rango de fecha en función de la fecha de creación.

RF 1.3 Cancelar Incidencia.

Cambia el estado de la incidencia ha cancelado, las mismas no se eliminan del sistema. De la acción se registra:

- Estado (pasa a estado cancelada, obligatorio).
- Fecha de Cancelación (obligatorio, formato dd/mm/aa).
- Cancelado por: Nombre y Apellidos del usuario autenticado.

RF 1.4 Mostrar Incidencia.

Se mostrará en la pantalla inicial los datos más importantes de las incidencias existentes en el sistema: Código de la Incidencia, Clasificación Incidencia, Organización, Fecha de ocurrencia, Fecha de Emisión y Estado.

RF 2 Asignar Incidencia.

Se corresponde con la asignación de una incidencia al especialista que se encargara de la evaluación.

De la misma se solicita:

- Asignado a: (Obligatorio, se selecciona Nombre y Apellidos a partir de las personas registradas en el sistema como Especialistas y que su cargo sea Jefe de Área).
- Fecha de Reporte: (Obligatorio, se toma automáticamente a partir de la fecha del servidor).

RF 3 Evaluar Incidencia.

Permite al especialista evaluar las incidencias asignadas a ellos y de cada una emitir un dictamen: procede o no procede, de manera tal que al emitir el dictamen el estado de la misma coincida con el dictamen emitido.

RF 4 Enviar Notificaciones a los especialistas.

RF 5 Mostrar Gráfica del Total de Incidencias VS Clasificación por año actual.

RF 6 Mostrar Gráfica de Total de Incidencias por Clasificación vs Estado.

Lista de Requisitos No Funcionales

Apariencia o Interfaz Externa.

RNF 1.1 El sistema debe tener una interfaz fácil de usar y amigable para que pueda ser utilizada sin mucho entrenamiento por el usuario.

RNF 1.2 La entrada de datos incorrecta será detectada claramente e informada al usuario mediante mensajes.

Confiabilidad

RNF 2.1 El sistema podrá estar inactivo durante 5 minutos, al cabo de ese tiempo se cerrará la sesión del usuario que estaba interactuando con él.

RNF 2.2 El sistema deberá guardar y dar la posibilidad de acceder a la información de la gestión de cada usuario. Cada vez que se haga una modificación deberá ser posible conocer que usuario la realizó.

RNF 2.3 Disponible el 100% del tiempo de utilización. El sistema debe estar disponible para su utilización cada vez que el usuario lo requiera. Solamente debe cesar la disponibilidad cuando sea necesario realizar mantenimiento.

Seguridad

RNF 3.1 El usuario debe autenticarse antes de entrar al sistema.

RNF 3.2 Garantizar el acceso controlado a la información. Este debe influir sobre cómo se presentan las interfaces para cada usuario dependiendo del nivel de acceso a la información.

RNF 3.3 El SI debe garantizar que cada usuario solo pueda acceder a la información a que está autorizado, con los derechos que le correspondan y que le sean previamente asignados.

RNF 3.4 Los datos de la aplicación solo podrán ser modificados y visualizados por aquellas personas autorizadas para ello a excepción del código. Los perfiles de usuario de la aplicación serán cinco: Administrador, Directivo, Especialista Superior Calidad, Especialista Calidad, Evaluador y Persona de Contacto.

RNF 3.5 Los usuarios con rol Especialista podrán realizar las búsquedas por el parámetro que desee, en función de los datos captados de las incidencias principalmente por el código que identifica a cada una de estas. La búsqueda por la fecha se podrá realizar para las incidencias que hayan sido creados ese mismo día, entre dos fechas e igual, antes o después de una fecha determinada.

RNF 3.7 Los usuarios con rol Especialista podrán una incidencias que este en el sistema y que su Estado sea Resuelta persistiendo los datos de los mismos en el historial del sistema.

Usabilidad

RNF 4.1 El sistema deberá facilitar la interacción usuario – sistema con el objetivo de evitar rechazo en el uso del mismo, y guiará mediante mensajes al usuario en las diferentes acciones que realice.

RNF 4.2 El usuario podrá acceder solo a las vistas de gestiones donde tenga permiso a través de los botones del menú de navegación.

RNF 4.4 A la hora de utilizar campos de selección y múltiple selección se podrá filtrar en el componente el contenido deseado para una búsqueda más rápida.

Portabilidad

RNF 5.1 El sistema será multiplataforma (Linux o Windows).

Hardware

RNF 6.1 Las conexiones del cliente con el servidor se realizan mediante el protocolo HTTPS. Un servidor de base de datos que debe tener instalado el sistema gestor de base de datos PostgreSQL v9.2.

RNF 6.2 Contar con servidor profesional para desplegar la aplicación, aunque se podrá desplegar en una máquina virtual que se encuentre corriendo sobre un servidor físico de tales características.

RNF 6.3 Contar con al menos 8 GB de memoria RAM.

RNF 6.4 Contar con al menos 500 GB de capacidad en HDD.

Software

8.1 Una PC Cliente que puede ser de tipo portátil o de escritorio la cual debe tener instalado un navegador web preferentemente Mozilla Firefox v50.0 o superior.

8.2 Un Servidor Web que debe tener instalado el contenedor web Apache Tomcat v7.0.52 que permita las conexiones con el cliente, generando las respuestas a sus peticiones.

2.3.2 Justificación de los actores del sistema

Un actor del sistema representa cualquier persona, entidad o sistema externo al mismo que interactúe con él para intercambiar información o como receptor pasivo de esta. Pueden representar el rol que juega una o varias personas, un equipo o un sistema automatizado.

Cada trabajador del negocio (inclusive si fuera un sistema ya existente) que tiene actividades a automatizar es un candidato a actor del sistema. Si algún actor del negocio va a interactuar con el sistema, entonces también será un actor del sistema.

Tabla 4 Justificación de los actores del sistema.

Actor	Descripción
Especialistas Superior Calidad	Especialista encargado registrar y reportar las Incidencias.
Evaluador Superior	Especialista del CECMED que entre sus funciones tiene la de asignar incidencias a sus especialistas utilizando el sistema.

Evaluador	Especialista encargado de evaluar incidencias.
Cliente	Se beneficia del sistema al poder registrar las incidencias que desee.

2.3.3 Diagrama casos de uso del sistema

El diagrama de casos de uso del sistema representa la interacción entre los actores del mismo y los casos de uso con que se relacionan. Se muestra, además, la relación entre casos de uso, ya sea extensión, inclusión, asociación o generalización/especialización y las relaciones entre actores, por asociación o generalización/especialización. Sirve como contrato entre el cliente y los desarrolladores. Se utiliza como entrada esencial para las actividades de análisis, diseño y prueba. (Ver Figura 8 Diagrama de casos de uso del sistema)

Aplicación de patrones de casos de uso

Los casos de uso son artefactos narrativos que describen, bajo la forma de acciones y reacciones, el comportamiento del sistema desde el punto de vista del usuario. Para la obtención de los casos de uso se emplearon los patrones de casos de uso mencionados en el capítulo uno. El patrón CRUD propone formar un caso de uso a partir de los requisitos funcionales relacionados con las acciones de insertar, listar o mostrar, modificar, y eliminar una determinada información. (Anexo 1: Patrones de casos de uso. Solución del patrón CRUD). Para cada una de las incidencias que son objeto de informatización, se repiten estos requisitos que se agruparon dando lugar a los Casos de Uso Gestionar. Se le suman además operaciones relacionadas con otras funciones específicas.

Toda aplicación Web debe permitir la autenticación de los usuarios en el sistema para habilitar los privilegios que le permitan acceder al resto de los servicios. De igual forma es necesario que el usuario se desconecte o finalice su sesión al terminar. Un caso de uso no debe ser inicializado por más de un actor, cuando dos o más actores intervienen en un mismo caso de uso se debe modelar de forma tal que no se incumpla con esta primera regla. Existe un patrón de casos de uso que le da solución a este problema de una forma sencilla. El patrón Múltiples Actores propone crear un nuevo actor del sistema que represente una generalización.

La solución del patrón aplicada al DCUS da como resultado la creación de un nuevo actor Usuario, que generaliza las responsabilidades de los actores Especialista Evaluador, Especialista Calidad, Especialista Superior de Calidad, Jefe Departamento y Cliente, para representar su interacción con el caso de uso Gestionar Incidencia.

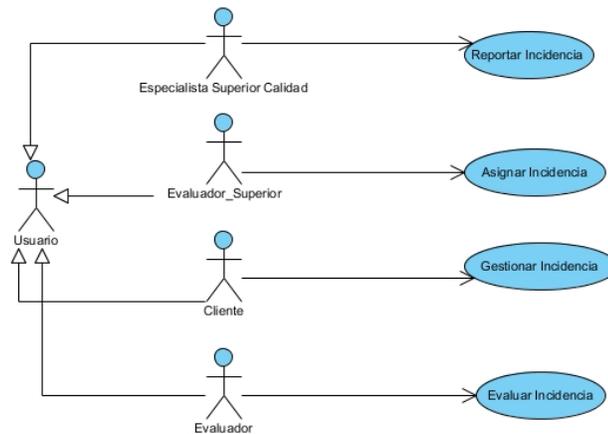


Fig. 7 Diagrama de Clases del Modelo de Objetos para el trabajador del negocio Jefe DGC.

2.3.4 Descripción textual Caso de Uso del Sistema Gestionar Incidencias.

Tabla 5 Descripción textual del Caso de Uso del sistema Gestionar Incidencias.

