



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

Herramienta para la gestión de la información de los procesos medioambientales en la Oficina Reguladora Ambiental y de Seguridad Nuclear.



Autores:


Gisela Sagarra Mendoza

Edry Garcia Ramos

Tutor:

Ing. Arcadio Abad Márquez

La Habana, junio 2014
“Año 56 de la Revolución”



“Es que, cuando los hombres llevan en la mente un mismo ideal, nada puede incomunicarlos, ni las paredes de una cárcel, ni la tierra de los cementerios, porque un mismo recuerdo, una misma alma, una misma idea, una misma conciencia y dignidad los alienta a todos.”

Fidel Castro

13 de marzo de 1991

Declaración de autoría

DECLARACIÓN DE AUTORÍA:

Por este medio declaramos que somos los únicos autores de este trabajo y autorizamos a la Universidad de las Ciencias Informáticas (UCI) para que hagan el uso que estimen pertinente con este trabajo.

Para que así conste firmamos la presente a los ____ días del mes de _____ del 2014.

Firma del Autor

Gisela Sagarra Mendoza

Firma del Autor

Edry Garcia Ramos

Firma del Tutor

Ing. Arcadio Abad Márquez



Agradecimientos

A mamá, a tía Mary y a Edry.

A la familia en general que se preocupa porque seamos mejor cada día.

A Arcadio por acogernos en su casa hasta altas horas de la noche.

A los compañeros de la universidad por los ratos buenos que pasamos.

En general a todo aquel que hizo posible que hoy estuviera aquí.

Gisela.



Agradecimientos

A mis padres por guiarme en la vida, a Gisela por ser Gisela, a la familia por hacerme mejor persona cada día, a los amigos que he hecho en la universidad, autores de buenos momentos, al tutor, Arcadio, que siempre estuvo en los momentos que lo necesitábamos, en general a todo el que estuvo cerca.

Edry



Dedicatoria

A mamá y a tía Mary por velar por mí siempre.

Gisela.

A mis padres por apoyarme en todo, o casi todo.

Edry.



Resumen

La Oficina de Regulación Ambiental y Seguridad Nuclear (ORASEN) es creada como órgano regulador adscrito al Ministerio de Ciencia, Tecnología y Medio Ambiente (CITMA), esta oficina recibe información de cuatro centros con sucursales en todas las provincias del país, este flujo de información se realiza mayoritariamente por correo electrónico, y se guarda tanto en formato digital como en papel, se genera gran volumen de documentación lo que dificulta cualquier consulta o reporte y se corre el riesgo de pérdida de información, por esto se decide implementar un sistema de gestión de información que contribuya al control y seguimiento de los procesos medioambientales que realiza la ORASEN. Se realizó un estudio para seleccionar las herramientas informáticas más viables, teniendo en cuenta la tendencia del desarrollo de aplicaciones para la gestión de información, además se realiza un análisis de sistemas informáticos similares con el objetivo de identificar posibles funcionalidades de las que se pudiera hacer uso. Para desarrollar la herramienta informática deseada se hace uso de las fases de la metodología seleccionada. Para confirmar que el sistema tenga el funcionamiento correcto se realizarán pruebas de aceptación y pruebas unitarias. Se obtuvo un producto en total funcionamiento y un documento presto a ser base para cualquier investigación en el futuro, así como para posibles cambios que se quieran hacer a la solución. Como resultado de la investigación se deja constancia documental de la metodología empleada la cual permitió cumplir con los objetivos propuestos así como cumplir con las necesidades presentadas por el cliente.

Palabras claves: autorización, conciliación, desechos peligrosos, gestión de información, inspección, medidas, medioambiente, sucesos extraordinarios.



Índice

Índice	III
Índice de Tablas	VI
Índice de Figuras	X
Introducción.....	1
Capítulo 1 Fundamentación teórica	6
1.1 Marco Conceptual	6
Gestión de la información ambiental	6
Regulación Ambiental	7
Contravenciones	7
1.2 Soluciones similares.....	7
Gestión ambiental corporativa.....	7
Sistema para el seguimiento y mejora de la gestión ambiental y energética	8
Sistema de información ambiental	9
Aporte del estudio de las aplicaciones a la propuesta de solución	10
1.3 Lenguajes de programación	11
Lenguajes de programación del lado del servidor	11
ASP 3.0	11
PHP 5.3	12
Lenguajes de programación del lado del cliente	12
JavaScript 1.5.....	12
CSS3	13
HTML5.....	13
1.4 Frameworks de desarrollo web	13
Frameworks del lado del servidor	13
CodeIgniter 2.1.4	13
Symfony 2.3.....	14
Propel ORM2.0	15
Doctrine ORM 2.4.2	15
<i>Frameworks</i> del lado del cliente:.....	16
JQuery 2.1.0	16
Ext JS 4.2	16
1.5 Metodologías de desarrollo de software	17



Proceso Unificado de Desarrollo de Software (RUP)	17
Extreme Programming (XP)	18
1.6 Herramientas CASE	19
Rational Rose 7.0	19
Visual Paradigm 8.0	19
1.7 Sistemas gestores de base de datos:	20
MySQL Database Server 5.5.9	20
PostgreSQL 9.1	20
1.8 Entorno de desarrollo integrado (IDE)	21
Zend Studio 10.0.0	21
NetBeans IDE 7.3	21
1.9 Servidores web	22
Internet Information Server (IIS) 8.5	22
Apache 2.2	22
1.10 Herramientas para la edición de imágenes	23
Adobe Photoshop CS3	23
GIMP 2.8	23
Conclusiones	24
Capítulo 2 Descripción de la solución propuesta	25
2.1. Aspectos no funcionales del sistema	25
2.2. Planificación del sistema	27
Historias de usuario	28
Estimación de esfuerzos por HU	30
Plan de iteraciones	31
2.3. Diseño del sistema	34
Tarjetas Clases Responsabilidad Colaborador CRC	35
Patrones de diseño	37
Patrones arquitectónicos	39
Modelo Vista Controlador (MVC)	39
Prototipo de interfaz de usuario	40
Diagrama de Base de Datos, Modelo Entidad- Relación	43
Conclusiones	45
Capítulo 3 Implementación y prueba del sistema	47
3.1. Implementación del sistema	47



Índice

Tareas de programación	47
Estándares de codificación	51
Módulos de la solución propuesta.....	51
3.2. Pruebas al sistema.....	54
Pruebas de aceptación	¡Error! Marcador no definido.
Pruebas unitarias.....	58
Conclusiones parciales	59
Conclusiones generales	61
Recomendaciones.....	63
Bibliografías	64
Anexo 1 Historias de usuario.....	69
Anexo 2 Tareas de programación	74
Anexo 3 Tarjetas CRC.....	85
Anexo 4 Caso de prueba.....	91
Anexo 5 Entrevista	99



Tabla 1 Historia de Usuario: Generar Reportes.....	29
Tabla 2 Historia de Usuario Gestionar medidas.....	29
Tabla 3 Historia de usuario Gestionar autorización.....	30
Tabla 4 Historia de usuario Gestionar suceso extraordinario.....	30
Tabla 5 Estimación de esfuerzos por Historia de Usuario.....	31
Tabla 6 Plan de duración de las iteraciones.....	32
Tabla 7 Plan de entrega.....	33
Tabla 8 Tarjeta CRC SIGA.portal.Inspecciones.View.....	35
Tabla 9 Tarjeta CRC SIGA.portal.Inspecciones.....	36
Tabla 10 Tarjeta CRC SIGA.portal.Autorizaciones.View.....	36
Tabla 11 Tarjeta CRC SIGA.portal.Autorizaciones.....	36
Tabla 12 Descripción de las tablas de la base de datos.....	45
Tabla 13 Tareas de programación por Historias de usuario.....	47
Tabla 14 Tarea de programación 1 Generar reporte para la Historia de usuario Generar reporte	48
Tabla 15 Tarea de programación 1 Crear conciliación para la Historia de usuario Gestionar Conciliación.....	49
Tabla 16 Tarea de programación 2 Modificar conciliación para la Historia de usuario Gestionar conciliación.....	50
Tabla 17 Tarea de programación 3: Eliminar conciliación para la Historia de usuario Gestionar conciliación.....	50
Tabla 18 Caso de Prueba de Aceptación Generar reporte.....	55
Tabla 19 Caso de Prueba de Aceptación Adicionar autorización.....	55
Tabla 20 Caso de Prueba de Aceptación Modificar autorización.....	56
Tabla 21 Caso de Prueba de Aceptación Eliminar autorización.....	56
Tabla 22 Historia de usuario Gestionar usuario.....	69
Tabla 23 Historia de usuario Gestionar Inspección.....	69
Tabla 24 Historia de usuario Gestionar desechos.....	69
Tabla 25 Historia de usuario Gestionar conciliaciones.....	70
Tabla 26 Historia de usuario Autenticar usuario.....	70
Tabla 27 Historia de usuario Mostrar tipo de autorización.....	71
Tabla 28 Historia de usuario Mostrar tipo de inspección.....	71



Tabla 29 Historia de usuario Mostrar estado de la inspección	71
Tabla 30 Historia de usuario Mostrar resultado de las medidas.....	72
Tabla 31 Historia de usuario Mostrar normativa de la medida	72
Tabla 32 Historia de usuario Mostrar categoría de desechos	73
Tabla 33 Historia de usuario Mostrar tipo de suceso extraordinario	73
Tabla 34 Historia de usuario Generar gráficos	74
Tabla 35 Tarea de programación 1 Crear usuario para la Historia de usuario Gestionar usuario.	74
Tabla 36 Tarea de programación 2 Modificar usuario para la Historia de usuario Gestionar usuario.	74
Tabla 37 Tarea de programación 3 Eliminar usuario para la Historia de usuario Gestionar usuario.	75
Tabla 38 Tarea de programación 1 Autenticar usuario para la Historia de usuario Autenticar usuario	75
Tabla 39 Tarea de programación 1 Crear autorización para la Historia de usuario Gestionar autorización.....	76
Tabla 40 Tarea de programación 2 Modificar autorización para la Historia de usuario Gestionar autorización.....	76
Tabla 41 Tarea de programación 3 Eliminar autorización para la Historia de usuario Gestionar autorización.....	76
Tabla 42 Tarea de programación 1 Mostar tipo de autorización para la Historia de usuario Mostar tipo de autorización	77
Tabla 43 Tarea de programación 1 Crear inspección para la Historia de usuario Gestionar inspección	77
Tabla 44 Tarea de programación 2 Modificar inspección para la Historia de usuario Gestionar inspección	78
Tabla 45 Tarea de programación 3 Eliminar inspección para la Historia de usuario Gestionar inspección	78
Tabla 46 Tarea de programación 1 Mostar tipo de inspección para la Historia de usuario Mostar tipo de inspección.....	78
Tabla 47 Tarea de programación 1 Mostar estado de la inspección para la Historia de usuario Mostar estado de la inspección	79
Tabla 48 Tarea de programación 1 Crear desechos para la Historia de usuario Gestionar desechos.....	79



Tabla 49 Tarea de programación 2 Modificar desechos para la Historia de usuario Gestionar desechos.....	80
Tabla 50 Tarea de programación 3 Eliminar desechos para la Historia de usuario Gestionar desechos.....	80
Tabla 51 Tarea de programación 1 Mostar categoría de desechos para la Historia de usuario Mostar categoría de desechos	80
Tabla 52 Tarea de programación 1 Crear medida para la Historia de usuario Gestionar medida	81
Tabla 53 Tarea de programación 2 Modificar medida para la Historia de usuario Gestionar medida	81
Tabla 54 Tarea de programación 3 Eliminar medida para la Historia de usuario Gestionar medida	82
Tabla 55 Tarea de programación 1 Mostar resultado de la medida para la Historia de usuario Mostar resultado de la medida	82
Tabla 56 Tarea de programación 1 Mostar normativa de la medida para la Historia de usuario Mostar normativa de la medida	82
Tabla 57 Tarea de programación 1 Crear suceso extraordinario para la Historia de usuario Gestionar suceso extraordinario.....	83
Tabla 58 Tarea de programación 2 Modificar suceso extraordinario para la Historia de usuario Gestionar suceso extraordinario.....	83
Tabla 59 Tarea de programación 3 Eliminar suceso extraordinario para la Historia de usuario Gestionar suceso extraordinario.....	84
Tabla 60 Tarea de programación 1 Mostar tipo de suceso extraordinario para la Historia de usuario Mostar tipo de suceso extraordinario	84
Tabla 61 Tarea de programación 1 Generar gráficos para la Historia de usuario Generar gráficos	84
Tabla 62 Tarjeta CRC SIGA.portal.Escritorio.....	85
Tabla 63 Tarjeta CRC SIGA.componets.WinPersonas	85
Tabla 64 Tarjeta CRC SIGA.Lib.Util	85
Tabla 65 Tarjeta CRC SIGA.Lib.Base64	86
Tabla 66 Tarjeta CRC Portal.acceso.Login.....	86
Tabla 67 Tarjeta CRC Portal.acceso.Login.View	86
Tabla 68 Tarjeta CRC SIGA.portal.Usuario	86
Tabla 69 Tarjeta CRC SIGA.portal.Usuario.View.....	87