Caso de Uso:	Gestionar Incidencias.
Actor(s):	Cliente, Especialista Calidad, Especialista Superior Calidad
Resumen:	<p>El Caso de Uso que se describe se divide en 4 secciones las cuales se explican a continuación:</p> <p>Registrar Incidencia: permite a los usuarios del sistema con el rol Cliente, Especialista Calidad o Especialista Superior Calidad registrar en el sistema las incidencias detectadas, para ello los mismos accederán a la funcionalidad Registrar Incidencia y en función de su clasificación registrarán los datos que les solicitan para cada una de ellas. Para el caso de los usuarios con el rol cliente solo podrán registrar las incidencias relativas a la entidad que ellos representan sin embargo, los usuarios con el rol Especialista Calidad y Especialista Superior Calidad podrán registrar las incidencias a nombre de los clientes.</p> <p>Buscar Incidencia: permite a los usuarios realizar las búsquedas por el parámetro que desee, en función de los datos captados de las Incidencias.</p> <p>Editar Incidencia: Permite la corrección de los datos captados en las incidencias registradas en el sistema, para ello los usuarios con los rol Cliente, Especialista Calidad o Especialista Superior Calidad realizará una búsqueda en el sistema en función de un criterio previamente establecido, una vez encuentre la incidencia que desea corregir, la selecciona y cambia los datos deseados.</p> <p>Cancelar Incidencia: La cancelación de las incidencias permite a los usuarios con los roles Cliente, Especialista Calidad, Especialista Superior Calidad declarar que no se</p>

	quiere proceder con la evaluación de la incidencia registrada en el sistema. Para ello los usuarios con el rol Cliente, Especialista Calidad, Especialista Superior Calidad realizará una búsqueda en el sistema en función de criterios previamente establecidos, una vez encuentre la incidencia que se desea cancelar, se selecciona y se cancela.
Precondiciones:	El usuario tiene que estar autenticado en el sistema.
Referencias:	RF 9.1, RF 9.2, RF 9.3, RF 9.4
Prioridad:	Alta
Flujo Normal de Eventos	
Acción del Actor	Respuesta del Sistema
1. El usuario accede al módulo de Incidencias.	<p>2. Muestra en la pantalla inicial un componente de tipo tabla con las Incidencias existentes en el sistema y los datos más importantes de cada una: Código de la Incidencia, Clasificación Incidencia, Organización, Fecha de Ocurrencia, Fecha de Emisión y Estado. Además se muestran las siguientes acciones:</p> <p>a) Registrar Incidencia: esta acción solo podrá ser realizada por el cliente, el especialista de calidad, el especialista superior de calidad.</p> <p>b) Editar Incidencia: esta acción solo puede ser ejecutada por el usuario con el rol cliente, especialista de calidad y especialista superior de calidad.</p> <p>c) Cancelar Incidencia: las incidencias podrán ser canceladas por los usuarios con rol cliente, especialista calidad, especialista superior calidad.</p> <p>d) Buscar Incidencia: la búsqueda podrá ser realizada por los usuarios con rol cliente, especialista calidad, especialista superior de calidad por el parámetro que desee, en función de los datos captados de las Incidencias, en el caso de la Fecha se podrá realizar para un día específico o en un rango de fecha en función de la fecha de creación.</p> <p>e) Evaluar Incidencia: se corresponde con la evaluación de una incidencia por parte del usuario con el rol evaluador que se hará cargo de la investigación.</p>

	f) Asignar Incidencia: se corresponde con la asignación de una incidencia por parte del usuario con el rol especialista calidad o especialista calidad superior al usuario con el rol evaluador superior que su cargo sea Jefe de Área.
3. Selecciona la opción deseada.	
Sección "Buscar Incidencia"	
Acción del Actor	Respuesta del Sistema
1. Selecciona en el componente la opción Buscar Incidencia.	2. Muestra una ventana modal desde la cual el usuario seleccionando la acción "+" de Añadir Filtro podrá añadir los filtros necesarios para realizar una búsqueda dinámica según sus necesidades. Estos filtros muestran todos los campos de una incidencia: No. Incidencia, Clasificación Incidencia, Organización, Fecha de ocurrencia, Fecha en que se emite la Incidencia y Estado. La búsqueda por la Fecha se podrá realizar para un día específico o en un rango de fecha en función de la fecha de creación de la incidencia.
3. Añade los filtros necesarios para realizar la búsqueda en función de los datos captados de la incidencia.	
4. Selecciona y/o añade los datos en los filtros de búsqueda.	
5. Selecciona en la ventana modal la acción Buscar.	6. El sistema busca en BD las incidencias que corresponda con la combinación de los parámetros y/o criterios de búsquedas seleccionados.
	7. Muestra todas las incidencias que coinciden con el criterio seleccionado y de las mismas muestra en el componente de tipo tabla siguientes datos: No. Incidencia, Clasificación de la Incidencia, Organización, Fecha de ocurrencia, Fecha en que se emite la Incidencia, Tiempo Restante y Estado.
Flujo Alternativo 1: No hay incidencias que cumplan con el criterio seleccionado.	
	7. El sistema muestra el componente vacío y un mensaje en el mismo: "No hay registros que coincidan con los criterios de búsquedas seleccionados, Por favor vuelva a intentarlo. "

Sección "Registrar Incidencia"	
Acción del Actor	Respuesta del Sistema
1. Selecciona en el componente la opción Registrar incidencia.	2. Muestra un formulario para introducir los datos de la incidencia que el usuario desea registrar: Clasificación de la Incidencia, Organización, Dirección, País, Número de Teléfono, Número de Fax, Correo electrónico, Datos del representante del reclamante, Persona de Contacto, Descripción del producto o servicio, Número de referencia del producto o pedido, Motivo de la Incidencia, Fecha de ocurrencia, Fecha en que se emite la Incidencia, Reportado por, Adjuntos: Lista de documentos que se adjuntan. Ver RF 7.1
3. Introduce los datos en el formulario.	
4. Selecciona la acción Guardar.	5. Verifica que los datos introducidos en el formulario sean correctos y de ser así guarda en la BD todos los datos introducidos.
	6. Guarda los datos introducidos: Clasificación de Incidencia, Núm. de la Incidencia (se genera automáticamente por el sistema), Organización, Dirección, País, Número de Teléfono, Número de Fax, Correo electrónico, Datos del representante del reclamante, Persona de Contacto, Descripción del producto o servicio, Número de referencia del producto o pedido, Motivo de la Incidencia, Fecha de ocurrencia, Fecha en que se emite la Incidencia, Reportado por, Adjuntos: Lista de documentos que se adjuntan y toma Estado: Nueva.
	7. Emite una notificación al usuario como acuse de recibo en el que constarán: nombre, cargo y fecha en que se recibió su incidencia.
	8. Muestra en pantalla los datos más importantes de la nueva incidencia creada con Estado: Nueva además: No. Incidencia, Clasificación de la Incidencia, Organización, Fecha de ocurrencia, Fecha en que se emite la Incidencia, Tiempo Restante.
Flujo Alternativo 1: Si el usuario con el rol Cliente registra la incidencia.	

	2. Se modifica en el paso 2 la acción relativa a la selección del campo persona de contacto y reportado por, los cuales son el Nombre y Apellidos del usuario autenticado.
Flujo Alternativo 2: Si el usuario que registra la incidencia es Especialista calidad o Especialista superior calidad.	
	2. Se modifica en el paso 2 la acción relativa a la selección del campo persona de contacto el componente se llena con el nombre y apellidos de todos los usuarios en el sistema que sean Clientes y el dato reportado por, se llena automáticamente a partir del Nombre y Apellidos del usuario autenticado
Flujo Alternativo 3: El usuario deja campos vacíos y/o introduce información que no respeta el formato definido para cada uno de ellos.	
	6. Muestra un mensaje de error: "Se han encontrado errores en el formulario, por favor rectifíquelos".
	7. El sistema marca los campos en los que hay errores e indica como corregirlos. Obligatorio: El campo es obligatorio. Continúa en el paso 4 del flujo normal de eventos.
Sección "Editar Incidencia"	
Acción del Actor	Respuesta del Sistema
1. Ejecuta la opción Buscar Incidencia.	
2. Selecciona en el componente de la pantalla inicial la Incidencia que desea editar	
3. Selecciona en el componente la acción Editar Incidencia.	4. El sistema chequea que el estado de la incidencia sea Nueva.
	5. Continúa en el paso 2 del flujo normal de la sección Registrar Incidencia.
Sección "Cancelar Incidencia"	
Acción del Actor	Respuesta del Sistema
1. El usuario ejecuta la opción Buscar Incidencia.	
2. Selecciona la Incidencia que desea cancelar.	
3. Selecciona la opción Cancelar Incidencia.	4. El sistema chequea que el estado de la incidencia sea nueva.

	5. Muestra una ventana modal con el mensaje: “¿Desea cancelar la incidencia seleccionada?”.
6. Selecciona en la ventana modal la acción Aceptar.	7. El sistema cambia el estado de la incidencia a Cancelada.
	8. Muestra un mensaje: La incidencia No. xxx ha sido cancelada satisfactoriamente
Flujo Alternativo 1: El estado de la incidencia no es nueva y el usuario es Cliente	
	5. Muestra mensaje indicando que la operación de cancelación no se puede realizar porque se está evaluando la incidencia. “La incidencia no puede ser cancelada porque se encuentra siendo evaluada, contacte con Departamento de Calidad para cancelar su incidencia”.
Flujo Alternativo 2: El estado de la incidencia no es nueva y el usuario es Especialista de Calidad y Especialista Calidad Superior.	
	5. Muestra un mensaje para confirmar la acción: “¿Seguro que desea cancelar la solicitud?”.
Flujo Alternativo 3: No quiere cancelar la Incidencia.	
5. Selecciona en la ventana modal la acción Cancelar. Finaliza el escenario.	

2.4 Conclusiones parciales

El estudio realizado a los procesos que se llevan a cabo en el DGC del CECMED ayudó a comprender su entorno de trabajo. Se identificaron posibles mejoras al determinar los procesos que se deben automatizar. Se identificaron los requisitos funcionales y no funcionales con los cuales se llegó a una definición más formal de lo que el sistema debe hacer y sus características, contribuyendo a establecer un convenio con los clientes de las características funcionales del software.

Se identificaron los casos de uso y la relación con los actores del sistema, esta relación se especificó en detalle en el diagrama de casos de uso del sistema, realizando además la descripción de los casos de uso. Para hacer una valoración del trabajo se emplearon técnicas de validación de requisitos, lo que dio como resultado la aprobación por parte del cliente de los requerimientos identificados asegurando la calidad de los mismos. El desarrollo de estos flujos de trabajo y los artefactos obtenidos, permiten comenzar con el análisis y diseño de la aplicación propuesta.

Capítulo 3

Diseño, implementación y prueba del sistema

Introducción

En el presente capítulo se hace referencia a los aspectos correspondientes al diseño, la implementación y las pruebas, teniendo presente la propuesta del sistema y las funcionalidades seleccionadas. Se confeccionan los diagramas de clases del diseño el cual contribuye a la consolidación de una arquitectura estable y el Diagrama Entidad-Relación que permite identificar los objetos de datos y sus relaciones. Se especifican además los patrones de diseño. Se realiza el diagrama de despliegue del sistema y se organizan los componentes mediante el diagrama de componentes para definir la estructura en capas de la aplicación. Se define el método de prueba a seguir y los resultados de las mismas, una vez aplicadas al sistema, contribuyendo a mejorar la calidad, usabilidad e identificar fallos en la aplicación.

3.1 Arquitectura de Grails

La arquitectura de Grails está concebida por 3 capas lógicas principales: Web Layer, Service Layer y Data Layer, su interacción se lleva a cabo mediante interfaces que definen funcionalidades que la misma debe brindar, también llamadas fachadas cuya función fundamental es asegurar que el acoplamiento sea el más bajo posible y la abstracción del funcionamiento de la capa inferior. (27)

Grails implementa el patrón (Modelo-Vista-Controlador, MVC) en la que se separa la lógica empresarial de la presentación de la aplicación, lo cual le permite cambiar fácilmente el aspecto de su aplicación sin modificar su comportamiento.



Fig. 8 Arquitectura de Grails (27)

3.2 Arquitectura del sistema

La arquitectura del sistema se refiere a “una vista del sistema que incluye los componentes principales del mismo, la conducta de esos componentes según se la percibe desde el resto del sistema y las formas en que los componentes interactúan y se coordinan para alcanzar la misión del sistema”. (46)

3.2.1 Capas lógicas del sistema

Capa Presentación: Los componentes de esta capa son los encargados de proveer una interfaz entre el sistema y el usuario, básicamente es responsable de brindarle información al usuario por parte del sistema y viceversa. A esta capa pertenecen los archivos Java Script (.js) encargados de validar los campos y velar que sean insertados correctamente; las vistas (.jsp) le permiten al usuario interactuar con el sistema de forma directa; los controladores que son los que reciben las solicitudes hechas por el usuario, realiza las operaciones de lógica de negocio sobre los modelos y decide la vista que será mostrada y las hojas de estilo (.css) las cuales por medio de reglas brindan un formato y colores a los componentes de la vista.

Capa de Negocio: Esta capa contiene los servicios que son los componentes encargados de implementar la lógica de negocio de la aplicación. Se comunica con la capa de presentación, para recibir las peticiones que el usuario ha realizado y presentar los resultados, y con la capa de datos, para solicitar al gestor de base de datos almacenar o recuperar datos de él.

Capa de Datos: Es la capa donde se almacenan los datos mediante la capa de negocio, encargada de ofrecer, modificar, almacenar, borrar y recuperar datos, mediante el gestor (o los gestores) de bases de datos que la aplicación requiera. El repositorio de datos descansa sobre XML para la correcta construcción de la capa de presentación y está formada por uno o más gestores de bases de datos que realizan todo el almacenamiento de datos y reciben solicitudes de almacenamiento o recuperación de información.

3.3 Patrones utilizados

Un patrón es conocido como “una descripción de un problema y la solución, a la que se da un nombre, y que se puede aplicar a nuevos contextos; idealmente, proporciona consejos sobre el modo de aplicarlo en varias circunstancias, y considera los puntos fuertes y compromisos”. (22) En otras palabras, brindan una solución ya probada y documentada a problemas de desarrollo de software que están sujetos a contextos similares.

Patrones Arquitectónicos

Un patrón arquitectónico especifica un conjunto predefinido de subsistemas con sus responsabilidades y una serie de recomendaciones para organizar los distintos componentes. Resuelve problemas

arquitectónicos, de adaptabilidad a requisitos cambiantes, performance, modularidad, acoplamiento, entre otros. (22)

Modelo Vista Controlador (MVC)

Este patrón separa los datos de una aplicación, la interfaz de usuario, y la lógica de control en tres componentes distintos. Grails sigue un patrón de diseño muy popular en el mundo del desarrollo de aplicaciones web, el MVC, el cual propone que los componentes del sistema estén organizados en 3 capas diferentes según su misión dentro del mismo. (27)

- **Modelo o Capa de Datos:** esta capa se encarga de encapsular los datos y las funcionalidades, contiene además los componentes que representan y gestionan los datos manejados por la aplicación, es decir, son los encargados de realizar las operaciones relacionadas con la base de datos. Está definido por clases de dominio de Groovy las cuales se mapean en la base de datos utilizada y permiten el manejo y almacenamiento de los datos.
- **Vistas o Capa de Presentación:** son los recursos que junto al modelo generado por los controladores le permiten al cliente visualizar la información. La tecnología JSP es utilizada por Grails para la interacción con el usuario, basada en una implementación mediante GSP, el mismo permite a los desarrolladores mezclar las etiquetas de lenguajes de marcas tradicionales como HTML con código Java para producir vistas dinámicas. Los componentes de esta capa son los encargados de interactuar con el usuario, mostrarle las diversas acciones disponibles y el estado actual de los datos del sistema.
- **Controlador o Capa de Control:** posee los componentes que luego de recibir las entradas u órdenes hechas por el usuario, los cuales son traducidos a solicitudes de servicio que gestionan la aplicación de la lógica del negocio sobre el modelo de datos y determina la vista que se debe mostrar a continuación. Es el encargado de mantener el flujo de comunicación entre las vistas y el modelo, están definidos por los GroovyControllers o controladores Groovy. Una de las actividades fundamentales llevadas a cabo por los controladores en Grails es obtener los datos que serán mostrados en la vista.

Cuando se dice que la capa de control “gestiona la aplicación de la lógica de negocio” se hace referencia a que estos son los responsables de que esta se aplique, esto no quiere decir que se implemente la lógica de negocio en las clases controladoras, esta es implementada en una cuarta capa que sería la de servicios.

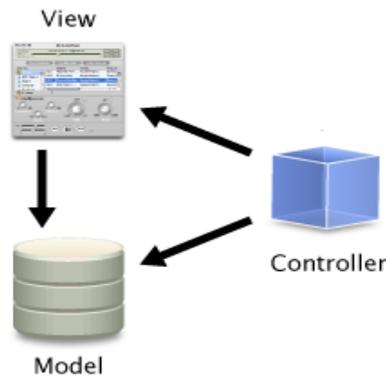


Fig. 9 Modelo-Vista-Controlador

3.3.2 Patrones de diseño

Un patrón de diseño está relacionado con los aspectos del diseño de los subsistemas. Es una solución estándar para un problema común de programación y una técnica para flexibilizar el código haciéndolo satisfacer ciertos criterios. (22)

Patrones GRASP

GRASP (Generales de Software para Asignación de Responsabilidades) son patrones generales de software que describen los principios fundamentales de la asignación de responsabilidades a objetos. El nombre se eligió para indicar la importancia de captar estos principios, con el objetivo de diseñar el software de manera eficaz. (22)

- **Controlador:** este patrón sugiere que la lógica de negocios debe estar separada de la capa de presentación para aumentar la reutilización de código y a su vez tener un mayor control del flujo de eventos del sistema, se encarga de asignar responsabilidades a clases específicas facilitando la centralización de actividades como la validación, la seguridad, entre otras. En Grails los controladores son los encargados de posibilitar una capa intermedia entre las vistas y el modelo.
- **Alta cohesión:** se aplica en la mayoría de las clases del diseño las cuales deben ser coherente, ya que en cada una solo se implementan las funcionalidades que le corresponden.
- **Bajo acoplamiento:** se refiere a tener las clases lo menos relacionadas entre sí que se pueda, de forma tal que al producirse una modificación en alguna de ellas, tenga la menor repercusión posible en el resto de las clases, incrementando la reutilización y disminuyendo la dependencia entre las clases.

- **Experto:** se mantiene el encapsulamiento, los objetos utilizan su propia información para llevar a cabo sus tareas de las clases que tiene. De esta manera se garantiza una alta cohesión y un bajo acoplamiento.

3.4 Diagrama de clases del diseño

Un diagrama de clases proporciona una perspectiva estática que representa el diseño estructural del sistema mostrando un conjunto de clases, sus atributos y las relaciones entre ellos. (47)

Los diagramas de clases son utilizados durante el proceso de diseño de los sistemas, donde se establece el diseño conceptual de la información que contendrá el sistema. A continuación se muestra el diagrama de clases Nueva Incidencia del CUS Gestionar Incidencia los restantes se encuentran en el Anexo 2.

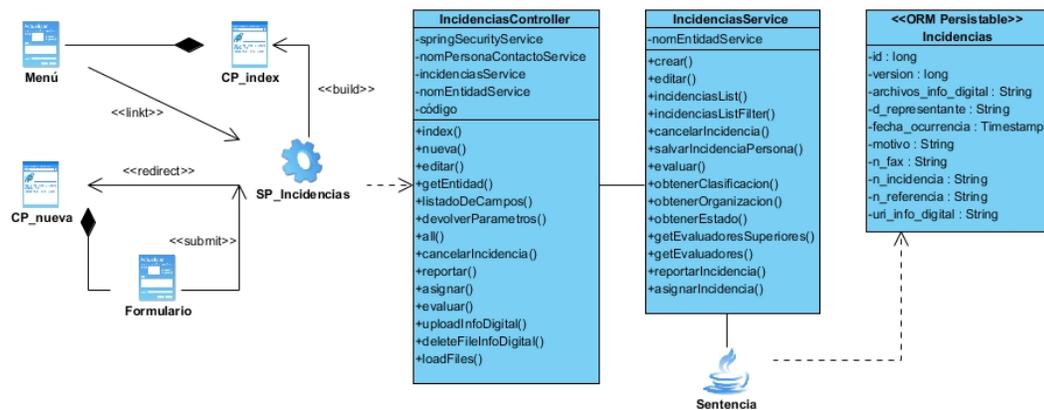


Fig. 10 Diagrama de clase del diseño CU Nueva Incidencia

3.5 Diagrama de clases persistentes

El diagrama de clases persistentes representa las relaciones entre las clases (asociación, agregación/composición y generalización/especialización), además de la información de larga duración o que persiste en el tiempo, es decir, la información que se requiere almacenar para gestionarla en cualquier momento.

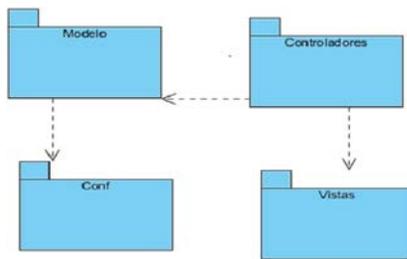


Fig. 13 Paquete de Componentes.