Índice

Tabla 70 Tarjeta CRC SIGA.portal. Sucesos.....	87
Tabla 71 Tarjeta CRC SIGA.portal.Sucesos.View	87
Tabla 72 Tarjeta CRC SIGA.portal.Conciliaciones.....	88
Tabla 73 Tarjeta CRC SIGA.portal. Conciliaciones.View	88
Tabla 74 Tareas CRC SIGA.portal.Desechos.....	88
Tabla 75 Tarjeta CRC SIGA.portal.Desechos.View	89
Tabla 76 Tarjeta CRC SIGA.portal.Medidas	89
Tabla 77 Tarjeta CRC SIGA.portal.Medidas.View	89
Tabla 78 Tarjeta CRC SIGA.portal.Autorizaciones	90
Tabla 79 Tarjeta CRC SIGA.portal.Autorizaciones.View.....	90
Tabla 80 Tarjeta CRC SIGA.portal.Escritorio.....	91
Tabla 81 Caso de Prueba de Aceptación Adicionar inspección.....	91
Tabla 82 Caso de Prueba de Aceptación Modificar inspección.	91
Tabla 83 Caso de Prueba de Aceptación Eliminar inspección.....	92
Tabla 84 Caso de Prueba de Aceptación Adicionar medida.....	92
Tabla 85 Caso de Prueba de Aceptación Modificar medida.	93
Tabla 86 Caso de Prueba de Aceptación Eliminar medida	93
Tabla 87 Caso de Prueba de Aceptación Adicionar desecho.	93
Tabla 88 Caso de Prueba de Aceptación Modificar desechos.....	94
Tabla 89 Caso de Prueba de Aceptación Eliminar desecho.	94
Tabla 90 Caso de Prueba de Aceptación Adicionar suceso extraordinario.....	95
Tabla 91 Caso de Prueba de Aceptación Modificar suceso extraordinario.	95
Tabla 92 Caso de Prueba de Aceptación Eliminar suceso extraordinario.....	96
Tabla 93 Caso de Prueba de Aceptación Adicionar conciliaciones.....	96
Tabla 94 Caso de Prueba de Aceptación Modificar conciliación.....	96
Tabla 95 Caso de Prueba de Aceptación Eliminar conciliación.....	97
Tabla 96 Caso de Prueba de Aceptación Adicionar usuario.....	97
Tabla 97 Caso de Prueba de Aceptación Modificar usuario.	98
Tabla 98 Caso de Prueba de Aceptación Eliminar usuario.....	98
Tabla 99 Caso de Prueba de Aceptación Autenticar usuario.....	98
Tabla 100 Caso de Prueba de Aceptación Mostar tipo de autorización.....	99



Índice de Figuras

Figura 1 Sistema de Gestión Ambiental Corporativa ECOSYSTEM.	8
Figura 2 Sistema para el seguimiento y mejora de la gestión ambiental y energética.	9
Figura 3 Sistema de Información Ambiental.	10
Figura 4 Definición de Roles.....	28
Figura 5 Diagrama del patrón de Arquitectura de Software Modelo Vista Controlador (MVC). ..	40
Figura 6 Prototipo de Interfaz de usuario de la página principal.	41
Figura 7 Prototipo de interfaz de usuario Gráfico.....	41
Figura 8 Prototipo de interfaz de usuario Adicionar Autorización.....	42
Figura 9 Prototipo de interfaz de usuario Insertar Autorización.....	42
Figura 10 Módulo administración.....	52
Figura 11 Módulo procesos	53
Figura 12 Módulo Análisis de datos.....	54
Figura 13 Módulo Análisis de datos1	54
Figura 14 Prueba unitaria 1	59
Figura 15 Prueba unitaria 2	59



Introducción

Desde su surgimiento, las Tecnologías de la Informática y las Comunicaciones (TIC) han estado en constante evolución, el siglo XXI ha sido el momento en que han alcanzado elevados niveles de desarrollo producto a la estrecha relación de estas con la enmarcada necesidad de satisfacer al hombre en los más diversos ámbitos de su vida. Las TIC son un medio para apoyar y potenciar las capacidades de almacenamiento, de procesamiento de la información y comunicación.

Desde el surgimiento del hombre, este siempre ha estado en constante interacción (o intercambio) con la naturaleza para satisfacer sus necesidades cada vez más crecientes. El Compañero Fidel Castro Ruz, el 1ro de enero de 1984 en Santiago de Cuba, hace alusión a los problemas medioambientales y de desarrollo sostenible alertando a la humanidad: *“Pasan rápido 20 años y (...) estamos conscientes de los problemas que esperan al mundo en las décadas futuras, en todos los sentidos: la población creciente, que alcanzará más de 6 000 millones de habitantes; los problemas de alimentación, del agua, de los recursos naturales, la contaminación, los problemas del subdesarrollo que se proyectan hacia los años futuros para una gran parte de la humanidad (...)”*.(1)

Actualmente el ser humano, junto con sus crecientes necesidades y el acelerado desarrollo tecnológico han acarreado disímiles problemas al medioambiente, entiéndase por medioambiente: *“(...) sistema complejo y dinámico de interrelaciones ecológicas, socioeconómicas y culturales, que evoluciona a través del proceso histórico de la sociedad, (...) abarca la naturaleza, la sociedad, lo creado por la humanidad: la cultura, la propia humanidad, y como elemento esencial las relaciones entre todos sus componentes de forma compleja (...)”* (2). Entre los principales problemas relacionados hoy con el medioambiente se pueden identificar: degradación de los suelos, contaminación del medio ambiente, agotamiento de la capa de ozono, cambio climático y pérdida de la diversidad biológica.

En la búsqueda para contrarrestar los disímiles problemas ambientales, surge el término desarrollo sostenible, entiéndase por desarrollo sostenible: *“desarrollo que debe responder a las necesidades del presente sin comprometer la capacidad de las generaciones futuras.”*(3) y sus tres objetivos fundamentales: continuar produciendo riquezas para satisfacer las necesidades de la población mundial (pilar económico), velar por la reducción de las desigualdades entre los pueblos del mundo (pilar social), no degradar al medio ambiente que heredarán las



Introducción

generaciones futuras (pilar medioambiental)(3). Estas son las razones que llevan a crear varias entidades gubernamentales, encargadas de controlar los temas relacionados con el medioambiente.

El Ministerio de Ciencia Tecnología y Medio Ambiente (CITMA), creado en el año 1994, es el organismo encargado de dirigir, ejecutar y controlar la política del Estado y del Gobierno en la actividad científica y tecnológica, la política ambiental y de uso pacífico de la energía nuclear, propiciando su integración coherente para contribuir al desarrollo sostenible del país.(4)

Existen diferentes organismos vinculados a este ministerio que atienden de manera más eficiente sectores del medio ambiente como es la Oficina Reguladora Ambiental y de Seguridad Nuclear (ORASEN), la que tiene como objetivos principales:

- Organizar y gerenciar el proceso de concesión de autorizaciones, permisos y licencias en las esferas reguladoras de control ambiental, la seguridad biológica, la seguridad nuclear y radiológica y el control de armas químicas, en correspondencia con la legislación vigente.
- Otorgar, modificar, renovar y suspender o revocar autorizaciones, permisos y licencias en las esferas de su competencia, en moneda nacional.
- Inscribir y registrar las entidades acreditadas para la realización de estudios y consultorías en las esferas de su competencia, en moneda nacional.
- Realizar eventos y otras actividades científicas técnicas en las esferas de su competencia, en moneda nacional.
- Brindar servicio científico técnicos de soluciones tecnológicas en instalaciones de alto riesgo, en moneda nacional y divisa al costo.
- Dirigir y ejecutar programas y proyectos científico técnico en las esferas de su competencia, en moneda nacional.

La entidad mencionada concentra información, a nivel nacional, referente a los procesos realizados por el Centro Nacional de Seguridad Biológica (CNSB), el Centro Ejecutivo de la Autoridad Nacional para la Convención de Armas Químicas (CEANPAQ), el Centro Nacional de Seguridad Nuclear (CNSN) y el Centro de Inspección y Control Ambiental (CICA), todos pertenecientes a la ORASEN. Este intercambio de información se realiza mediante tablas Excel, principalmente a través del correo electrónico y luego para su registro y control se emplean fundamentalmente archivos y carpetas en formato duro, generando así gran volumen de documentación, haciendo muy engorrosa y compleja para los expertos cualquier búsqueda o



Introducción

consulta de datos para su estudio. El gobierno constantemente solicita distintos informes que, por la importancia de los mismos, necesitan de la exactitud y seriedad de la información, en ocasiones no llegan a cumplir con estos requisitos, por la forma en que se almacena la información en la oficina.

Por una parte la ORASEN posee personal capacitado y herramientas informáticas que pudiera disponer para facilitar los diferentes procesos que se realizan, sin embargo no emplean las potencialidades de las TIC puestas a su disposición, de manera efectiva, para lograr este objetivo. La situación descrita anteriormente nos conduce a plantear el siguiente **problema de investigación**: ¿Cómo contribuir a la gestión de la información para un mayor control y seguimiento de los procesos medioambientales por los expertos de la ORASEN?

Objetivo general: Desarrollar un sistema informático para la gestión de la información de los procesos medioambientales de la oficina de regulación ambiental y seguridad nuclear, favoreciendo el control y seguimiento de estos por parte de los expertos.

El **objeto de investigación** lo constituyen los sistemas informáticos de gestión de información ambiental.

El **campo de acción** se centra en los sistemas informáticos de gestión de información de los procesos medioambientales de la oficina de regulación ambiental y seguridad nuclear.

Objetivos Específicos:

- Elaborar el marco teórico que sustenta la investigación.
- Implementar las funcionalidades identificadas en los procesos medioambientales a informatizar en la ORASEN para satisfacer las necesidades del cliente.
- Validar mediante pruebas la propuesta de solución desarrollada para comprobar la correcta ejecución de las funcionalidades implementadas.

La **idea a defender** que sustenta a la investigación es que el desarrollo de un sistema basado en los procesos medioambientales de la oficina de regulación ambiental y seguridad nuclear creado con tecnologías libres y capaces de generar reportes con tablas y gráficos contribuirá al control y seguimiento de dichos procesos medioambientales.



El estudio realizado para el desarrollo de esta investigación viene dado a través de los diferentes métodos científicos de investigación.

Los métodos científicos teóricos utilizados son:

- **Método Histórico-Lógico:** el uso de este método favoreció la realización del estado del arte de la ciencia en evolución y el desarrollo de herramientas para la gestión de la información.
- **Método Inductivo-Deductivo:** método utilizado durante la mayor parte de la investigación, ya que se puede definir como una forma de razonamiento por medio de la cual se pasa del conocimiento de casos particulares a generales y viceversa.
- **Método Analítico-Sintético:** este método funciona como sustento de la investigación ya que permite dividir el problema en pequeñas partes para luego buscar relaciones y características generales entre dichos elementos.

Los métodos empíricos utilizados:

- La **entrevista de tipo no estructurada:** permitió la obtención de información de los procesos medioambientales desarrollados por la ORASEN a través de encuentros con la especialista principal Mariela Marrero Garcia, definiéndose las funcionalidades necesarias que debería tener la aplicación.

Tareas de investigación:

- Análisis de los fundamentos teóricos y metodológicos que sustentan la creación de la herramienta para la gestión de la información de los procesos medioambientales que realiza la ORASEN.
- Realización de un estudio de soluciones similares existentes a nivel nacional e internacional así como la investigación sobre las tendencias actuales de las herramientas de programación, las tecnologías y metodologías de desarrollo que se utilizan para llevar a cabo el sistema que se desea implementar y así cumplir con las exigencias planteadas por el cliente.
- Análisis, diseño y creación de la interfaz gráfica de la aplicación.
- Modelado y creación de la base de datos.
- Implementación de los requisitos descritos.
- Creación de los casos de prueba de funcionalidades del sistema.
- Validación de las funcionalidades a partir de pruebas procedentes de la metodología elegida.



El presente trabajo está estructurado en:

Capítulo 1: Fundamentación teórica

Contiene el estudio valorativo realizado sobre sistemas internacionales y nacionales que tienen relación con el problema planteado, así como el estudio de tecnologías, técnicas, tendencias y metodologías para el desarrollo de sistemas de gestión de información además de seleccionar las que serán utilizadas para el desarrollo de la solución propuesta.

Capítulo 2: Descripción de la solución propuesta

En este capítulo se generan diferentes artefactos que propone la metodología seleccionada en estas fases, ejemplo la creación de las Historias de Usuario y su planificación, el diseño de las tarjetas Clases Responsabilidad Colaborador, tarjetas CRC, además se definen los patrones de diseño y arquitectónicos y se modela la Base de Datos.

Capítulo 3: Implementación y prueba

Este capítulo presenta la explicación de cómo se construyó la herramienta, guiados por la metodología de software seleccionada. Para la implementación del sistema quedaron definidas las tareas de programación y los estándares de codificación. Se describen las diferentes pruebas realizadas con el objetivo de validar que la herramienta funcione correctamente.



Capítulo 1. *Fundamentación teórica*

Capítulo 1 Fundamentación teórica

1.1 Marco Conceptual

El marco conceptual se hace necesario para presentar algunos de los conceptos importantes que se abordaron durante la investigación, con el fin de llevar al lector a una mejor comprensión de los temas explicados, además de reutilizar ideas que se han tratado en trabajos similares. Para el desarrollo de esta investigación se hizo énfasis en los conceptos: gestión de la información, regulación ambiental y contravenciones.

Se hará un análisis sobre los principales lenguajes de programación, *frameworks* y metodologías de desarrollo, herramientas CASE, sistemas gestores de bases de datos y entornos de desarrollo integrado, herramientas necesarias que serán escogidas para el desarrollo satisfactorio de la investigación.