3.7.1 Descripción de los componentes

Paquete de componentes “Vistas”

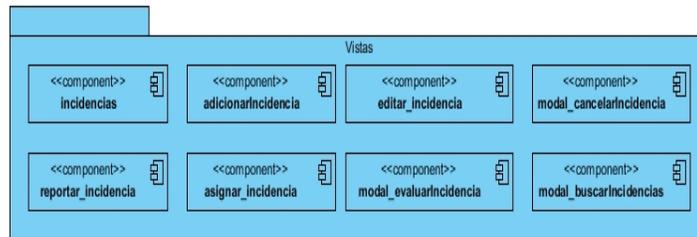


Fig. 14 Paquete Vistas.

El paquete de componente “Vistas” es una representación de las paginas gsp con las cuales el usuario interactúa, permitiendo la entrada de datos al sistema y realizar consulta de dichos datos.

3.7.2 Paquete de componentes “Controladores”

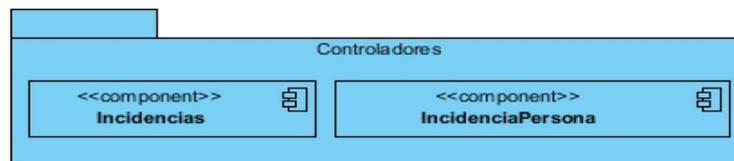


Fig. 15 Paquete Controladores.

El paquete de componentes “Controladores” es una representación de las operaciones que permiten el flujo y la comunicación entre las vistas y el modelo, son los encargados de recibir las órdenes por parte del usuario, gestionar la ejecución de la lógica de negocio y posteriormente actualizar la vista para que el usuario pueda ver como ha quedado el modelo de datos tras las actualizaciones.

3.7.3 Paquete de componentes “Modelo”

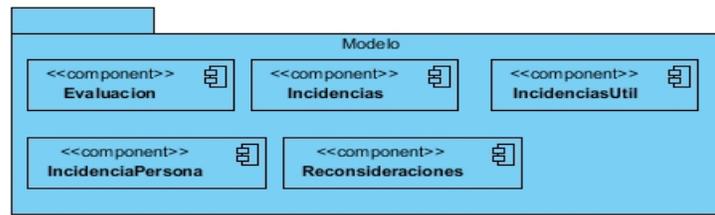


Fig. 16 Paquete Modelo

Son las clases que representan todo el negocio del sistema. Estas clases del dominio constituyen entidades o tablas que son utilizadas en la base de datos mapeadas por el GORM, siendo este un gestor de persistencia para controlar el ciclo de vida de dichas entidades.

3.7.4 Paquete de componentes “Conf.”

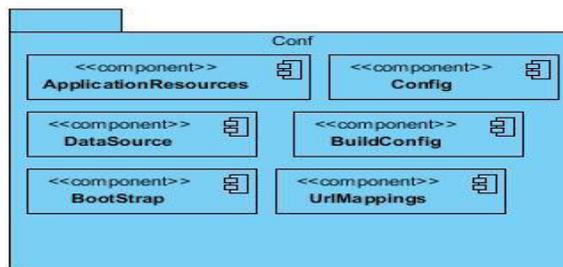


Fig. 17 Paquete Conf.

Este paquete es utilizado para la configuración y definición del acceso local o remoto a los datos. Constituye un componente del Framework donde se ubican ficheros Groovy de configuración. Garantiza el control de las acciones en los controladores para lograr el acceso a las acciones realizadas por cada usuario del sistema.

3.8 Diagrama de despliegue

“El diagrama de despliegue se utiliza para modelar la disposición física de los componentes de hardware utilizado en la implementación del sistema y la relación entre cada uno de ellos”. (48)



Fig. 18 Diagrama de Despliegue

Para el despliegue del sistema es necesario:

- Una PC Cliente que puede ser de tipo portátil o de escritorio la cual debe tener instalado un navegador web preferentemente Mozilla Firefox v50.0 o superior.
- Un Servidor Web que tiene instalado el contenedor web Apache Tomcat v7.0.52 que permita las conexiones con el cliente, generando las respuestas a sus peticiones. Las conexiones del cliente con el servidor se realizan mediante el protocolo HTTPS.
- Un servidor de base de datos que debe tener instalado el sistema gestor de base de datos PostgreSQL v9.2.

3.9 Pruebas del Sistema

El único instrumento adecuado para determinar el estado de la calidad de un producto software es el proceso de pruebas. En este proceso se ejecutan pruebas dirigidas a componentes del software en específico o al sistema de software en su totalidad, con el objetivo de medir el grado en que se cumplen los requisitos. En las pruebas se usan casos de prueba, especificados de forma estructurada mediante técnicas de pruebas. (49)

3.9.1 Métodos de prueba

Los métodos de prueba definen la estrategia a seguir en función de la verificación y validación del sistema diseñados para descubrir fallos. Los métodos más significativos son las **pruebas de caja blanca** y las **pruebas de caja negra**.

Pruebas de Caja Blanca: es un método de diseño de casos de prueba que usa la estructura de control del diseño procedimental para obtener los casos de prueba. Se encarga de comprobar los caminos lógicos del software a través de los casos de prueba que se ejerciten con conjuntos específicos de condiciones y/o bucles y sus límites, así como las estructuras de datos. (49)

Pruebas de Caja Negra: denominada también prueba de comportamiento, se centran en los requisitos funcionales del software, tiene como objetivo verificar que la entrada se acepta correctamente y que ejerciten en su totalidad todos los requisitos funcionales de un sistema. (49)

La prueba de caja negra intenta encontrar las siguientes categorías de errores (49):

- Funciones incorrectas o ausentes.
- Errores de interfaz.
- Errores en estructuras de datos o en accesos a bases de datos externos.
- Errores de rendimiento.
- Errores de inicialización y de terminación.

Para desarrollar la **prueba de caja negra** existen varias técnicas, entre ellas están:

- Técnica de la Partición de Equivalencia: esta técnica divide el campo de entrada en clases de datos que tienden a ejercitar determinadas funciones del software.
- Técnica del Análisis de Valores Límites: esta técnica prueba la habilidad del programa para manejar datos que se encuentran en los límites aceptables.
- Técnica de Grafos de Causa-Efecto: es una técnica que permite al encargado de la prueba validar complejos conjuntos de acciones y condiciones.

Dentro del método de Caja Negra la técnica de la Partición de Equivalencia es una de las más efectivas pues permite examinar los valores válidos e inválidos de las entradas existentes en el software, descubre de forma inmediata errores que de otro modo, requerirían la ejecución de muchos casos antes de detectar el error genérico. La Partición de Equivalencia se dirige a la definición de casos de pruebas que descubran clases de errores, reduciendo así en número de clases de prueba que hay que desarrollar. (50)

3.9.2 Casos de prueba

Un caso de prueba es un conjunto de entradas de pruebas, condiciones de ejecución, resultados esperados desarrollados para cumplir un objetivo en particular o una función esperada. La entidad más simple que siempre es ejecutada como una unidad, desde el comienzo hasta el final.

Los casos de prueba deben verificar:

- Si el producto satisface los requisitos del usuario, tal y como se describe en las especificaciones.
- Si el producto se comporta como se desea, tal y como se describe en las especificaciones funcionales del diseño.

Un caso de prueba se diseña según las funcionalidades descritas en los caso de uso, este es elaborado previo a la realización de las pruebas funcionales de la aplicación. Cada plantilla de caso de prueba recoge la especificación de un caso de uso, dividido en secciones y escenarios, para hacer más fructífera

la ejecución de las pruebas. A continuación se muestra el diseño del caso de prueba, correspondiente al flujo básico del CU Autenticar Usuario y Nueva Incidencia. Las pruebas restantes se podrán ver en el anexo 5.

Descripción general: Permite a los especialistas autenticarse en el sistema.

Tabla 6 Diseño de casos de prueba Autenticar Usuario.

Escenario	Descripción	Nombre de Usuario	Contraseña	Respuesta del sistema	Flujo central
EC 1.1: Autenticar Usuario	Se autentica un usuario en el sistema satisfactoriamente.	V Admin	V cecmecPS	Carga la configuración para el usuario autenticado	1. Introducir el Usuario y la Contraseña. 2. Selecciona la opción "Acceder".
EC 1.2: Datos de Autenticación incorrectos	Se muestra un mensaje indicando que existen datos incorrectos.	V Admin	F CecmedPS	Muestra un mensaje indicando que la contraseña no es correcta.	1. Introducir el Usuario y la Contraseña. 2. Selecciona la opción "Acceder".
		F Admin	F CecmedPS	Muestra un mensaje indicando que el usuario y la contraseña proporcionada no son correctos.	

Descripción general: Permite a los especialistas registrar una nueva incidencia.

Condiciones de ejecución: Los especialistas tienen que estar autenticados y haber seleccionado la opción "Nueva Incidencia".

Tabla 7 Diseño del caso de prueba Nueva Incidencia.

Nombre del Caso de Prueba	Registrar Incidencia	EC: 1.1
Entrada	Clasificación: Queja Persona de Contacto: Kenia Macías Organización: CECMED Dirección: 3ra e/ 60 y 62 Número de Teléfono: 72720525 Número de Fax: 72720520 Correo electrónico: kenia@cecmec.sld.cu Fecha Ocurrencia: 15/05/2016 Motivo de la Incidencia: Trato inadecuado	
Resultado Esperado	Crea una nueva Incidencia.	
Salida	Se visualiza en la interfaz principal la incidencia creada.	
Observaciones	1. Se selecciona la opción nueva. 2. Se introducen o seleccionan los datos correspondientes. 3. Se selecciona la opción Aceptar. 4. Se crea la Incidencia satisfactoriamente.	
Nombre del Caso de Prueba	Registrar Incidencia	EC: 1.2

Entrada	Clasificación: Reclamación Persona de Contacto: Josefina Ramírez Organización: LABIOFAM Dirección: 3ra e/ 60 y 62 Número de Teléfono: Número de Fax: Correo electrónico: josefina@labiofam.sld.cu Fecha Ocurrencia: 15/05/2016 Motivo de la Incidencia: Errores en la documentación	
Resultado Esperado	Crea una nueva Incidencia.	
Salida	Se visualiza en la interfaz principal la incidencia creada.	
Observaciones	1. Se selecciona la opción nueva. 2. Se introducen o seleccionan los datos correspondientes. 3. Se selecciona la opción Aceptar. 4. Se crea la Incidencia satisfactoriamente.	
Nombre del Caso de Prueba	Existen campos vacíos	EC: 1.3
Entrada	Clasificación: Persona de Contacto: Organización: BIOCUBAFARMA Número de Teléfono: Número de Fax: Correo electrónico: kenia@biocubafarma.sld.cu Fecha Ocurrencia: 15/05/2016 Motivo de la Incidencia: Errores en la documentación	
Resultado Esperado	Crea una nueva Incidencia.	
Salida	Muestra un mensaje de error, marca los campos vacíos o con errores e indica como corregirlos.	
Flujo Central	1. Se selecciona la opción nueva. 2. Se introducen o seleccionan los datos correspondientes. 3. Se selecciona la opción Aceptar. 4. Se crea la Incidencia satisfactoriamente.	