Gestión de la información ambiental

Según la bibliografía consultada: queda definido gestión de la información como un “(...)proceso mediatizado por un conjunto de actividades que permiten la obtención de información, lo más pertinente, relevante y económica posible, para ser usada en el desarrollo y el éxito de una organización.” (5)

Otra de las bibliografías consultadas plantea que gestión de la información es: “(...)un conjunto de componentes relacionados que recolectan (o recuperan), procesan, almacenan y distribuyen información para apoyar a la toma de decisiones y el control de una organización.” (6)

Dentro de un sistema de gestión de la información se realizan tres actividades fundamentales:

- Entrada: captura o recolecta datos en bruto tanto del interior de la organización como de su entorno externo.
- Procesamiento: convierte esa entrada de datos en una forma más significativa.
- Salida: transfiere la información procesada a los usuarios o a las actividades para las que se utilizará. (6)

Los autores después de consultar y analizar varias bibliografías llegaron a la conclusión de que un Sistema de Gestión de Información Ambiental es un sistema que recoge diferentes datos sobre el medioambiente, los procesará y permitirá mostrarlos para ser analizados de manera más novedosa, así como permitirá la toma de decisiones con respecto a la protección ambiental, además, de ser necesario, podrá realizar comparaciones.



Capítulo 1. *Fundamentación teórica*

Regulación Ambiental: Después de estudiar varias bibliografías los autores coinciden con que: se percibe regulación ambiental como un proceso mediante el cual se controla el cumplimiento de las legislaciones vigentes, contribuye a que las actividades de producción de bienes y de prestación de servicios se realicen de modo que propenda a la protección del medio ambiente y al logro de las metas del desarrollo sostenible permitiendo prevenir la comisión de contravenciones y delitos.(8)(9)

Contravenciones

El Dr. Manuel Sánchez Zuraty define en su libro Diccionario Básico de Derecho al término contravención como: "(...) *acto contrario a una norma jurídica o mandato*".(10)

En el Diccionario Jurídico Elemental, de Guillermo Caballenas, define a este término como: "(...) *la falta que se comete al no cumplir lo ordenado(...)*".(11)

En otra de las bibliografías consultadas queda definido el término contravención como: "(...) *actos que van en contra de las leyes o lo legalmente establecido y que por lo tanto pueden representar un peligro tanto para quien lo lleva a cabo como también para otros (...)*"(12).

Durante el desarrollo de la investigación se tomará como concepto de contravención a los actos de incumplimientos de licencias ambientales, dificultar o impedir el acceso de los inspectores ambientales estatales a las áreas o lugares a ser inspeccionados, acceder a áreas protegidas sin la debida autorización en los casos en que se requiera, extraer arena de las playas y de sus fuentes de alimentación.

1.2 Soluciones similares

Se realizó un análisis de la documentación y organizaciones existentes que abarcaban temas medioambientales, para identificar las diferentes herramientas de gestión de información, a nivel nacional e internacional, con la intención de analizar la forma en que se maneja la información, cómo se muestran los resultados, cómo se gestionan los roles, etc., para así acumular experiencias y ponerlas en práctica en la solución a proponer.

Gestión ambiental corporativa

La gestión ambiental corporativa (Ecosystem) es una aplicación web que realiza una serie de procesos y tareas, facilitando al departamento colombiano de gestión ambiental, tramitar procesos, definiendo entradas, salidas, impactos ambientales orientándolos hacia el cumplimiento de su política y objetivos ambientales. También presenta la facilidad de realizar



Capítulo 1. Fundamentación teórica

registros, seguimientos de las mediciones de cada uno de sus indicadores ambientales, centralizar la información de forma segura, mantener un seguimiento de las tareas a realizar, generar informes de requisitos aplicables a aspectos ambientales que deben cumplir las organizaciones, además de visualizar y generar reportes que tienen relación con el desempeño ambiental de las empresas los cuales pueden ser exportados a formatos como Excel (xls), pdf y txt. El uso de este software es restringido por leyes de *copyright*.(13)



Figura 1 Sistema de Gestión Ambiental Corporativa ECOSYSTEM.

Accesible en: <http://www.aplicacionesgml.com/Ecosystem/default.aspx>

Sistema para el seguimiento y mejora de la gestión ambiental y energética

El Sistema para el Seguimiento y Mejora de la Gestión Ambiental y Energética (Ekomanager) es una aplicación web que permite la medición, control, automatización, elaboración y administración de la información en el área energética y medioambiental. Esta herramienta permite llevar un seguimiento de la información, permitiendo así la gestión de los indicadores, realizar registros para efectuar seguimientos de estos a través del tiempo, generar reportes, gráficas y realizar un grupo de tareas que hacen más fácil el trámite de procesos a niveles empresariales. Ekomanager es un avance para poder llevar un excelente control y rendimiento



Capítulo 1. Fundamentación teórica

de la eficiencia energética, tiene como objetivos: un mejor aprovechamiento de los recursos, de la energía, la aplicación de ecoeficiencia y buenas prácticas ambientales. El uso de este software es restringido por leyes de *copyright*.(14)



Figura 2 Sistema para el seguimiento y mejora de la gestión ambiental y energética.

Accesible en: <http://www.ekomanager.com/>

Sistema de información ambiental

El sistema de información ambiental (SIA) lleva un control de la entrada de la materia prima destinada a la producción de leche pasteurizada, registra los consumos energéticos, los registros de producción y los desechos que generan tanto la producción como la limpieza de los equipos además de contribuir a facilitar la toma de decisiones por parte de la dirección de la empresa. Las herramientas que se utilizaron para su desarrollo fueron Macromedia Dreamweaver, HTML y PHP. No existe ninguna disponibilidad del código fuente y no existe posibilidad de adaptar al tema. (15)



Capítulo 1. Fundamentación teórica



Figura 3 Sistema de Información Ambiental.

Aporte del estudio de las aplicaciones a la propuesta de solución

Luego de haber realizado el análisis de varias herramientas sobre el tema, no se encontró ninguna que se pueda utilizar para la gestión de los procesos medioambientales en la ORASEN, pero se identificó que las aplicaciones para la gestión de la información tienden a desarrollarse de tipo web. Estos sistemas aportan una serie de elementos significativos para la futura propuesta de solución tales como: análisis de componentes, gráficos de comportamiento de indicadores, la concepción de reportes, la visualización de la información, la generación automática de informes, el manejo de datos mediante estadísticas y la gestión y centralización de información de forma segura.

Se concluye que los diferentes sistemas similares con respecto a la protección ambiental, brindan un espacio donde se podrá interactuar con la información necesaria sobre el tema, permitiéndoles generar reportes que sean de gran utilidad además de lograr una mejor comprensión de estos. Este estudio permitirá la reutilización de ideas que llevadas a la propuesta de solución harán de este un sistema que satisfaga los requerimientos de la oficina.



Capítulo 1. Fundamentación teórica

1.3 Lenguajes de programación

El análisis de la bibliografía describe a los lenguajes de programación “como un traductor entre el usuario y el equipo. En lugar de aprender el lenguaje nativo del equipo (conocido como lenguaje máquina), se puede utilizar un lenguaje de programación para dar instrucciones al equipo de un modo que sea más fácil de aprender y entender.”(16)

Un lenguaje de programación es un “...conjunto de reglas o normas que permiten asociar a cada programa correcto un cálculo que será llevado a cabo por un ordenador (sin ambigüedades)...”(17)

Después de realizado el análisis de varias bibliografías se llegó a la conclusión de que un lenguaje de programación es un conjunto de caracteres, símbolos y reglas que permiten interactuar con un ordenador para controlar su comportamiento.

Los lenguajes de programación que se han creado hasta el momento para el desarrollo de aplicaciones web se han agrupado en dos tipos: lenguajes de programación del lado del servidor y lenguajes de programación del lado del cliente.

Lenguajes de programación del lado del servidor

Los lenguajes de programación del lado del servidor son aquellos que se ejecutan e interpretan en el servidor y este envía al cliente la información de forma entendible para él. Este lenguaje es flexible a los cambios que pueden existir entre los navegadores y las versiones de estos.(18)

Algunos ejemplos de los lenguajes de programación del lado del servidor que se encuentran son Procesador de Hipertexto o *Hypertext Pre-Processor* (PHP), *Java Server Pages* (JSP) y *Microsoft Active Server Pages* (ASP).

ASP 3.0

Active Server Pages conocido por ASP es un lenguaje de programación del lado del cliente fundado por Microsoft para la creación de páginas dinámicas del lado del servidor. Su flexibilidad permite agregar componentes al servidor para realizar tareas, que de forma diferente no se pudiesen realizar: el envío de *e-mails*, acceso a archivos, gestión de variables del cliente o servidor, entre otros.

Este lenguaje funciona de forma muy sencilla, el servidor es quien se ocupa de ejecutarlo, interpretarlo y enviarlo al navegador en forma de código HTML. Este lenguaje no se escoge



Capítulo 1. Fundamentación teórica

porque el tipo de servidores que emplean son aquellos que funcionan con un sistema operativo de la familia de Windows y el cliente utiliza servidores GNU/Linux Ubuntu v13.04.(19)

PHP 5.3

PHP es creado en 1994 por Rasmus Lerdorf, ha tenido contribuciones de varios desarrolladores, es un lenguaje de programación del lado del servidor que se caracteriza por ser gratuito e independiente de plataforma, rápido, además posee un gran número de funciones, de documentación y gran seguridad.(18)

Es un lenguaje que facilita el acceso a bases de datos y conexiones de red. Existe un módulo de PHP para casi todos los servidores web, lo que hace que cualquier sistema pueda ser compatible con el lenguaje, lo que lleva una gran ventaja porque se puede trasladar de un sistema a otro sin casi ningún problema. *“PHP se encuentra muchas veces instalado sobre servidores Unix o Linux, que son de sobra conocidos como muy veloces y seguros(...), además, PHP permite configurar el servidor de modo que se permita o rechacen diferentes usos, lo que puede hacer al lenguaje(...) seguro dependiendo de las necesidades de cada cual(...).”*.(20)

Para la implementación de la propuesta de solución se seleccionó PHP al presentar varias ventajas que son de gran utilidad para el desarrollo de la aplicación, mostrando una gran cantidad de funciones, con las que se pueden realizar diferentes tipos de aplicaciones web. PHP permite: conexiones de red, tratamiento de fechas, creación de archivos, entre otros.(20)

Lenguajes de programación del lado del cliente

Los lenguajes de programación del lado del cliente permiten ejecutar código sin que sea enviado al servidor para ahorrar tiempo, es decir que no necesitan ser tratados directamente por el servidor. Algunos de los ejemplos de lenguajes de programación del lado del cliente son JavaScript, Flash, VBScript, XHTML, HTML, CSS, de los cuales se escogieron JavaScript, CSS y HTML, pues son los más utilizados porque tienen menor curva de aprendizaje y brindan mayores opciones.(21)

JavaScript 1.5

El lenguaje de programación JavaScript se utiliza principalmente para hacer mejoras de la interfaz de usuario, para hacer validaciones en los formularios de entrada de datos y generar gráficas para mostrar información, entre otros. Para el uso de JavaScript no es necesario compilar los programas para ejecutarlos, los programas que se encuentran escritos en JavaScript no necesitan de procesos intermedios para poder probarlos. Este lenguaje es



Capítulo 1. *Fundamentación teórica*

desarrollado y actualizado por Sun Microsystems.(21) Su compatibilidad con la gran mayoría de los navegadores modernos lo sitúa como uno de los más utilizados.(22)

CSS3 (*Cascading Style Sheets*)

Las hojas de estilo en cascada definen la estética del documento HTML, separando el contenido de la estructura de la página. Las hojas de estilo son utilizadas para definir el aspecto de cada elemento: color, tamaño, tipo de letra de texto, entre otros. Esta herramienta, que como se especifica anteriormente, se utiliza para dar estilo a documentos HTML, separando el contenido de la presentación, permite controlar el estilo de diversas páginas web en un mismo momento, cualquier cambio para un elemento afectará a todas las páginas que estén vinculadas a ese elemento.(23) (24)

HTML5

HTML es un lenguaje de marcado que predomina en la confección de páginas web, definiendo su sintaxis. Se utiliza para describir la estructura y el contenido, y presentarlo en forma de hipertexto como formato de las páginas web. Permite publicar documentos, imágenes, etc. (23)(25).

1.4 Frameworks de desarrollo web

En este epígrafe se hará un estudio sobre los principales tipos de *frameworks* (marcos de trabajo) que existen para el desarrollo de aplicaciones web, se analizarán CodeIgniter, Symfony y Doctrine ORM presentando sus principales características además se realizará un análisis sobre sus licencias y en qué versión se encuentran liberados.

Frameworks del lado del servidor

Existen diferentes *frameworks* del lado del servidor, para el desarrollo de la propuesta de solución se analizarán para la ejecución de la lógica del negocio a CodeIgniter y Symfony y para el acceso a datos a Doctrine.

CodeIgniter 2.1.4

CodeIgniter es un poderoso *framework* PHP construido para programadores que necesiten una herramienta muy simple para desarrollar sus aplicaciones web, presenta un excepcional desempeño, extensa y clara documentación, es posible desarrollar un sitio web haciendo pocas configuraciones, existe una comunidad extremadamente grande, no es necesario aprender



Capítulo 1. *Fundamentación teórica*

nuevas normas de codificación y no requiere aprender otros lenguaje para generar las plantillas o interfaces.

CodeIgniter se encuentra liberado bajo la licencia: Copyright (c) 2008 - 2011, EllisLab, Inc. que permite el uso, copia, modificación y distribución del software y de su documentación para cualquier propósito, su última versión liberada es la 2.1.4.

El *framework* es utilizado principalmente para desarrollar aplicaciones a pequeña escala y que no necesiten gran configuración de seguridad.(26)

Symfony 2.3

Symfony es un *framework* de código abierto utilizado para el desarrollo de aplicaciones web, el cual usa el lenguaje PHP. Fue originalmente concebido por la agencia interactiva SencioLabs para el desarrollo de sitios web y este fue liberado bajo la licencia MIT Open Source en el 2005 la cual expresa: "(...)Se concede permiso, de forma gratuita, a cualquier persona que obtenga una copia de este software y archivos de documentación asociados (el " Software"), para utilizar el Software sin restricción, incluido sin restricciones los derechos de usar, copiar, modificar, fusionar, publicar, distribuir, sublicenciar y/o vender copias del Software (...)".(27)

EL *framework* presenta soporte por la compañía, cuenta con una amplia comunidad que hace grandes aportes, contiene gran cantidad de documentación, listas de correo para la comunidad, donde estos aportan ideas y solucionan problemas comunes, tiene soporte profesional y entrenamientos en línea.

Luego de haber realizado un análisis de CodeIgniter y Symfony se llega a la conclusión de que el primero es un *framework* que ayudan a la creación de aplicaciones pequeñas y que no sea necesaria mucha seguridad, este no utiliza a Doctrine como capa de abstracción de datos, además cuenta con un sistema para la creación de sesiones de usuario extremadamente limitado, no utiliza el motor de plantillas Twig para mejorar la organización de las interfaces de usuario, por lo cual se toma a Symfony 2.3 por sus visibles ventajas que lo diferencian de los demás como: la utilización de *firewall* para comprobar el acceso de roles a determinadas direcciones, utiliza motor de plantillas Twig, cuenta con un sistema de seguridad nativa que permite integrarse con las tablas usuario y rol de una base de datos, utiliza a Doctrine como capa de abstracción de datos, posee amplio soporte por la empresa que lo desarrolla y una licencia que permite el uso y modificación libre, además una gran cantidad de bibliotecas para



Capítulo 1. Fundamentación teórica

el desarrollo y sobre todo el sistema de seguridad que presenta, que permite la asignación de roles a usuarios.

Propel ORM2.0

Propel es un kit de mapeo objeto relacional (ORM) de código abierto escrito en el lenguaje PHP el cual formaba parte del *framework* Symfony hasta su versión 2.0, este es un ORM creado para las bases de datos SQL, permite acceder a la información almacenada a través de objetos y proporciona una API para almacenar y recuperar datos. Además cuenta con una gran cantidad de documentación.

Por determinadas características como complejidad de configuración, complejidad para el tratamiento del modelo de datos y la aparición de Doctrine ORM, que brinda al usuario una capa de abstracción de datos mucho más amplia y con mejores características se sustituye en el *framework* Symfony a partir de su versión 2.0.(28)

Doctrine ORM 2.4.2

Doctrine es una biblioteca de funciones desarrolladas en PHP, es un poderoso mapeador de objetos y generador de consultas extremadamente flexible, soporta la programación tanto de alto nivel como de bajo nivel, este tiene una gran comunidad y se integra fácilmente con Symfony. Se encuentra liberada su versión 2.4.2, desde la liberación de la versión 2.0 este está dividido en diferentes proyectos los cuales son:

- **Object-Relational Mapper (ORM):** una de sus funciones es escribir consultas en la base de datos usando Doctrine Query Language (*DQL*), inspirado en Hibernate HQL el cual provee a los desarrolladores una alternativa al clásico SQL.
- **Common:** es la biblioteca que se encarga de brindar las extensiones y funcionalidades para el lenguaje PHP.

Este ORM se libera bajo la licencia GNU Lesser General Public License la cual permite el uso, distribución y modificación gratuita del software.

Actualmente se encuentra en su versión 2.2.0 el cual ya presenta un cúmulo importante de aplicaciones desarrolladas y contiene prestigiosas referencias como Yahoo, Dailymotion, Opensky, Exercise.com, phpBB y Drupal.(29)



Capítulo 1. *Fundamentación teórica*

Se toma a Doctrine ORM para el desarrollo de la aplicación por sus claras ventajas mostradas anteriormente como el mapeador de objetos y el generador de consultas extremadamente flexible que presenta, además de la cantidad de información brindada por la comunidad.

Frameworks del lado del cliente:

Existen diferentes *frameworks* del lado del cliente, para el desarrollo de la propuesta de solución se analizarán Ext JS y JQuery, se comparará el modo de ejecución en los distintos navegadores web, cantidad de componentes brindados y licencia bajo la cual está liberado.

JQuery 2.1.0

JQuery es una biblioteca de funciones desarrollada en JavaScript, rápida, pequeña y rica en características como el manejo de eventos, animaciones y un Ajax mucho más simple de usar. Trabaja alrededor de una multitud de navegadores, presenta un grupo de selectores de HTML para amenizar el trabajo de los desarrolladores y flexibilizar el trabajo con el Document Object Model (DOM).

De *JQuery*, su creador John Resig expresó: “*JQuery ha cambiado la forma en que millones de personas escriben JavaScript*”.(30)

Esta biblioteca presenta soporte por parte de la fundación JQuery. Se encuentra liberada la versión 2.1.0 bajo la licencia MIT que concede de forma gratuita el uso, copia y modificación de esta.

Ext JS 4.2

Este es un *framework* de desarrollo del lado del cliente el cual brinda una gran cantidad de componentes para ser utilizados por los desarrolladores, es puramente desarrollado en JavaScript.

Para el desarrollo de aplicaciones solo es necesario un navegador web y un editor de texto o un entorno integrado de desarrollo (IDE), dentro de los principales componentes se encuentran formularios que contienen algunos validadores de campo como: validadores de caracteres *alpha* y *e-mail*, contiene tablas (*Grids*) con un formato prefijado que son altamente modificables e incluyen un paginado muy intuitivo, además dentro de los componentes más importantes están un grupo de paneles y ventanas que dan a los usuarios la sensación de estar frente a una aplicación de escritorio.



Capítulo 1. *Fundamentación teórica*

Este *framework* contiene componentes capaces de crear gráficas como: pasteles, barras, área, línea y mixtas a partir de datos llegados del servidor, ayudando así a ver el comportamiento de variables a través de determinadas circunstancias.

La última versión: Ext JS 4.2 fue liberada el 13 de marzo de 2013 bajo la licencia *Open Source* y dos licencias comerciales.(31)(32) (33)

Para el desarrollo de la aplicación se escoge como *framework* del lado del cliente a Ext JS pues este brinda un grupo de componentes como paneles, gráficas, formularios y validadores que harán mucho más fácil el desarrollo de la solución.

1.5 Metodologías de desarrollo de software

Las metodologías de desarrollo de software son un conjunto de procedimientos y técnicas que permiten la construcción de un software con calidad. El uso de metodologías en el desarrollo de sistemas informáticos facilita el trabajo del equipo de desarrolladores y permite un mayor control en el proceso de desarrollo, obteniendo los resultados esperados en el menor tiempo posible. Dentro de las metodologías más utilizadas se encuentran:

Proceso Unificado de Desarrollo de Software (RUP)

El proceso Unificado de Desarrollo de Software es una metodología de desarrollo pesada la cual está centrada en la arquitectura, es un proceso iterativo e incremental y es guiado por casos de uso, utiliza como lenguaje de modelado el Unified Modelling Language (UML) para describir todo el proceso. El ciclo de vida de RUP está dividido en cuatro fases (inicio, elaboración, construcción y transición) en las cuales dependiendo del proyecto se realizarán las iteraciones que sean necesarias siguiendo un modelo secuencial o cascada orientado por los flujos de trabajo. Define 9 flujos de trabajo fundamentales, de los seis primeros se consideran tres como flujos de ingeniería y tres como flujos de apoyo, los flujos son: Modelado de Negocio, Requerimientos, Análisis y Diseño, Implementación, Pruebas, Despliegue, Configuración y Administración de Cambios, Administración de Proyecto y Entorno.(34) RUP es una metodología pesada que requiere considerable documentación, poca interacción con el cliente, cualquier error detectado en alguna etapa conduce necesariamente a rediseño y programación del código afectado, aumentando los costos de desarrollo, y, además, es sumamente metódica.



SCRUM

Capítulo 1. Fundamentación teórica

SCRUM es una metodología ágil orientada más a las personas que a los procesos. Dentro de las características que se destacan de estas metodologías están el desarrollo del software es realizado mediante iteraciones con una duración máxima de 30 días, el resultado de cada iteración es un incremento del producto que se muestra al cliente, además de las reuniones que se realizan a lo largo del proyecto. Esta metodología posee varias desventajas como: requiere un experto que monitorice su cumplimiento, plantea que no debe ser modificada, además representa un problema si el desarrollo del sistema está restringido por una fecha y precio de entrega cerrado por contrato.(35)(36)(37)

Extreme Programming (XP)

Metodología de desarrollo clasificada dentro del grupo de metodologías de desarrollo ligeras, orientada al cliente, y de iteraciones cortas. La base para el desarrollo de software que usa esta metodología son las llamadas historias de usuario (*User Stories*), historias escritas por el cliente en las que describe escenarios sobre el funcionamiento del sistema y que no sólo están limitados a los formularios de usuario, sino que también pueden describir modelos. Estas historias de usuario, junto a la arquitectura que se persigue, sirven de base para crear un plan de “entregas de software” entre el equipo de desarrollo y el cliente. Proponer integrar al equipo de desarrollo un representante constante del cliente que tenga amplios conocimientos sobre el negocio y que esté a disposición para cualquier duda, el cliente se mantiene en todo momento informado de las actividades que se están desarrollando y a medida que se realiza una entrega se realiza una discusión con el representante y se repite la nueva iteración del software. La programación siempre se define en pareja con el objetivo de lograr resultados con más calidad, mientras uno codifica una funcionalidad el otro se encarga de analizar si dicha funcionalidad es adecuada y está bien diseñada. (38)(39)(40)

Se realizó un estudio de las metodologías RUP, SCRUM y XP, para el desarrollo de la aplicación y se determinó que sería utilizada la metodología XP dado que permite realizar un trabajo orientado directamente al objetivo, basado en las relaciones interpersonales y la velocidad de reacción a los cambios que puedan ocurrir, además plantea la creación de un plan de entrega de las funcionalidades al cliente, define la programación del software en pareja, se cuenta con poco tiempo para la entrega de la herramienta y la participación activa del cliente en el proceso de desarrollo.



1.6 Herramientas CASE

Las herramientas CASE son herramientas informáticas destinadas a aumentar la productividad en el proceso de desarrollo de software, las cuales reducen tiempo y dinero, ayudando en el ciclo de desarrollo de software, en el diseño del proyecto, cálculo de costos e implementación de gran parte del código automáticamente.

Sommerville define como herramienta CASE (Ingeniería de Software Asistida por Computadora) “ (...) programas que son utilizados para ayudar a las actividades del proceso de software como el análisis de requerimientos, el modelado de sistemas, la depuración y las pruebas.”(39)

Rational Rose 7.0

Rational Rose es una herramienta CASE que proporciona un lenguaje común de modelado, presenta una alta capacidad de análisis de código, modelado UML para trabajar en diseños de clases, bases de datos y es capaz de generar informes para optimizar la comunicación entre el equipo. Rational Rose pertenece a la empresa SYBEX y se encuentra protegido por *copyright* provocando que no esté permitida su venta, distribución, publicación y su uso libre, por lo que no es escogido.

Visual Paradigm 8.0

Visual Paradigm para el Lenguaje Unificado de Modelado (UML) es una herramienta que soporta casi todo el ciclo de desarrollo de software, análisis y diseño orientados a objetos, construcción, pruebas y despliegue, permite hacer dibujos de todo tipo de diagramas de clases, generar código desde diagramas y genera documentación, se distribuye bajo cuatro tipos de licencias: licencia de un solo asiento (*Single seat license*), licencia flotante (*Floating license*), licencia de suscripción (*Subscription license*) y la licencia académica (*Subscription license*) que permite el uso del software para motivos académicos, por lo que es utilizada en esta investigación.

Las principales características de la herramienta son:

- Soporta aplicaciones web.
- Varios idiomas.
- Generación de código para Java y exportación como HTML (Lenguaje de Mercado de Hipertexto).



Capítulo 1. Fundamentación teórica

- Fácil de instalar y actualizar.
- Compatibilidad entre ediciones.

En cuanto a las herramientas se debe tener en cuenta las características del entorno de desarrollo y las licencias, por lo cual es elegido el Visual Paradigm para UML 8.0 como herramienta CASE para modelar los procesos del sistema.(41)(42)(43)

1.7 Sistemas gestores de base de datos:

Los Sistemas de Gestión de Base de Datos son herramientas informáticas que sirven de interfaz entre el usuario, las aplicaciones que utiliza y la bases de datos, en su mayoría utilizan el lenguaje SQL como lenguaje de consulta.

MySQL Database Server 5.5.9

MySQL es el sistema de administración de bases de datos de código abierto más usado en el mundo pues es un servidor de bases de datos sencillo y rápido de usar, es multiplataforma y la extensiva reutilización del código dentro del software ha dado lugar a un sistema de administración de la base de datos incomparable en velocidad, compactación, estabilidad y facilidad de despliegue.(44)

PostgreSQL 9.1

PostgreSQL es un sistema gestor de bases de datos objeto-relacional bajo la licencia BSD la cual a diferencia de la licencia GPL permite el uso del código fuente para el desarrollo de software no libre, utiliza el modelo cliente servidor y multiprocesos en vez de multihilos garantizando la estabilidad del sistema. Este presenta soporte nativo para el lenguaje PHP, además se puede encontrar en los principales sistemas operativos como: GNU/Linux, Windows y Mac. Dentro de sus principales ventajas se encuentra que soporta la gran mayoría de las características de una base de datos profesional como: *triggers*, funciones, secuencias, relaciones, reglas, tipos de datos definidos por usuarios y vistas, además se ser altamente adaptable.