3.10 Resultados de las pruebas

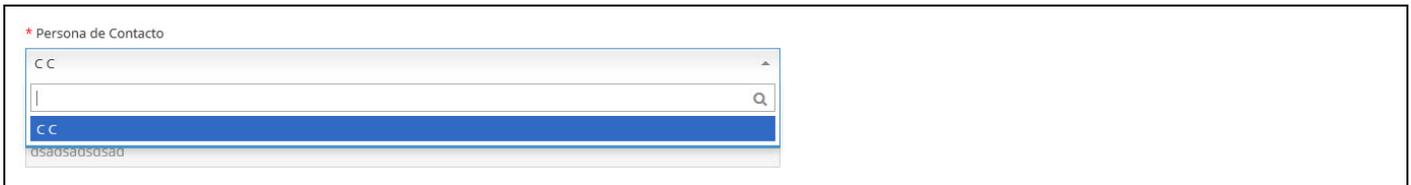
Entre las principales no conformidades encontradas se pueden mencionar:

Como Especialista Superior de Calidad al llenar la solicitud de incidencia no permite seleccionar la entidad solicitante. Error 403 Forbidden

The screenshot shows a web form with a dropdown menu labeled '* Organización de pertenencia'. The dropdown menu is currently displaying 'Select an Option'. Below the dropdown are two input fields: 'Número de Teléfono' and 'Correo Electrónico'. A red arrow points to the dropdown menu.

Como Cliente (C) al crear una incidencia se muestra el campo Persona de Contacto como selección, cuando este debería tomarlo del usuario autenticado.

Según datos del RF 1.1: Persona de Contacto (se toma el nombre y apellidos a partir del usuario autenticado en el sistema, solo para el caso en que el rol del usuario sea Cliente, obligatorio).



Como Cliente (C) al realizar la búsqueda por la Clasificación se muestran todas las incidencias incluidas las que no pertenecen a la entidad que el usuario representa.

21.2016	Queja	softel	05/06/2016
23.2016	Queja	softel	05/06/2016
25.2016	Queja	CIGB	06/12/2016
26.2016	Queja	CIGB	06/05/2016

Como Cliente (C) cuando se elimina una incidencia el sistema no muestra en los detalles: la fecha de la cancelación ni el dato cancelado por.

Número de Incidencia: 26.2016 **Fecha Evaluación:** -

Clasificación: Queja **Documentos Adjuntos:** -

Organización: CIGB **Detalles Evaluación:** -

Fecha de Ocurrencia: 06/05/2016

Fecha de Reporte: 29/06/2016

Estado: Cancelada

Como Cliente (C) al realizar la búsqueda después de la Fecha de Emisión seleccionada se muestran incidencias que no cumplen con el criterio seleccionado.

Buscar Incidencias ✕

Añadir Filtro Seleccione... +

Fecha Emisión des... 29/06/2016 📅 ✕

Como Especialista Calidad cuando accede al módulo incidencia no se muestra el icono Registrar Incidencia.

Bienvenido,
ec ▾

Incidencias

Mostrar 10 registros

Número Incidencia

32.2016
31.2016
30.2016
23.2016
3.2016

Mostrando registros del 1 al 5 de un total

Las no conformidades encontradas fueron resueltas satisfactoriamente, logrando un producto más fiable para el cliente.

3.11 Conclusiones parciales

Se realizó la descripción de la solución propuesta a través de un conjunto de artefactos definidos por la metodología RUP generándose los diagramas de clases del diseño elaborados para darle solución al sistema, el diagrama de entidad relación, el diagrama de clases persistente perteneciente al mismo, se muestran los artefactos generados durante la fase de implementación, como el diagrama de despliegue y el diagrama de componentes; los cuales se documentaron mediante el uso de la herramienta de modelación Visual Paradigm. Se realizó el análisis de la arquitectura del sistema propuesta por el marco de trabajo Grails, además se hizo referencia a los patrones seleccionados para el diseño e implementación del sistema. La implementación estuvo enfocada en el cumplimiento de los requisitos funcionales y no funcionales antes definidos. Se realizaron pruebas para demostrar el correcto funcionamiento del sistema siguiendo el método de Caja Negra usando la técnica de Partición de Equivalencia pues permite examinar los valores válidos e inválidos de las entradas existentes en el software. Finalmente se mostraron los resultados obtenidos durante la realización de las pruebas, para determinar que el software posee una calidad aceptable.

CONCLUSIONES GENERALES

Se culmina la presente investigación logrando cumplir los objetivos trazados con técnicas y tecnologías para cada uno de ellos. Se investigó la existencia de soluciones aplicadas a la gestión de la información como punto de partida para definir la implementación de una propia, para el cumplimiento de los objetivos y en concordancia con las necesidades expuestas por el cliente, se requirió hacer un estudio de la situación actual del proceso de gestión de la calidad en el CECMED, para con ello elevar la eficiencia del proceso en cuestión. Se describieron las herramientas y tecnologías a utilizar para la implementación del sistema. Se definieron los requisitos más importantes a ser resueltos con el sistema a desarrollar.

Al utilizar la metodología RUP se generó abundante documentación del sistema que sirve de base para la implementación de versiones posteriores. La utilización de Grails como framework que facilita el desarrollo de aplicaciones web, permitió la reducción de tiempo y esfuerzo por parte del desarrollador, logrando una mayor productividad. El uso del patrón de diseño MVC permitió obtener como resultado un diseño sólido para la implementación de la aplicación. Se generaron los artefactos relacionados con el diseño y la implementación del sistema. Se realizaron pruebas al sistema desarrollado de acuerdo al método escogido obteniendo varias no conformidades que fueron resueltas para satisfacción del cliente.

Recomendaciones

Con los resultados obtenidos en la investigación y la experiencia adquirida en el desarrollo de la aplicación, se recomienda:

- Ampliar las funcionalidades permitiendo enviar al cliente una notificación cuando su incidencia haya sido asignada a especialista que realizara la investigación de su incidencia.
- Implementar reportes de las incidencias evaluadas.
- Continuar la interacción con los usuarios que propicie una retroalimentación, dando así margen a la corrección de posibles errores que no hayan sido detectados.
- Incluir en el Manual de Usuario del CECMED toda la información referente a las incidencias, para un mejor aprovechamiento de la aplicación.

Referencias Bibliográficas

1. Centro para el Control Estatal de Medicamentos Equipos y Dispositivos Médicos. 14 de enero de 2015.
<http://www.cecmed.cu/acerca-de/historia>
2. Valdés Gálvez, Oscar. Capítulo V Principios de la Gestión de la Calidad. Barcelona: Martínez Roca, 2005. Pp. 1-20.
3. García Camargo, Raúl. Capítulo IV Principios de la Gestión de Procesos. Barcelona: Martínez Roca, 2005. Pp. 20-30.
4. Normas ISO 9000. DuocUC. 10 de mayo 2012.
http://www.iso.org/iso/home/standards/management-standards/iso_9000.htm
5. Cruz Ramírez, José. Historia de la calidad en Excellentia, pp. 8-14. 08 de marzo de 2011.
<http://www.tecnologiaycalidad.galeon.com/calidad/6.html>
6. International Organization for Standardization. 08 de marzo de 2012.
http://www.iso.org/iso/home/standards/management-standards/iso_9000.htm
7. Sistemas y Calidad Total.com. 20 de abril de 2011.
<http://www.sistemasycalidadtotal.com/acerca-de/>
8. Portal Web Importancia de la Gestión de la Calidad.
<http://www.normas9000.com/importancia-gestion-calidad.html>
9. Capítulo IV Principios de la Gestión de la Calidad [En línea] [Fecha de consulta: 08 de marzo de 2016.]
10. Gaceta Oficial de la República de Cuba. Compendio Oficial. Documentos rectores. [En línea] [Fecha de consulta: 14 de enero 2016.]
11. QACTION Empresa de Software para el Control de la Calidad. Portal Web de WILSOFT. [En línea] [Fecha de consulta: 14 de enero de 2016.]. <http://www.wilsoft-la.com/QAction>
12. SE Audit Software para la Gestión de Auditorías. Portal Web de SoftExpert.
<http://www.gestion-calidad.com/software-softexpert.html>
13. WILSOFT Empresa de Software para el Control de la Calidad. Portal Web de WILSOFT. [En línea] [Fecha de consulta: 08 de marzo de 2016.]
<http://www.wilsoft-la.com/index.php/productos/qceo.html>
14. PYMES [En línea] [Fecha de consulta: 08 de marzo de 2016.]. <https://es.wikipedia.org/wiki/pymes>
15. KMKey Quality. Software de Gestión de la calidad. 08 de marzo de 2016
http://www.kmkey.com/software_gestion_calidad
16. Portal Web Normalización y Calidad [En línea] [Fecha de consulta: 08 de abril de 2016.]
<http://www.nc.cubaindustria.cu/ncmundomas.html>