Dentro de sus principales característica se encuentran:

- **Atomicidad (Indivisible):** esta propiedad da al sistema la capacidad de decir al usuario si alguna tarea fue finalizada o no, ninguna tarea puede quedar a media.



Capítulo 1. Fundamentación teórica

- **Consistencia:** se ejecutan aquellas tareas que no romperán las reglas o las directrices de las bases de datos.
- **Aislamiento:** propiedad que permite que ninguna tarea afecte a las demás.
- **Durabilidad:** una vez que sea realizada una operación esta persistirá.(45)(46)

Luego del análisis de las características de los dos gestores de bases de datos se puede concluir que PostgreSQL presenta todos los requisitos para mantener la integridad y seguridad de los datos ya que este tiene pleno apoyo en todas las características de una base de datos profesional como *triggers*, funciones, secuencias, relaciones, reglas, etc.

1.8 Entorno de desarrollo integrado (IDE)

Los entornos de desarrollo integrado son programas informáticos que contienen diferentes herramientas que son muy útiles para los desarrolladores de software, ya que estos entornos optimizan la tarea de escribir programas, corregirlos y ejecutarlos.

Zend Studio 10.0.0

Zend Studio es un entorno escrito en Java que en ocasiones se torna un poco lento, funciona en los principales sistemas operativos como Windows, GNU/Linux y Mac OS X, dispone de características como: resaltado de sintaxis, ayuda y autocompletado de código y detección de errores en tiempo real. El software no es distribuido bajo una licencia gratuita.(47)

NetBeans IDE 7.3

NetBeans permite desarrollar aplicaciones de escritorio y web en Java, además es posible trabajar con HTML5, JavaScript y CSS, proporciona herramientas para desarrollar en PHP, es de código abierto y gratuito, presenta una gran comunidad en todo el mundo y puede ser utilizado tanto en Windows, GNU/Linux y Mac. Se encuentra liberado en su versión 7.3.(48)

Al analizar las herramientas anteriormente mencionadas se decidió utilizar el NetBeans IDE como entorno de desarrollo integrado por la integración que tiene este con los *frameworks* anteriormente seleccionados.



1.9 Servidores web

Según la bibliografía estudiada un servidor web es un programa informático que permite transmitir hipertextos para que los usuarios puedan acceder a esta información a través de un ordenador remoto. Los servidores web suministran las páginas web que son solicitadas por los navegadores.

Internet Information Server (IIS) 8.5

Es un servidor web, creado por Microsoft. Posee componentes programables desde ASP accediendo a cada uno de sus módulos para cada función específica. Este servidor web permite procesar páginas de ASP y ASP.NET, también se pueden incluir PHP o Perl. Una de las ventajas que presenta es que permite alojar varios sitios web en un solo equipo, además reiniciar los servicios de internet sin tener que reiniciar el equipo. Este es creado para Windows pero no viene instalado por defecto en el sistema operativo.(49)

Respecto a su seguridad y administración se puede decir en ese orden que permite asignar o rechazar accesos web a equipos individuales, grupos de equipos o dominios enteros, facilita de forma segura el intercambio de información entre el cliente y los servidores. IIS posee herramientas fundamentales de administración en web que admiten la administración remota del servidor desde casi cualquier explorador en cualquier plataforma además de poder realizar copia de seguridad, cambios en la capacidad y los procedimientos para efectuar las copias de seguridad.(49)

Apache 2.2

Es uno de los servidores web más robustos y estables del mundo lo que lo convierte en uno de los más usados. Está dividido en módulos que pueden ser instalados de acuerdo a las necesidades. Su licencia descende de las licencias BDS que permiten configurar el código fuente a gusto siempre y cuando se reconozca el trabajo. Dentro de sus principales características se encuentran que es multiplataforma y que es altamente configurable en la gestión de *logs*. Un servidor web Apache viene definido como “ ... un servidor web HTTP de código abierto para plataformas Unix (BSD, GNU/Linux, etc.), Microsoft Windows, Macintosh y otras, que implementa el protocolo HTTP/1.1 y la noción de sitio virtual.”(50)



Capítulo 1. *Fundamentación teórica*

Se propone utilizar Apache para la solución del problema porque es altamente configurable y modular, permite alojar varios sitios web en una misma máquina, es altamente configurable en la gestión de logs y es posible ejecutarlos en los principales sistemas operativos.

1.10 Herramientas para la edición de imágenes

La edición de imágenes digitales tiene como metas el tratamiento de estas con el fin de eliminar fallas como falta de contraste, ruido en la imagen, etc. Existen varias herramientas de este tipo, algunas de ellas serán analizadas a continuación.

Adobe Photoshop CS3

Adobe Photoshop es una aplicación informática creada por la compañía Software Adobe System Inc., está destinada a la edición, retoque fotográfico y pintura de imágenes. En sus inicios es creado para computadoras Apple pero luego es creado para el sistema operativo Windows.

Adobe Photoshop es un software privativo utilizado en su mayoría por diseñadores y fotógrafos profesionales. El propósito principal de la herramienta es la edición fotográfica aunque también se pueden crear imágenes, gráficos, efectos con muy buena calidad. Soporta varias extensiones de imágenes como .jpg, .gif, .png, .bmp entre otros. La herramienta se puede encontrar en diversos idiomas alrededor del mundo.

GIMP 2.8

GIMP (GNU Image Manipulation Program) es un programa libre y gratuito, utilizado para la edición de imágenes digitales, que pueden ser dibujos o fotografías. Este programa se encuentra disponible bajo la Licencia pública general de GNU, fue creado para GNU/Linux, aunque Windows y Mac OS X cuentan con versiones de esta herramienta(51)

La herramienta trabaja con la mayoría de las extensiones de imágenes como son .jpg, .gif, .png, entre otras. Posee varias herramientas que permiten realizar funciones como son combinar, seleccionar y pintar imágenes, herramientas de tratamiento de textos y el aspecto de las imágenes, además cuenta con un menú para el catálogo de efectos. (51)

Se seleccionó la herramienta GIMP, en su versión 2.8, para la edición de imágenes, por considerarse el más adecuado debido a que es utilizado en diferentes sistemas operativos,



Capítulo 1. Fundamentación teórica

posee herramientas claves que permiten el tratamiento y la creación de imágenes para la aplicación, es libre, gratuita y el equipo de desarrollo tiene experiencia en su trabajo.

Conclusiones

El análisis bibliográfico muestra que los procesos medioambientales controlados por la ORASEN, por ser una temática propia de esta oficina y de los centros adscritos a ella, presentan una terminología en ocasiones propia de sus procesos, pero los mismos se encuentran documentados en un grupo de leyes, decretos y decretos leyes dictados por la dirección del ministerio y del país, que regulan el actuar de las partes en dichos procesos.

Se evidencia un alto compromiso del país con el control y seguimiento de la información ambiental, el mismo fue ratificado en el VI Congreso del Partido Comunista de Cuba.

El estudio del estado del arte muestra la inexistencia de una herramienta que contribuya al control y seguimiento de los procesos medioambientales que realiza la ORASEN, además que existen insuficiencias de las herramientas similares existentes para poder ser utilizadas en esta oficina. Se concluye además en la necesidad de desarrollar una aplicación web, utilizando tecnologías libres, que contribuya a la mejora de la problemática planteada en la investigación.



Capítulo 2. Descripción de la solución propuesta

Capítulo 2 Descripción de la solución propuesta

En el capítulo anterior se realizó un análisis del estado del arte donde quedaron definidos los métodos y las herramientas necesarios para poder plantear una solución al problema descrito.

En el presente capítulo se hace alusión a los elementos esenciales para el desarrollo de la solución como es definir los diferentes usuarios que van a interactuar con la aplicación, según la metodología XP, se abordarán las fases de exploración, planificación y diseño, en las que se generan las historias de usuario, los planes de iteraciones, las tarjetas CRC, entre otros.

2.1. Aspectos no funcionales del sistema

Cada sistema presenta distintas características no funcionales que a la par de las funciones que realiza lo haga marcar la diferencia entre un producto de alta aceptación y uno de poca aceptación, estas características en conjunto crean un sistema capaz de proporcionarle al cliente un atractivo visual, usabilidad, rápido desempeño y que llene sus expectativas.

Usabilidad

El sistema debe presentar un acceso fácil y rápido a los componentes para facilitar el uso del mismo a los usuarios con pocos conocimientos en el campo de la informática. Para una fácil comprensión y adaptación, el producto deberá tener semejanza con el estilo de trabajo manual que se emplea en la oficina.

Disponibilidad

La aplicación siempre estará disponible para todos los usuarios que pertenecen a la red de la oficina, por lo que con solo autenticarse, siempre que los datos de autenticación sean correctos, podrá acceder al sistema.

Apariencia o Interfaz Externa

El sistema proporciona al cliente una interfaz que lo mantiene cómodo en todo momento. La información está organizada según los procesos que se realizan, mantiene una constancia en el diseño de la aplicación, presenta una buena arquitectura de la información para que el cliente sea capaz de encontrar lo que está buscando en todo momento. Muestra colores agradables a la vista, el tamaño y tipo de letra está acorde con las preferencias del cliente.



Capítulo 2. Descripción de la solución propuesta

Seguridad

Se plantea una aplicación cuya seguridad sea orientada a roles lo que permite asignar los permisos de distintas acciones a distintos tipos de roles y asignar a estos los usuarios que necesitará el sistema. Es necesario destacar que solo los usuarios pertenecientes al sistema tienen acceso a sus funcionalidades.

La autenticación se implementó haciendo uso del sistema de seguridad que brinda el *framework* Symfony el cual permite a determinados roles acceder o no a distintas funciones que presentará la aplicación, además permite el uso de algoritmos de encriptación en las contraseñas para su envío por la red y almacenamiento en la base de datos.

Como Symfony 2.3 es uno de los *framework* para el desarrollo de aplicaciones web más utilizados en el mundo este presenta una infraestructura de seguridad que permite la protección contra ataques XSS (Cross Site Scripting), inyección SQL y otros tipos, además, utiliza como ORM a Doctrine el cual permite una capa de seguridad extra en cuanto al acceso a la base de datos.

Otra de las principales capas de seguridad presentadas por Symfony es la implementación de firewalls, los cuales permiten mediante el uso de rutas y roles determinar si el usuario presenta permiso o no de acceder a diferentes recursos brindados por la aplicación.

Confidencialidad

Una de las principales características de un sistema basado en roles es la confidencialidad: permite que un usuario no sea capaz de ver información a la que no tenga permiso.

Requisitos de Hardware

Servidor web

- RAM: 1GB o superior
- HDD: 500 MB libre o más
- CPU: Dual Core 2.0GHz o superior

Servidor de bases de datos:

- RAM: 1GB o superior
- HDD: 500 MB libre o más
- CPU: Dual Core 2.0GHz o superior



Capítulo 2. Descripción de la solución propuesta

Requisitos de software (servidor)

Servidor web:

- Sistema Operativo: Ubuntu server 12.04 o superior
- Servidor web: Apache2 versión: 2.2.22 o superior

Datos adicionales:

- PHP5 o superior
- php5-psql
- libapache2-mod-php5

Servidor de Base de Datos:

- Sistema Operativo: Ubuntu server 12.04 o superior
- Sistema Gestor de Base de Datos: PostgreSQL 9.1 o superior

Requisitos de software (cliente)

- Navegador Web: Mozilla Firefox 19 o superior, Google Chrome 8 o superior.

2.2. Planificación del sistema

Para una mejor organización en el desarrollo del software, este se realizará siguiendo el orden de las fases de la metodología escogida: XP. La fase de planificación del sistema consiste en un plano de comunicación constante entre el cliente y los programadores. En esta fase se realiza una estimación de esfuerzo requerido para la implementación de cada historia de usuario, los clientes deciden sobre el tiempo de entrega de cada iteración además de establecer la prioridad de cada historia de usuario, según el valor que tengan para el negocio. Por otra parte los programadores están encargados de estimar el esfuerzo asociado a cada historia de usuario. Las historias de usuario son ordenadas según su prioridad y esfuerzo pudiendo definir el contenido y entrega de cada iteración.

Usuarios definidos para el sistema

Para el desarrollo del sistema se determinaron cuatro roles dependiendo de las responsabilidades necesarias para el flujo del proceso, esto permitirá que usuarios de distintos roles no tengan las mismas libertades dentro del sistema, es decir que un usuario podrá realizar tareas que otros no.

Los nombres determinados para cada uno de los cuatros roles son:



Capítulo 2. Descripción de la solución propuesta



Figura 4 Definición de Roles.

Historias de usuario

Las Historias de usuario (HU) son un tipo de técnica utilizadas en *Extreme Programming* (XP), por el usuario, para especificar los requisitos que debe tener el software. En estas el cliente realiza una descripción breve de las características que desea que el sistema tenga, estas características pueden ser de tipo funcional o no funcional. Las HU pueden sufrir cambios o remplazos en cualquier momento, poseen gran flexibilidad y deben estar redactadas de forma comprensible para una rápida implementación por parte de los usuarios.

Existen sugerencias de varias plantillas para representar la información que debe tener una HU, la mayoría de la veces lo que se propone es utilizar un nombre y una descripción, también se le puede agregar una estimación de esfuerzos, en días. (52, 53)

Se llegó al consenso con el cliente que se tomarán como “gestionar” las funcionalidades adicionar, modificar y eliminar de un mismo proceso.



Capítulo 2. Descripción de la solución propuesta

A continuación se presentan ejemplos de algunas HU. Para ver las restantes HU se debe acceder a: Anexo 1 Historias de usuario.

Tabla 1 Historia de Usuario: Generar Reportes.

Historia de usuario	
No. 13	Nombre: Generar reportes
Usuario: Director	
Modificación de HU: Ninguna	
Programador: Gisela Sagarra Mendoza– Edry Garcia Ramos	
Prioridad de Negocio: Alta	Puntos Estimados: 0.5
Nivel de Complejidad: Alta	Iteración Asignada: 4
Descripción: Genera reporte según los datos que estén en la base de datos	
Observaciones: Devuelve los reportes en la extensión .pdf.	

Tabla 2 Historia de Usuario Gestionar medidas.

Historia de usuario	
No. 8	Nombre: Gestionar medidas
Usuario: Usuario, Director, Administrador, Experto	
Modificación de HU: Ninguna	
Programador: Gisela Sagarra Mendoza – Edry Garcia Ramos	
Prioridad de Negocio: Alta	Puntos Estimados: 1
Nivel de Complejidad: Baja	Iteración Asignada: 3
Descripción: Permite crear, modificar y eliminar medidas en la aplicación	
Observaciones:	



Capítulo 2. Descripción de la solución propuesta

Tabla 3 Historia de usuario Gestionar autorización.

Historia de usuario	
No. 3	Nombre: Gestionar autorización
Usuario: Usuario, Director, Administrador, Experto	
Modificación de HU: Ninguna	
Programador: Gisela Sagarra Mendoza – Edry Garcia Ramos	
Prioridad de Negocio: Alta	Puntos Estimados: 1
Nivel de Complejidad: Baja	Iteración Asignada: 3
Observaciones:	

Tabla 4 Historia de usuario Gestionar suceso extraordinario.

Historia de usuario	
No. 14	Nombre: Gestionar suceso extraordinario
Usuario: Usuario, Director, Administrador, Experto	
Modificación de HU: Ninguna	
Programador: Gisela Sagarra Mendoza – Edry Garcia Ramos	
Prioridad de Negocio: Alta	Puntos Estimados: 0.5
Nivel de Complejidad: Baja	Iteración Asignada: 4
Observaciones:	

Estimación de esfuerzos por HU

La estimación de esfuerzos para cada HU es establecida por los programadores. Se miden en semanas y debe ser de una a tres, en caso de que el tiempo sea menor que una semana se combina con otra HU y si es mayor que tres semanas debe ser dividida hasta que su realización sea en un tiempo igual o menor a las tres semanas.



Capítulo 2. Descripción de la solución propuesta

Tabla 5 Estimación de esfuerzos por Historia de Usuario.

No.	Historia de usuario	Estimación de esfuerzo (semanas)
1	Gestionar usuario	1
2	Autenticar usuario	0.5
3	Gestionar autorización	1
4	Mostrar tipo de autorización	0.5
5	Gestionar inspección	1
6	Mostrar tipo de inspección	0.5
7	Mostrar estado de la inspección	0.5
8	Gestionar medidas	1
9	Mostrar resultado de las medidas	0.5
10	Mostrar normativa de la medida	0.5
11	Gestionar desechos	0.5
12	Mostrar categoría de desechos	0.5
13	Generar reporte	1
14	Gestionar suceso extraordinario	1
15	Mostrar tipo de suceso extraordinario	0.5
16	Gestionar conciliaciones	1
17	Generar gráficos	0.5

Plan de iteraciones.

El plan de iteración está compuesto por las historias de usuarios que quedaron definidas en el plan de entrega. Las iteraciones están en el rango de una a tres semanas

Iteración 1: En esta primera iteración se realizará la entrega de alguna de las funcionalidades que poseen un nivel de prioridad alta, siendo estas:

HU 1: Gestionar usuario

HU 2: Autenticar usuario

HU 3: Gestionar autorización

HU 4: Mostrar tipo de autorización



Capítulo 2. Descripción de la solución propuesta

Iteración 2: En esta iteración se continuará con la realización de HU con prioridad de tipo alta para el cliente:

HU 5: Gestionar Inspección

HU 6: Mostrar tipo de inspección

HU 7: Mostrar estado de la inspección

HU 11: Gestionar desechos

HU 12: Mostrar categoría de desechos

Iteración 3: En la iteración número tres se realizan las historias de usuario que restan y presentan prioridad alta para el cliente:

HU 8: Gestionar medidas

HU 9: Mostrar el resultado de las medidas

HU 10: Mostrar la normativa de las medidas

HU 13: Generar reportes

Iteración 4: Finalmente en esta iteración se implementan las últimas HU, quedando definidas de la siguiente manera:

HU 14: Gestionar suceso extraordinario

HU 15: Mostrar tipo de suceso extraordinario

HU 16: Gestionar conciliaciones

HU 17: Generar gráficos

Plan de duración de las iteraciones.

El plan de duración de las iteraciones es realizado luego de tener el tiempo estimado en días que demora implementar cada HU, para su realización se toma en cuenta el nivel de complejidad que presentan y la prioridad que tienen para el cliente.

Tabla 6 Plan de duración de las iteraciones.

Iteraciones	Orden de las HU	Duración total (semanas)
1	Gestionar usuario Autenticar usuario Gestionar autorización Mostrar tipo de autorización	3
2	Gestionar Inspección Mostrar tipo de inspección	3



Capítulo 2. Descripción de la solución propuesta

	Mostrar estado de la inspección Gestionar desechos Mostrar categoría de desechos	
3	Gestionar medidas Mostrar el resultado de las medidas Mostrar la normativa de las medidas Generar reportes	3
4	Gestionar suceso extraordinario Mostrar tipo de suceso extraordinario Gestionar conciliaciones Generar gráficos	3

Plan de entrega.

El plan de entrega consiste en tener reunidos las historias de usuario en varias iteraciones. En este plan se seleccionan las historias de usuario de cada iteración y las pruebas funcionales correspondientes a cada una serán validadas al final de cada iteración. El cronograma de entrega establece que historias de usuario serán agrupadas para establecer el orden de su entrega, este cronograma es el resultado de una reunión entre el cliente y el equipo de desarrollo.

Tabla 7 Plan de entrega.

Historia de usuario	1ra Iteración	2da Iteración	3ra Iteración	4ta Iteración
Gestionar usuario	V1.0	Finalizado	Finalizado	Finalizado
Autenticar usuario	V1.0	Finalizado	Finalizado	Finalizado
Gestionar autorización	V1.0	Finalizado	Finalizado	Finalizado
Mostrar tipo de autorización	V1.0	Finalizado	Finalizado	Finalizado
Gestionar inspección	-	V1.0	Finalizado	Finalizado



Capítulo 2. Descripción de la solución propuesta

Mostrar tipo de inspección	-	V1.0	Finalizado	Finalizado
Mostrar estado de la inspección.	-	V1.0	Finalizado	Finalizado
Gestionar desechos	-	V1.0	Finalizado	Finalizado
Mostrar categoría de desechos	-	V1.0	Finalizado	Finalizado
Gestionar medidas	-	-	V1.0	Finalizado
Gestionar resultado de la medida.	-	-	V1.0	Finalizado
Mostrar normativa de la medida	-	-	V1.0	Finalizado
Generar reportes.	-	-	V1.0	Finalizado
Gestionar sucesos extraordinarios.	-	-	-	Finalizado
Gestionar tipo de suceso extraordinario	-	-	-	Finalizado
Gestionar conciliaciones.	-	-	-	Finalizado
Generar gráficos	-	-	-	Finalizado

2.3. Diseño del sistema

La metodología XP plantea que el diseño del sistema debe ser sencillo, en el caso de que alguna parte del sistema sea de desarrollo complejo, lo adecuado es que sea dividido en varias partes. En el caso de que existiesen problemas en los diseños, se recomienda que sean corregidos rápidamente, de lo contrario se verían presentados en el producto, reduciendo su calidad. Un correcto diseño del sistema brinda la posibilidad de mayor éxito al finalizar el producto deseado. En este caso se desarrollarán las tarjetas clase responsabilidad colaborador, se escogerán los tipos de patrones de diseño a utilizar y quedarán definidos los prototipos de interfaz de usuario.



Capítulo 2. Descripción de la solución propuesta

Tarjetas Clases Responsabilidad Colaborador CRC

Según el análisis realizado por los autores de varias bibliografías la metodología XP incita el uso de tarjetas Clase-Responsabilidad-Colaborador (CRC) como una mejor forma para pensar en el software orientado a objetos. Estas tarjetas se generan en la fase de diseño de la metodología, son divididas en tres secciones que contienen la información del nombre de la clase, sus responsabilidades y colaboradores. Una clase puede ser cualquier objeto, las responsabilidades de una clase son sus atributos y métodos, los colaboradores son clases utilizadas para realizar sus responsabilidades.(54) (55)

Las tarjetas CRC permiten identificar y acomodar las clases que son importantes para el incremento actual del software, además son mostradas al cliente de manera que se pueda llegar a un acuerdo de la validez de estas. Las tarjetas le permiten al programador centrarse y poder realizar un trabajo más rápido y seguro dejando a un lado la programación procedural y dando uso a la programación orientada a objetos (POO).(54)(55)

Tabla 8 Tarjeta CRC SIGA.portal.Inspecciones.View.

SIGA.portal.Inspecciones.View	
Responsabilidades	Colaboradores
✓ init	✓ Ext.data.Model
✓ editInsp	✓ SIGA.Lib.Util
✓ dellInsp	✓ SIGA.Lib.Util
✓ gridInsp	✓ Ext.grid.Panel
✓ windowsEdit	✓ Ext.Window ✓ Ext.TabPanel ✓ Ext.form.FormPanel
✓ windowsSave	✓ Ext.Window ✓ Ext.TabPanel ✓ Ext.form.FormPanel



Capítulo 2. Descripción de la solución propuesta

Tabla 9 Tarjeta CRC SIGA.portal.Inspecciones.

SIGA.portal.Inspecciones	
Responsabilidades	Colaboradores
✓ init	✓ Ext.data.Model
✓ editInsp	✓ SIGA.Lib.Util
✓ dellInsp	✓ SIGA.Lib.Util

Tabla 10 Tarjeta CRC SIGA.portal.Autorizaciones.View.

SIGA.portal.Autorizaciones.View	
Responsabilidades	Colaboradores
✓ windowsReport	✓ Ext.Window ✓ Ext.TabPanel
✓ chartAuto	✓ Ext.chart.Chart ✓ Ext.Panel
✓ gridAuto	✓ Ext.grid.Panel
✓ windowsEdit	✓ Ext.Window ✓ Ext.form.FormPanel
✓ windowsSave	✓ Ext.Window ✓ Ext.form.FormPanel

Tabla 11 Tarjeta CRC SIGA.portal.Autorizaciones.

SIGA.portal.Autorizaciones	
Responsabilidades	Colaboradores
✓ init	✓ Ext.data.Model
✓ editAuto	✓ SIGA.Lib.Util
✓ delAuto	✓ SIGA.Lib.Util
✓ reportAuto	✓ SIGA.Lib.Util



Capítulo 2. Descripción de la solución propuesta

Patrones de diseño

Existen diversos tipos de patrones de diseño con distintos propósitos. Se pueden encontrar los patrones de comportamiento, los patrones estructurales, los patrones de creación, entre otros.

Patrones Generales de Software para Asignar Responsabilidades (GRASP)

Realizar una correcta asignación de responsabilidades es un paso muy importante para el desarrollo de un sistema orientado a objetos, es por ello que se decidió hacer uso de los patrones generales de software para asignar responsabilidades, los patrones GRASP, quedando definidos de la siguiente manera: “... describen los principios fundamentales de la asignación de responsabilidades a objetos, expresados en forma de patrones ...”(56)

Experto

Este patrón es el principio básico para la asignación de responsabilidades para el diseño de sistemas orientados a objetos. Es el encargado de asignar una responsabilidad al experto en información, es decir, la clase que tiene la información necesaria para realizar determinada responsabilidad.

Ese patrón se evidencia en todo el sistema, ejemplo en la clase de nombre “Autorizacion” encargada de almacenar todos los atributos que son necesitados por dicha clase.

Creador

Este patrón permite asignar a determinada clase (A) una instancia de otra clase (B) si se cumple uno o más de los casos siguientes:

- A agrega objetos de B.
- A contiene objetos de B.
- A registra instancia de objetos de B.

El principal problema a resolver de este patrón es: ¿Quién debería ser el responsable de la creación de una nueva instancia de alguna clase?

Una de las actividades más utilizadas en aplicaciones orientadas a objetos es la creación de instancias de distintas clases por lo que es necesario contar con un principio que permita realizar la asignación de la responsabilidad de creación. Un buen uso de este patrón puede significar para una aplicación informática un bajo acoplamiento, mayor claridad, encapsulación y reutilización. Es posible encontrar un ejemplo del uso de este patrón en la clase



Capítulo 2. Descripción de la solución propuesta

“InspeccionController” la cual es la encargada de crear, modificar, visualizar y eliminar las inspecciones.