17. Portal Web Órgano Nacional de Acreditación de la República de Cuba [En línea] [Fecha de consulta: 08 de abril de 2016.]. <http://www.onarc.cubaindustria.cu>
18. Portal Web de Normalización, Metrología y Calidad [En línea] [Fecha de consulta: 08 de marzo de 2016.]. <http://www.cepec.cu/es/normalizacion>
19. **Menéndez, E.** Herramientas CASE para el proceso de desarrollo de Software. [En línea] 2016 <http://www.monografias.com>
20. Página Web de Arquitectura y Patrones de diseño. [En línea] [Fecha de consulta: 10 de marzo de 2016.]
21. **Jacobson, Ivan, Booch, Grady y Rumbaugh, James.** El Proceso Unificado de Desarrollo de Software. España: Adison-Wesley.
22. **Object Management Group.** UML. [En línea] 2016. [Fecha de consulta: 15 de marzo de 2016.] <http://www.uml.org/13304>
23. **Visual Paradigm International.** New Releases and Feature Enhancements of Visual Paradigm Products. *Visual Paradigm*. [En línea] Febrero de 2016. [Fecha de consulta: 20 de febrero de 2016.] <http://www.visual-paradigm.com/product/news.jsp#VPUML50>.
24. **SÁNCHEZ, M. A. M.** Metodologías de Desarrollo de Software. 2004, Disponible en: http://www.informatizate.net/articulos/metodologias_de_desarrollo_de_software_07062004.html
25. **Potencie, Fabián. 2006. Symfony Project.** [En línea] 2016. [Fecha de consulta: 1 de marzo de 2016.]. <http://www.symfony-project.org>.
26. Portal Web [En línea] 2016. [Fecha de consulta: 25 de marzo de 2016]. <https://es.wikipedia.org/wiki/framework>
27. **Brito, Nacho.** Manual de desarrollo Web con GRAILS. España: s.n., 2009. ISBN: 978-84-613-2651.
28. **Smith, Glen y Ledbrook, Peter.** Grails in Action. New York: Manning Publications. Co, 2009. ISBN-978-1-933988-93-1.
29. **Mora, Sergio Luján. 2010.** Programación en Internet: Clientes Web. S.I.: Club Universitario, 2010.
30. **Castillo, Alejandro, Cantón.** Manual de HTML 5 en español. [En línea] 2016. [Fecha de consulta: 26 de febrero 2016]. <http://www.theproc.es>
31. Aprende HTML 5 desde 0. [En línea] 2016. [Fecha de consulta: 26 de febrero 2016]. <http://www.tutosytips.com>
32. **Eguíluz, Javier, Pérez.** Introducción a CSS. [En línea] 2016. [Fecha de consulta: 20 de febrero de 2016]. <http://www.librosweb.es>
33. **Navajas, Antonio, Ojeda.** Guía Completa de CSS 3. [En línea] 2016. [Fecha de consulta: 25 de febrero de 2016]. <http://www.antonionavajas.com>

34. Portal Web de NetBeans. [En línea] 2016. [Fecha de consulta: 25 de marzo de 2016]
<https://netbeans.org/&prev=search>
35. Portal Web [En línea] 2016. [Fecha de consulta: 25 de marzo de 2016]
https://es.wikipedia.org/wiki/Intellij_IDEA
36. **Worley, John y Drake, Joshua.** Practical PostgreSQL. Copyright © 2002 Command Prompt, Inc., 2002. ISBN: 1-56592-846-6.
37. **PESCADOR, J. S.** Manual: ¿Qué es una Base de Datos?, Disponible en:
<http://www.mihosting.net/areaclientes/knowledgebase.php?action=displayarticle&catid=12&id=63>.
38. **BURBANO PROAÑO, I. D. J.** Comparativa de base de datos, [PDF]. 2012
39. **ISO 10002:2004 Gestión de la calidad.** Satisfacción del cliente. Directrices para el tratamiento de las quejas en las organizaciones.
40. **Pressman, Roger S.** Ingeniería del Software. Un enfoque Práctico. Quinta Edición. Madrid: McGraw-Hill, 2006. ISBN: 970-10-5473-3.2012
41. **Pérez Cristian, Rafael,** Frías Ferreiro, Gretel y Isa Sánchez, Ana Mayra. Procedimiento para la realización de auditorías internas de la Calidad. Cuarta Edición. CECMED. 2015
42. **Pérez Cristian, Rafael, Frías Ferreiro, Gretel y Campos Carbonell, Tamara.** I.00.002 Atención a Quejas y Reclamaciones. Tercera Edición. CECMED. 2015.
43. **BRPS. Buenas Prácticas Reguladoras Cubanas.** Resolución BRPS No. 05 del 28 de agosto de 2002. [Monografía de internet]. Disponible en: <http://www.Cecmed.sld.cu/Docs/Reg.Farm/DRA/DispGen/Reg./BPR>
44. **IEEE.** Standard IEEE. Enero de 2012.
45. **Pan, D, Zhu, D y Johnson, K.** Requirements Engineering Techniques. Canadá: s.n., 2001. Departamento de Ciencias de la Computación Universidad de Calgary.
46. **Arciniegas Herrera, José Luis.** Architectural patterns regarding web application domain usability. Pages. 52-55, Universidad Nacional de Colombia: s.n., 1980, Vols. Vol. 30, Nº. 1, 2010. ISSN: 0129-5608.
47. **Laman, Craig.** UML y Patrones. Una introducción al análisis y diseño orientado a objetos y al proceso unificado. Segunda Edición. Madrid: Pesaron Educación, s.a., 2003. ISBN: 84-205-3438-2
48. **Pressman, Roger S.** Ingeniería del Software. Un enfoque Práctico. Sexta Edición. Madrid: McGraw-Hill, 2006. ISBN: 970-10-5473-3.
49. **Bolaños Alonso, Daniel, Sierra Alonso, Almudena y Alarcón Rodríguez, Idoia.** Pruebas de software y JUnit. Madrid: Pearson Educación, 2008. ISBN: 978-84-8322-354-3.
50. **Pressman, Roger S.** *Software engineering: a practitioner's approach.* McGraw-Hill Higher Education, 7ma, 2010. Nº de pág. 928. ISSN 970-10-5473-3.