Bajo acoplamiento

La base de este patrón es: asignar una responsabilidad de manera que el acoplamiento permanezca bajo. Este patrón soporta el diseño de las clases del sistema cada vez más independientes una de otra, lo cual reduce el impacto en los cambios.

Este patrón se utilizará en toda la aplicación ya que cada proceso es modelado con la menor cantidad de relación posible con los demás, lo que permite separar cualquier proceso sin que produzcan grandes cambios en los demás.

Alta cohesión

Patrón de asignación de responsabilidades que busca la solución para asignar una responsabilidad de manera que la cohesión permanezca alta. Su principal problema es: ¿Cómo mantener la complejidad manejable?

La cohesión es la medida de focalización que presenta un objeto en cuanto a sus responsabilidades, un objeto que presenta baja cohesión realiza actividades que no están relacionadas con su objetivo, o hace demasiado trabajo, esto lleva a los siguientes problemas:

- Difíciles de entender.
- Difíciles de reutilizar.
- Difíciles de mantener.

En muchos casos las clases que presentan una baja cohesión es porque a estas se le asignaron responsabilidades que deberían haberse delegado a otras clases, este patrón es evidenciable en el software ya que cada clase para realizar sus responsabilidades se relaciona con otras evitando realizar acciones que no le correspondan.

Controlador

El principal objetivo de este patrón es asignar la responsabilidad de recibir o manejar un mensaje de evento del sistema a una clase que representa el sistema global, un dispositivo o un subsistema, el problema a resolver de este patrón es: ¿Quién debe ser el responsable de gestionar un evento de entrada al sistema? (56)

Este patrón, en la aplicación, se podrá encontrar en las clases que tengan la terminación “Controller”.



Capítulo 2. Descripción de la solución propuesta

Patrones arquitectónicos

Los patrones arquitectónicos o patrones de arquitectura de software muestran un amplio repertorio de principios generales basados en la experiencia, que proponen una guía para crear el software. Presentan un esquema que se organiza en estructuras, esencial para el software. Los patrones arquitectónicos se especifican describiendo sus componentes, con sus responsabilidades, relaciones y la forma en que colaboran, son una vía de solución a los problemas de arquitectura en la ingeniería de software.(57)(56)

Modelo Vista Controlador (MVC)

El patrón de arquitectura Modelo Vista Controlador (MVC) es uno de los patrones de arquitectura de software más conocidos y utilizados, permite realizar programación multicapa, quedando separados en tres componentes distintos: los datos de la aplicación, la interfaz de usuario y la copia de control. Estos componentes quedan definidos de la siguiente forma: el Modelo maneja los datos del programa y controla sus transformaciones, la Vista estaría representada por la interfaz que interactúa con el usuario, recibe datos del modelo y los muestra al usuario, se debe tener en cuenta que pueden existir una o más vistas y el Controlador es el encargado de administrar todas las acciones que pueda realizar la aplicación, responde a eventos que usualmente son acciones de los usuarios, entre otras. En la aplicación se puede apreciar el modelo como las clases que identifican a cada una de las tablas de la base de datos generada por la herramienta Doctrine, ejemplo: la clase "Autorizacion", las vistas serían las interfaces creadas con el framework Ext JS, ejemplo "AdicionarAutorizacion", y el controlador "AutorizacionController".



Capítulo 2. Descripción de la solución propuesta

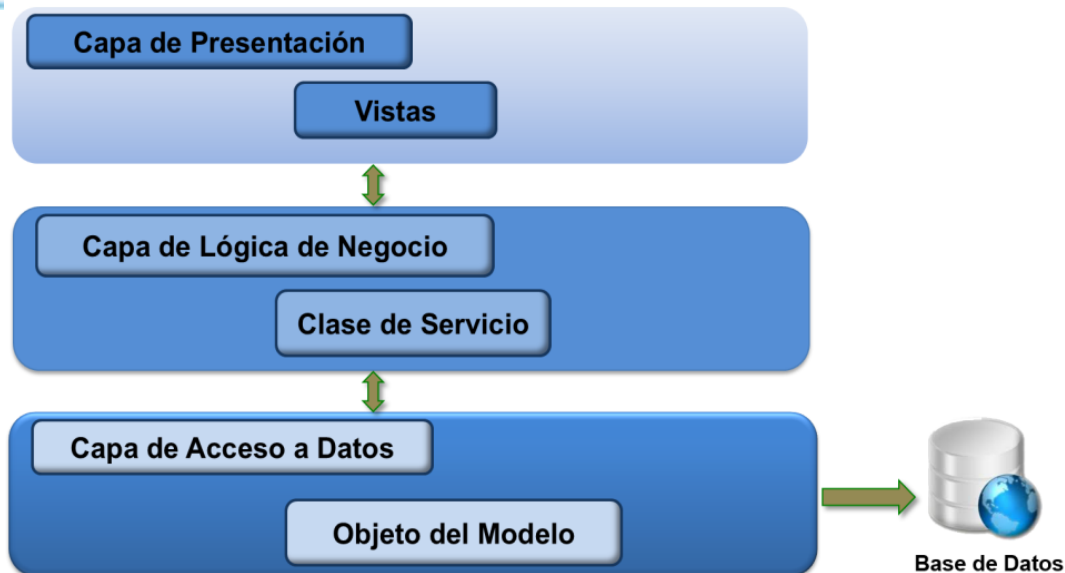


Figura 5 Diagrama del patrón de Arquitectura de Software Modelo Vista Controlador (MVC).

Prototipo de interfaz de usuario

Los prototipos de interfaz de usuario permiten al equipo de desarrollo tener una idea de cuáles serán las partes visuales del sistema y de la misma manera al cliente confirmar el entendimiento de los requisitos del sistema y poder comenzar con el desarrollo de las interfaces de usuario. A continuación se presenta el prototipo de la interfaz principal y el prototipo del formulario de autorización.



Capítulo 2. Descripción de la solución propuesta

Menu		Window Name						Salir	
Procesos		Titulo Gráficos							
Proceso 1		Adicionar Modificar Reporte							
Proceso 2		Dato1	Dato2	Dato3	Dato4	Dato5	Dato6	Dato7	Dato8
Proceso 3									<input checked="" type="checkbox"/>
Proceso 4									<input checked="" type="checkbox"/>
									<input checked="" type="checkbox"/>
									<input checked="" type="checkbox"/>
									<input checked="" type="checkbox"/>
									<input checked="" type="checkbox"/>
									<input checked="" type="checkbox"/>
									<input checked="" type="checkbox"/>
									<input checked="" type="checkbox"/>
									<input checked="" type="checkbox"/>
									<input checked="" type="checkbox"/>
									<input checked="" type="checkbox"/>

Figura 6 Prototipo de Interfaz de usuario de la página principal.

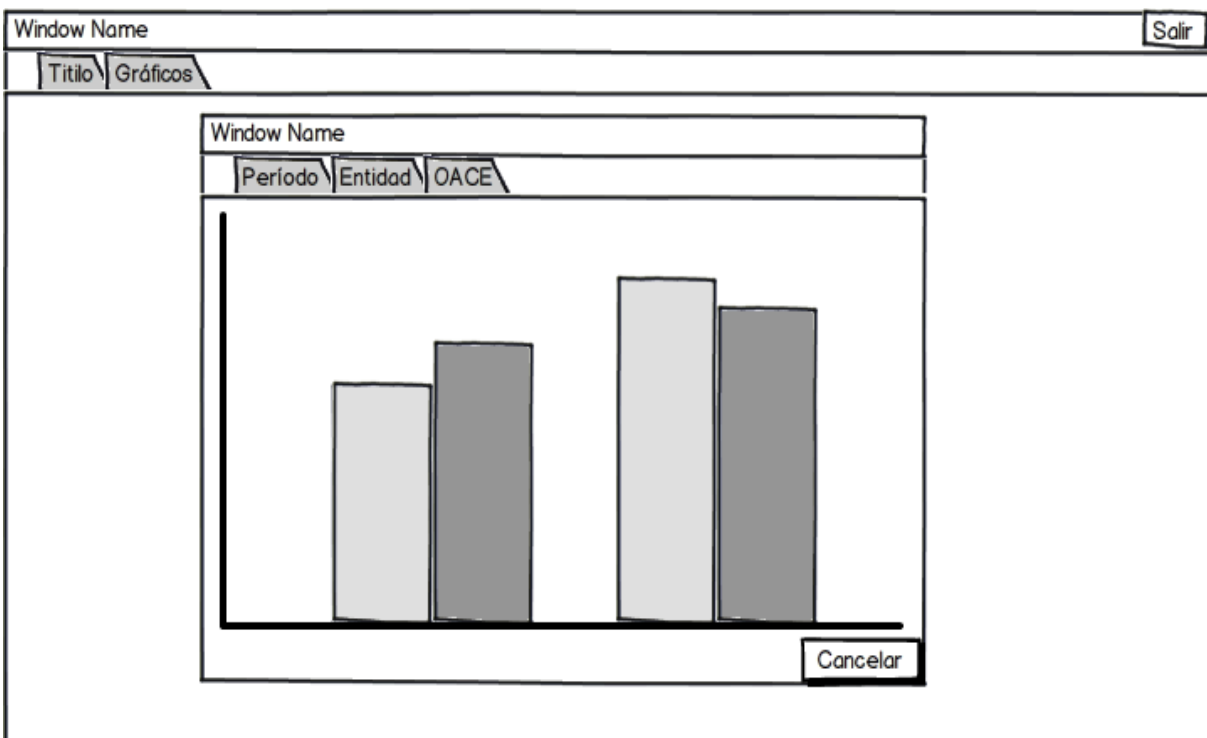


Figura 7 Prototipo de interfaz de usuario Gráfico.



Capítulo 2. Descripción de la solución propuesta

Adicionar Autorización [X]

Empresas

Entidad [OACE]

Localización

Provincia [▼] Municipio [▼]

Group Name

Tipo [▼] Tiempo [▼] Año [▲▼]

Estado 1 Estado 1

Estado 1 Estado 1

[Adicionar] [Cancelar]

Figura 8 Prototipo de interfaz de usuario Adicionar Autorización.

Adicionar Autorización [X]

Datos Primarios | Objetivos

Empresas

Entidad [OACE] Provincia [▼] Municipio [▼]

Tipo [▼] Estado [▼] Esfera [▼] Trimestre [▼]

Infracciones [▲▼] Medidas Controladas [▲▼] Medidas no Controladas [▲▼]

[Adicionar] [Cancelar]

Figura 9 Prototipo de interfaz de usuario Insertar Autorización.

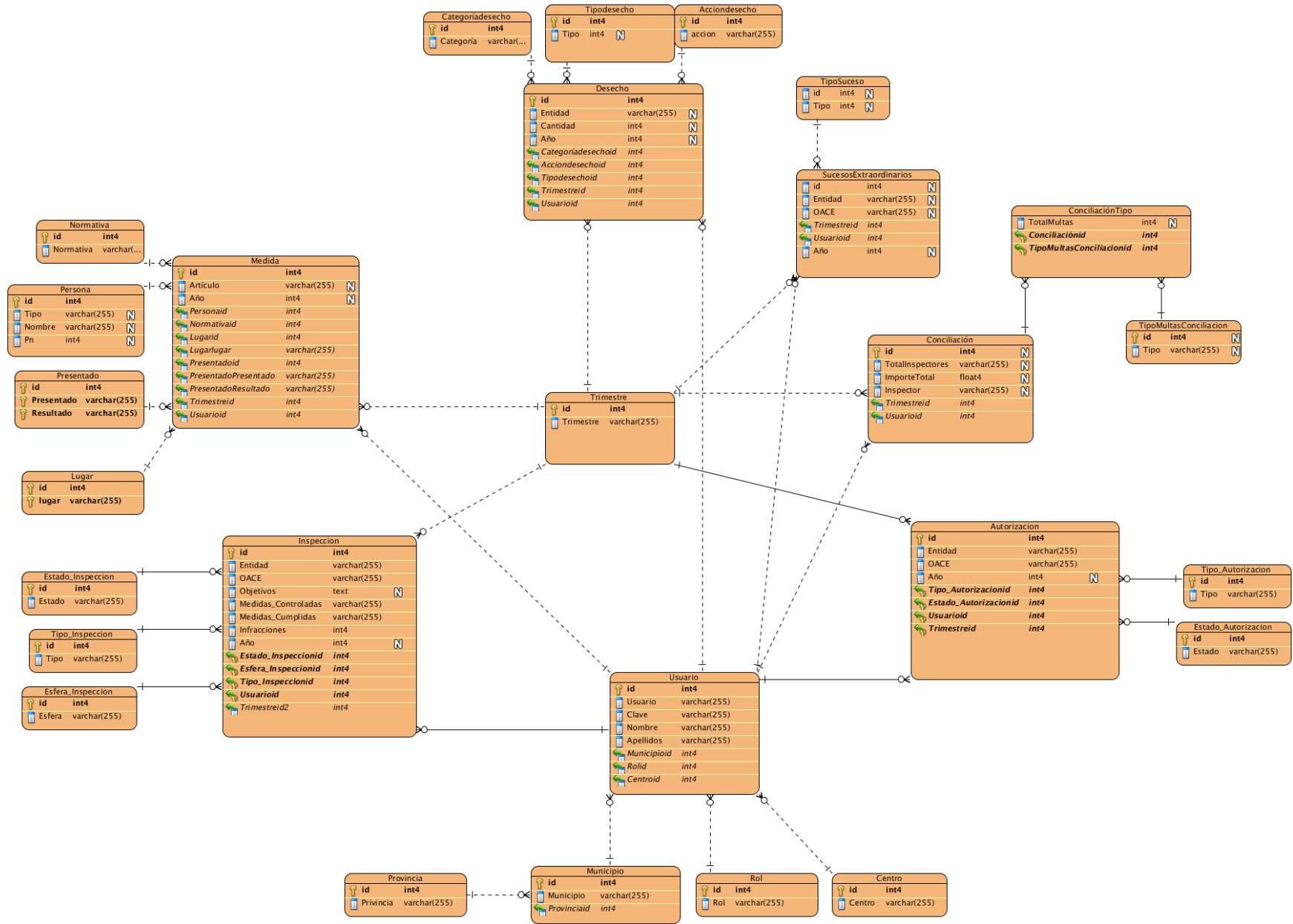


Capítulo 2. Descripción de la solución propuesta

Diagrama de Base de Datos, Modelo Entidad- Relación

El diagrama de base de datos no está definido en la metodología XP pero para detallar la forma de almacenamiento de la información es necesario realizar el modelo de datos, el cual permite determinar la estructura lógica, visualizar las entidades y relaciones existentes entre ellas. En la siguiente figura se muestra el modelo entidad relación de la base de datos:

Capítulo 2. Descripción de la solución propuesta





Capítulo 3. Análisis de resultados

Tabla 12 Descripción de las tablas de la base de datos.