Anexo 1: Patrones de casos de uso

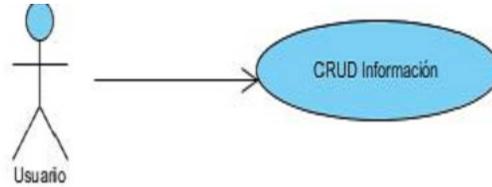


Fig. 19 Patrón CRUD

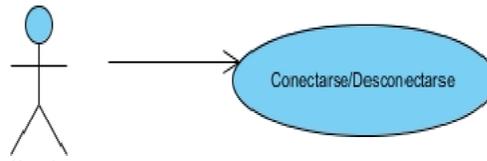


Fig. 20 Patrón Login

Anexo 2: Diagramas de clases del diseño

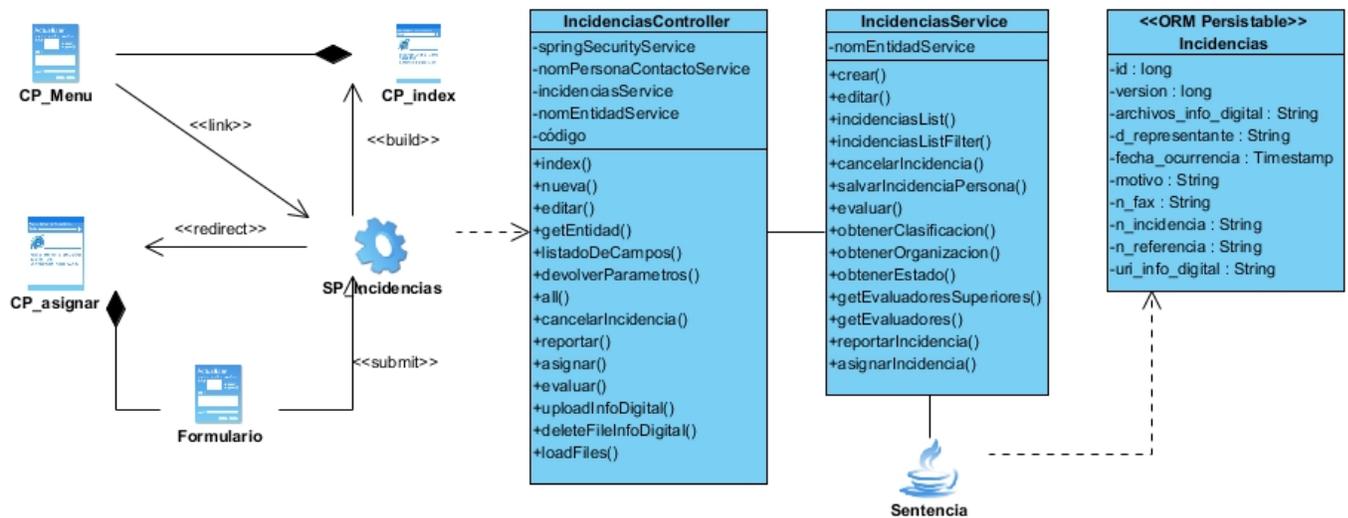


Fig. 21 Diagrama de clases del diseño Asignar Incidencia

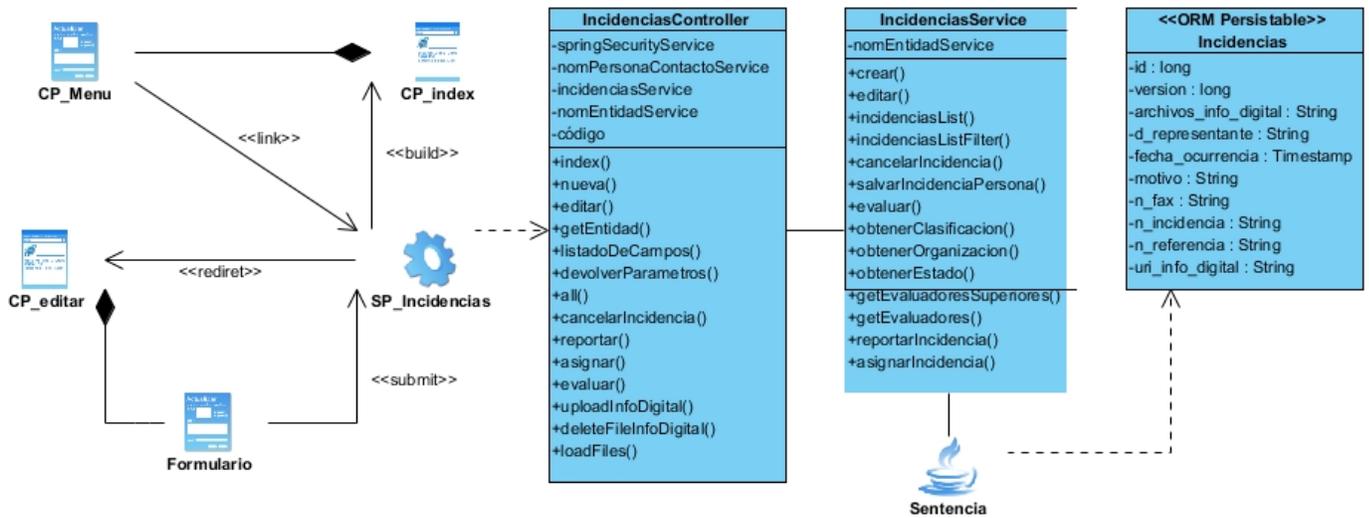


Fig. 22 Diagrama de clases del diseño Editar Incidencia

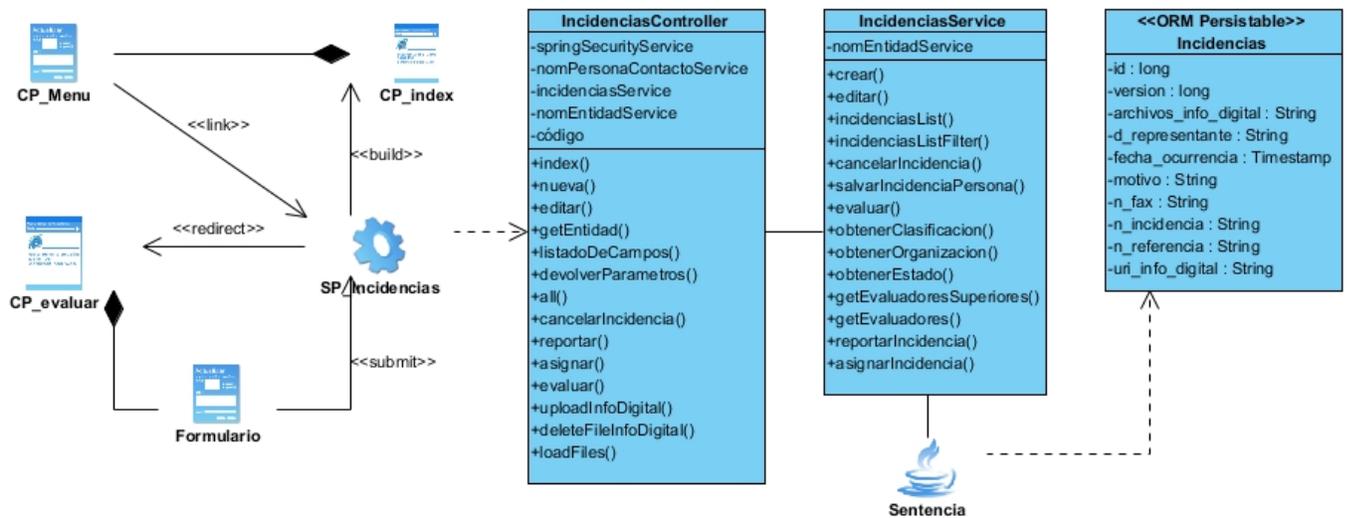


Fig. 23 Diagrama de clases del diseño Evaluar Incidencia

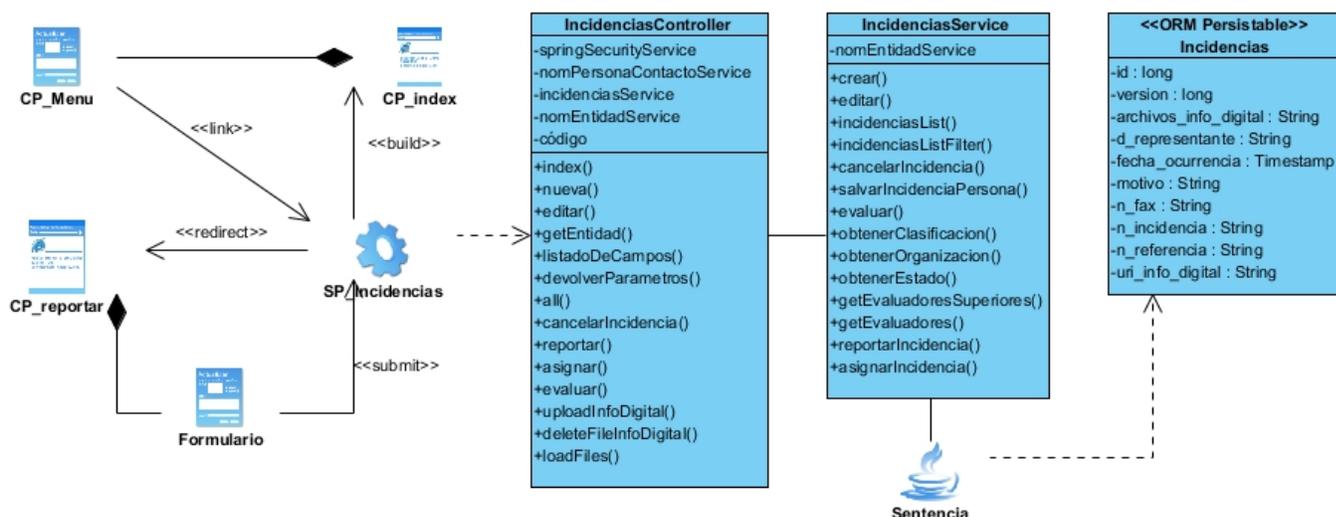


Fig. 24 Diagrama de clases del diseño Reportar Incidencia

Anexo 3: Clasificación de las Incidencias

Queja: Expresión de insatisfacción manifestada al CECMED con respecto a sus decisiones, servicios o al propio proceso de tratamiento de las quejas, donde se espera una respuesta o solución explícita o implícita. (44)

Reclamación: Expresión de insatisfacción manifestada al CECMED con respecto a sus decisiones, servicios o procedimientos, derivadas del incumplimiento de los compromisos adquiridos conforme los requisitos establecidos y vigentes, en la que se solicita o pretende algún tipo de reparación o corrección.(44)

Reconsideración: Proceso administrativo formal para la solución de discrepancias, en el que se solicita al CECMED considerar nuevamente una decisión tomada en busca de producir su cambio, presentada por el regulado al que fue dirigida. (45)

Anexo 4: Descripción de las variables

No	Nombre de campo	Clasificación	Puede ser nulo	Descripción
1	Clasificación	Campo de selección.	No	Permite seleccionar el tipo de incidencia a registrar.

2	Persona de Contacto	Campo de selección simple.	No	Toma el nombre y apellidos a partir del usuario autenticado en el sistema.
3	Organización	Campo de selección.	No	Permite seleccionar la Entidad a la que pertenece.
4	Dirección	Campo de texto	No	Se llena automáticamente a partir de la organización seleccionada.
5	Número de Teléfono	Campo de números enteros.	Si	Se llena automáticamente a partir de la organización seleccionada, aceptando solo números.
6	Número de Fax	Campo de números enteros.	Si	Se llena automáticamente a partir de la organización seleccionada, aceptando solo números.
7	Correo electrónico	Campo de texto.	Si	Se llena automáticamente a partir de la organización seleccionada.
8	Fecha Ocurrencia	Campo de texto. (Date)	No	Permite seleccionar la fecha en la que ocurrió la incidencia.
9	Motivo de la Incidencia	Campo de texto.	No	Permite describir brevemente el motivo de la incidencia.

Anexo 5: Descripción ampliada de los CUN y los CUS

Tabla 8 Descripción textual del CUS Reportar Incidencias.

Caso de Uso:	Reportar Incidencia.
Actor(s):	Especialista Calidad y Especialista Superior Calidad
Resumen:	El caso de uso se inicia cuando los usuarios con rol Especialista Superior Calidad o Especialista Calidad acceden al módulo de las Incidencias para Reportar a los usuarios con rol Evaluador Superior las incidencias cuyo Estado sea Nueva.
Precondiciones:	El usuario tiene que estar autenticado.
Referencias:	RF 10
Prioridad:	Alta
Flujo Normal de Eventos	
Acción del Actor	Respuesta del Sistema
1. El usuario accede al Módulo de Incidencias.	<p>2. Muestra en la pantalla inicial un componente de tipo tabla con las Incidencias existentes en el sistema y los datos más importantes de cada una: Código de la Incidencia, Clasificación Incidencia, Organización, Fecha de Ocurrencia, Fecha de Emisión y Estado. Además se muestran las siguientes acciones:</p> <p>a) Registrar Incidencia: esta acción solo podrá ser realizada por el cliente, el especialista de calidad, el especialista superior de calidad.</p> <p>b) Editar Incidencia: esta acción solo puede ser ejecutada por los usuarios con rol Especialista Calidad y Especialista Superior Calidad. Para los usuarios con rol cliente solo podrá editar solo las incidencias registradas por él.</p> <p>c) Cancelar Incidencia: las incidencias podrán ser canceladas por los usuarios con rol Cliente, Especialista Calidad, Especialista Superior Calidad.</p> <p>d) Buscar Incidencia: la búsqueda podrá ser realizada por los usuarios con rol Cliente, Especialista Calidad, Especialista Superior Calidad por el parámetro que desee, en función de los datos captados de las Incidencias, en el caso de la Fecha</p>

	<p>se podrá realizar para un día específico o en un rango de fecha en función de la Fecha de Creación.</p> <p>e) Evaluar Incidencia: se corresponde con la evaluación de una incidencia por parte del usuario con el rol evaluador que se hará cargo de la investigación.</p> <p>f) Reportar Incidencia: se corresponde con la asignación de una incidencia por parte del usuario con el rol Especialista Calidad o Especialista Superior Calidad al usuario con el rol Evaluador Superior que su cargo sea Jefe de Área.</p>
3. Ejecuta la acción Buscar Incidencia.	
4. Selecciona en la pantalla inicial la Incidencia que desea Reportar.	
5. Selecciona la opción Reportar Incidencia.	6. Muestra una ventana modal con los datos Reportado a y Fecha de Reporte.
7. Selecciona el especialista a quien le va a asignar la Incidencia, la Fecha del Reporte y la opción Aceptar.	8. Verifica que los campos obligatorios se encuentren llenos y valida que respeten el formato definido para cada uno de ellos. Ver RF 10
	9. Guarda en BD los datos introducidos: Reportado a y Fecha de Reporte.
	10. Muestra en la pantalla inicial del componente, la Incidencia cuyo estado era Nueva se mostrará con estado Asignada.

Tabla 9 Descripción textual del CUS Evaluar Incidencias

Caso de Uso:	Evaluar Incidencia.
Actor(s):	Evaluador
Resumen:	El caso de uso se inicia cuando el usuario con rol Evaluador accede al módulo de Incidencias donde en la pantalla principal podrá visualizar las Incidencias reportadas a él para su evaluación.
Precondiciones:	El usuario tiene que estar autenticado.
Referencias:	RF 11

Prioridad:	Alta
Flujo Normal de Eventos	
Acción del Actor	Respuesta del Sistema
1. Accede al Módulo de Incidencias.	<p>2. Muestra en la pantalla inicial un componente de tipo tabla con las Incidencias asignadas a él y los datos más importantes de cada una: Código de la Incidencia, Clasificación Incidencia, Organización, Fecha de Ocurrencia, Fecha de Emisión y Estado. Además se muestran las siguientes acciones:</p> <p>a) Registrar Incidencia: esta acción solo podrá ser realizada por el Cliente, el Especialista Calidad y el Especialista Superior Calidad.</p> <p>b) Editar Incidencia: esta acción puede ser ejecutada por los usuarios con el rol Cliente, Especialista Calidad y Especialista Superior Calidad.</p> <p>c) Cancelar Incidencia: las incidencias podrán ser canceladas por los usuarios con rol Cliente, Especialista Calidad y Especialista Superior Calidad.</p> <p>d) Buscar Incidencia: la búsqueda podrá ser realizada por los usuarios con rol Cliente, Especialista Calidad y Especialista Superior Calidad por el parámetro que desee, en función de los datos captados de las Incidencias, en el caso de la Fecha se podrá realizar para un día específico o en un rango de días en función de la fecha de creación.</p> <p>e) Evaluar Incidencia: se corresponde con la evaluación de una incidencia por parte del usuario con el rol evaluador que se hará cargo de la investigación.</p> <p>f) Reportar Incidencia: se corresponde con la asignación de una incidencia por parte de los usuarios con rol Especialista Calidad o Especialista Superior Calidad al usuario con el rol Evaluador</p>

	Superior que su cargo sea Jefe de Área.
3. Ejecuta la acción Buscar Incidencia.	
4. Selecciona la Incidencia deseada. 4.1 Selecciona la opción Evaluar Incidencia.	5. Muestra un formulario con los datos de la Incidencia además de un campo de texto libre para argumentar la evaluación de la Incidencia.
6. Revisa los datos de la Incidencia escogida. 6.1 Argumenta su evaluación y selecciona la opción Aceptar.	7. Verifica que los campos obligatorios se encuentren llenos y valida que respeten el formato definido para cada uno de ellos. Ver RF 11
	8. Guarda en BD los datos introducidos.
	9. Muestra en la pantalla inicial del componente, la Incidencia cuyo estado era Nueva se mostrará con Estado Resuelta.
Flujo Alternativo 1: El usuario introduce datos en el sistema que no respetan el formato definido para cada uno de ellos.	
	7. Muestra un mensaje de error: "Se han encontrado errores en el formulario, por favor rectifíquelos".
	8. El sistema marca los campos en los que hay errores e indica como corregirlos. Obligatorio: El campo es obligatorio. Continúa en el paso 6 del flujo normal de eventos.

Tabla 10 Descripción textual del CUS Enviar Notificaciones.

Caso de Uso:	Enviar Notificaciones.
Actor(s):	Especialista
Resumen:	Enviar notificaciones de correo electrónico una vez se emita el dictamen o evaluación de la incidencia.
Precondiciones:	El usuario tiene que estar autenticado.
Referencias:	RF 12
Prioridad:	Alta
Flujo Normal de Eventos	
Acción del Actor	Respuesta del Sistema
	1. Una vez que se emita un dictamen se conecta al servidor de correo y envía un correo notificando al cliente que registró la incidencia el dictamen emitido.

Tabla 11 Descripción textual del CUS Mostrar Gráfica del Total de Incidencias VS Clasificación por año actual.

Caso de Uso:	Mostrar Gráfica del Total de Incidencias VS Clasificación por año actual.
---------------------	---

Actor(s):	Especialista
Resumen:	Cuando los usuarios con rol Especialista acceden al módulo Incidencias se le mostrará un gráfico de barra que muestre en el período del año actual el Total de Incidencias vs Clasificación.
Precondiciones:	El usuario tiene que estar autenticado.
Referencias:	RF 13
Prioridad:	Media
Flujo Normal de Eventos	
Acción del Actor	Respuesta del Sistema
1. Accede al Módulo de Incidencias.	2. Muestra gráfico de barra que refleja el período del año actual con el total de las Incidencias vs total por clasificación de las Incidencias: Quejas, Reclamaciones y Reconsideraciones.

Tabla 12 Descripción textual del CUS Mostrar Gráfica del Total de Incidencias por Clasificación vs Estado.

Caso de Uso:	Mostrar Gráfica de Total de Incidencias por Clasificación vs Estado	
Actor(s):	Especialista	
Resumen:	Cuando los usuarios con rol Especialista acceden al módulo de las Incidencias se le mostrará un gráfico de barra que muestre en el período del año actual el Total de Incidencias vs Clasificación.	
Precondiciones:	El usuario tiene que estar autenticado.	
Referencias:	RF 14	
Prioridad:	Alta	
Flujo Normal de Eventos		
Acción del Actor	Respuesta del Sistema	
1. Accede al Módulo de Incidencias.	2. Muestra gráfico de barra con el período del año actual con el Total de Incidencias, su Clasificación vs Estado: Nueva, Reportado, En evaluación, Procede, No Procede.	

Anexo 6: Diseño de Casos de Prueba

Tabla 13 Caso de prueba Cancelar Incidencia.

Nombre del Caso de Prueba	Cancelar Incidencia	EC: 1.1
Entrada	Se selecciona la incidencia con estado Terminada.	
Resultado Esperado	Se cancela la Incidencia.	

Salida	Se visualiza en la interfaz principal la incidencia con estado Cancelada.	
Flujo Central	<ol style="list-style-type: none"> 1. Se selecciona la Incidencia con Estado Terminada. 2. Se selecciona la opción Cancelar Incidencia. 3. Se muestra un mensaje para confirmar la cancelación. 3. Se selecciona la opción Aceptar. 4. Se cancela la Incidencia satisfactoriamente. 	
Nombre del Caso de Prueba	Cancelar Incidencia	EC: 1.2
Entrada	No se selecciona la incidencia.	
Resultado Esperado	Se cancela la Incidencia.	
Salida	Se visualiza en la interfaz principal el mensaje “Debe seleccionar una Incidencia para realizar esta acción.”	
Observaciones	<ol style="list-style-type: none"> 1. Se selecciona la opción Cancelar Incidencia. 2. Se muestra un mensaje en la pantalla principal: “Debe seleccionar una Incidencia para realizar esta acción.” 	

Tabla 14 Caso de prueba Buscar Incidencia.

Nombre del Caso de Prueba	Buscar Incidencia	EC: 1.1
Entrada	Se selecciona la opción Buscar Incidencia.	
Resultado Esperado	Se muestra los datos de la Incidencia deseada.	
Salida	Se visualiza en la interfaz principal la incidencia deseada.	
Flujo Central	<ol style="list-style-type: none"> 1. Se selecciona la opción Buscar Incidencia. 2. Se muestra una ventana modal con las opciones de los datos para realizar la búsqueda por los criterios establecidos. 3. Se seleccionan los tipos de datos para realizar la búsqueda. 3. Se selecciona la opción Aceptar. 4. Se muestra la Incidencia que coincide con los datos seleccionados. 	
Nombre del Caso de Prueba	Campos vacíos en la búsqueda	EC: 1.2
Entrada	Se dejan los campos vacíos.	
Resultado Esperado	Se muestre la Incidencia en la pantalla principal.	
Salida	Se visualiza en la interfaz principal el mensaje “Debe seleccionar los datos para realizar esta acción.”	
Flujo Central	<ol style="list-style-type: none"> 1. Se selecciona la opción Buscar Incidencia. 2. No selecciona los criterios de búsqueda. 3. Se selecciona la opción Aceptar. 4. Muestra en pantalla el mensaje “Debe seleccionar los datos para realizar esta acción.” 	

Tabla 15 Caso de prueba Editar Incidencia.

Nombre del Caso de Prueba	Editar Incidencia	EC: 1.1
Entrada	Se muestran todos los datos de la incidencia a excepción de la persona de contacto, la organización, la dirección los restantes datos.	
Resultado Esperado	Guarde los cambios realizados	

Salida	Se visualiza en la interfaz principal la incidencia modificada.
Observaciones	<ol style="list-style-type: none">1. Se selecciona la incidencia con Estado Nueva.2. Se seleccionan y editan los datos a modificar.3. Se selecciona la opción Aceptar.4. Se muestra un mensaje "La incidencia fue modificada satisfactoriamente".

GLOSARIO DE TÉRMINOS

CECMED: Centro para el Control Estatal de Medicamentos Equipos y Dispositivos Médicos.

MINSAP: Ministerio de Salud Pública.

ISO 9000: Gerenciamiento de Calidad y Normas de Aseguramiento de la Calidad –Lineamientos para la selección y uso, explica los conceptos fundamentales en calidad; define términos claves; y provee una guía para la selección, uso y adecuación a las ISO 9001, 9002, y 9003.

Jefe DGC: Jefe del Departamento de Gestión de la Calidad.

Especialista DGC: Especialista del Departamento de Gestión de la Calidad.

DGC: Departamento de Gestión de la Calidad.

CU: Caso de Uso.

CUN: Caso de Uso del Negocio.

CUS: Caso de Uso del Sistema.

RUP: Rational Unified Process. El Proceso Unificado es un proceso de desarrollo de software (Conjunto de actividades necesarias para transformar los requisitos de un usuario en un sistema de software).

UML: Lenguaje de Modelado Unificado (Unified Modeling Language).

MVC: Modelo Vista Controlador.

RN: Regla del negocio

ISO: Internacional Standard Organization.

DER: Diagrama Entidad Relación.

UCI: Universidad de Ciencias Informáticas.