Nombre de tabla	Descripción
Usuario	Almacena datos referentes a los usuarios del sistema como: usuario, clave, nombre y apellidos.
Inspeccion	Guarda datos como: entidad inspeccionada, objetivos a inspeccionar, medidas controladas, medidas cumplidas e infracciones cometidas, de las empresas que fueron sometidas a inspecciones.
Medida	Registra información referente a medidas aplicadas a empresas o personas, se registran datos como: artículos violados, normativas y lugar.
Desecho	Almacenamiento de datos con respecto a los desechos, tanto generados como manipulados por las empresas, además se registra su cantidad.
SucesoExtraordinario	Almacena información sobre los sucesos declarados extraordinarios, los datos almacenados están organizados en: entidad, tipo, año y trimestre.
Conciliacion	Guarda datos como: tipos de conciliaciones, total de inspectores y el importe total a pagar.
Autorizacion	Autorizaciones ambientales registradas por entidad, tipo, año y trimestre.

Conclusiones

La propuesta descrita en el presente capítulo toma como punto de partida la información enviada por los centros nacionales adscritos a la ORASEN, los que la recopilan de sus sucursales provinciales. Esta propuesta se basa en dos componentes fundamentales desde el punto de vista de la funcionalidad: la gestión de la información a partir de los datos que se introducen en el sistema y el análisis que se realiza sobre los mismos, lo cual constituye el mayor aporte de la investigación desarrollada.



Capítulo 3. Análisis de resultados

En este capítulo se describen además los elementos necesarios para la implementación de la aplicación:

- Las funcionalidades a implementar quedando redactadas 17 historias de usuarios, así como varios requisitos que debía cumplir el sistema desde el punto de vista no funcional.
- El diseño de la aplicación como principal entrada a la implementación de la propuesta. Se realizó el modelo de datos que consta de veintisiete tablas. Se describió la arquitectura, en este caso la arquitectura en 3 capas y las ventajas de su utilización.
- Las herramientas de visualización de la información definidas (gráficos de barra) así como los reportes y la información a mostrar en ellos que facilitarán la presentación de la información registrada en la base de datos.

La propuesta presentada muestra el papel que pueden jugar las tecnologías de la información en el control y seguimiento de procesos medioambientales, como herramienta de apoyo para empresas que realizan este tipo de procesos.



Capítulo 3 Implementación y prueba del sistema

En este capítulo se desarrollarán elementos esenciales de la metodología XP: la fase de desarrollo o implementación y la fase de pruebas. Dentro de estas fases quedará definido el estándar de codificación que se utilizará en la implementación, las tareas de programación, las pruebas de aceptación y las pruebas unitarias, así como los resultados que arrojaron dichas pruebas.

3.1. Implementación del sistema

La fase de desarrollo o fase de implementación del sistema propone seguir una programación en parejas, utilizar estándares de codificación además de limitar el tiempo de trabajo hasta 40 horas semanales. Debido a su importancia, se decide realizar las tareas de programación y definir cuál será el estándar de codificación a seguir.

Tareas de programación

Las tareas de programación son actividades que se elaboran para simplificar la programación de una historia de usuario, una historia de usuario puede tener 1 o más tareas de programación. Estas tareas son asignadas a los programadores para ser desarrolladas en la iteración que le corresponda. Las tareas de programación gracias a su flexibilidad admiten cambios. A continuación quedan definidas las tareas de programación por cada historia de usuario.

Tabla 13 Tareas de programación por Historias de usuario.

Historia de usuario	Estimación de esfuerzo (semanas)
Gestionar usuario	Crear Usuario Modificar usuario Eliminar usuario
Autenticar usuario	Autenticar usuario
Gestionar autorización	Crear autorización Modificar autorización Eliminar autorización
Mostrar tipo de autorización	Mostrar tipo de autorización
Gestionar inspección	Crear Inspección



Capítulo 3. Análisis de resultados

	Modificar inspección Eliminar inspección
Mostrar tipo de inspección	Mostrar tipo de inspección
Mostrar estado de la inspección	Mostrar estado de la inspección
Gestionar medidas	Crear medidas Modificar medidas Eliminar medidas
Gestionar resultado de medidas	Mostrar resultados de la medida
Mostrar normativa de la medida	Mostrar normativa de la medida
Gestionar desechos	Crear desechos Modificar desechos Eliminar desechos
Mostrar categoría de desechos	Mostrar categoría de desechos
Generar reporte	Generar reporte
Gestionar suceso extraordinario	Crear sucesos extraordinarios Modificar sucesos extraordinarios Eliminar sucesos extraordinarios
Mostrar tipo de sucesos extraordinarios	Mostrar sucesos extraordinarios
Gestionar conciliaciones	Crear conciliaciones Modificar conciliaciones Eliminar conciliaciones
Generar gráficos	Crear gráficos

Tabla 14 Tarea de programación 1 Generar reporte para la Historia de usuario Generar reporte

Tarea: Generar reporte	
Número tarea: HU 13-1	Historia de usuario: HU 13 Generar reporte
Nombre de la tarea: Generar reporte	



Capítulo 3. Análisis de resultados

Tipo de tarea: Desarrollo	
Fecha inicio: 24/02/2014	Fecha fin: 28/02/2014
Programador responsable: Edry Garcia Ramos – Gisela Sagarra Mendoza	
Descripción: Recibe como parámetros el trimestre y el año, la aplicación debe mostrar un archivo de extensión .pdf con el reporte.	

Tabla 15 Tarea de programación 1 Crear conciliación para la Historia de usuario Gestionar Conciliación.

Tarea: Gestionar conciliación	
Número tarea: HU 16-1	Historia de usuario: HU 16 Gestionar conciliación
Nombre de la tarea: Crear conciliaciones	
Tipo de tarea: Desarrollo	
Fecha inicio: 23/4/2014	Fecha fin: 25/4/2014
Programador responsable: Edry Garcia Ramos – Gisela Sagarra Mendoza	
Descripción: Recibe como parámetros todos los datos de la conciliación.	



Capítulo 3. Análisis de resultados

Tabla 16 Tarea de programación 2 Modificar conciliación para la Historia de usuario Gestionar conciliación.

Tarea: Gestionar conciliación	
Número tarea: HU 16-2	Historia de usuario: HU 16 Gestionar conciliación
Nombre de la tarea: Modificar conciliación	
Tipo de tarea: Desarrollo	
Fecha inicio: 28/4/2014	Fecha fin: 29/4/2014
Programador responsable: Edry Garcia Ramos – Gisela Sagarra Mendoza	
Descripción: Recibe como parámetros la conciliación que se quiera modificar, se introducen los datos y se modifican.	

Tabla 17 Tarea de programación 3: Eliminar conciliación para la Historia de usuario Gestionar conciliación.

Tarea: Gestionar conciliación	
Número tarea: HU 16-3	Historia de usuario: HU 16 Gestionar conciliación
Nombre de la tarea: Eliminar conciliación	
Tipo de tarea: Desarrollo	



Capítulo 3. Análisis de resultados

Fecha inicio: 29/03/2014	Fecha fin: 30/03/2014
Programador responsable: Edry Garcia Ramos – Gisela Sagarra Mendoza	
Descripción: Se selecciona la conciliación a eliminar y se elimina	

Estándares de codificación

Los estándares de codificación se centran en proveer una estructura de código uniforme que posibilite a los programadores seguir ciertos patrones de programación ya sean definidos por el equipo o la organización, estos mantienen el código legible.

Nombre de las variables

Todas las variables están escritas en minúscula.

Ejemplo: var medida;

Nombre de las clases y métodos

Los nombres de las clases y métodos están escritos de la misma manera: la primera letra de cada palabra en mayúscula, el resto de la palabra en minúscula. En el caso de las palabras compuestas por la letra inicial de cada palabra se escriben en mayúscula.

Ejemplo método: EditAuto:function(){ (...)}.

Comentarios del código

Los comentarios están escritos de la siguiente manera, comienzan con `/*` a continuación se escribe la aclaración que se desee, se termina con `*/`.

Ejemplo: `/* Texto del comentario*/`

Módulos de la solución propuesta

Módulo Administración

El módulo Administración permite la gestión de los elementos a los que tienen accesos solo los administradores del sistema como son: administrar los usuarios y los roles.



Capítulo 3. Análisis de resultados

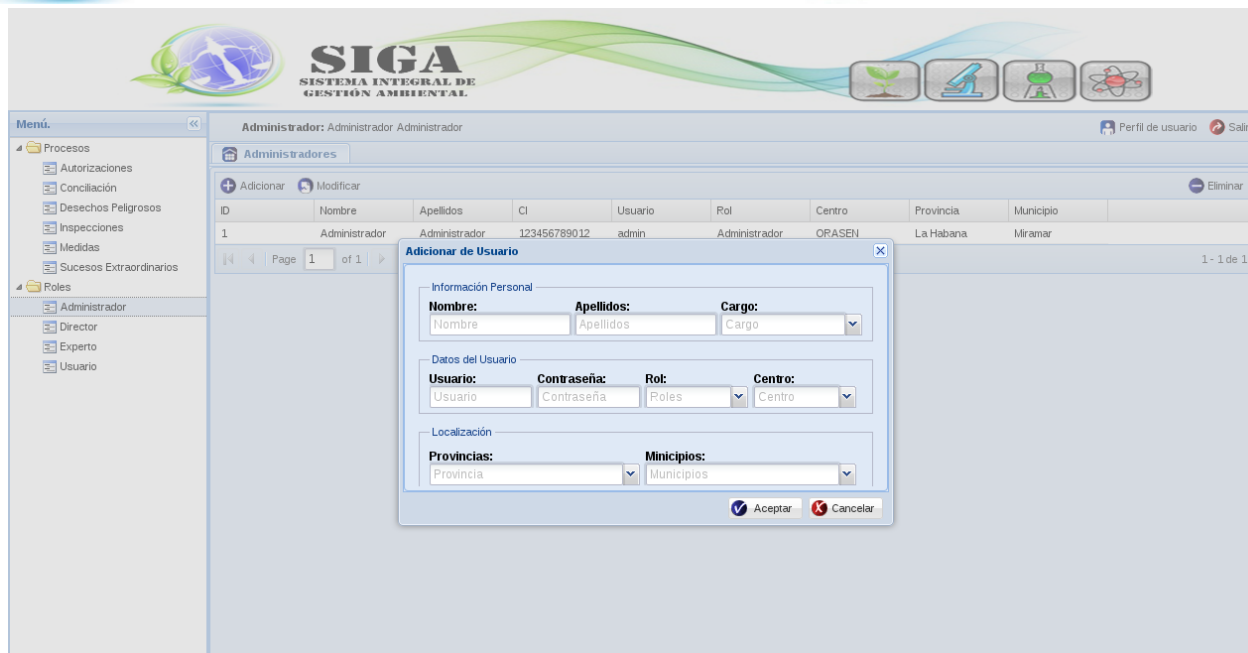


Figura 10 Módulo administración.

La gestión de usuarios constituye un elemento de apoyo a los procesos de la aplicación, se incluye dentro de los requisitos de seguridad del sistema, de esta manera se tendrá control sobre la información a la que cada usuario tendrá acceso según su rol.

La aplicación tiene definidos cuatro roles: administrador, director, experto y usuario, que podrán ser establecidos para las cuentas de usuario cuando sean creadas en el sistema.

Módulo Procesos

Este módulo permite la gestión de los procesos administrados por la oficina tales como: autorizaciones, inspecciones, medidas, sucesos extraordinarios, conciliaciones y desechos peligrosos.



Capítulo 3. Análisis de resultados



SIGA
SISTEMA INTEGRAL DE
GESTIÓN AMBIENTAL



Administrador: Administrador Administrador

Perfil de usuario | Salir

Autorizaciones | Gráficos

Adicionar | Modificar | Reporte

Entidad	OACE	Provincia	Municipio	Tipo	Estado	Trimestre	Año	Elaborada por:	Revisada por:	Aprobada por:
1 MINISTERIO D...	MINISTERIO D...	La Habana	Plaza de la R...	Licencias Am...	Proceso	Enero-Marzo	2014	Administrador	Administrador	
2 MINISTERIO D...	MINISTERIO D...	La Habana	Plaza de la R...	Licencias Am...	Proceso	Abril-Junio	2014	Administrador	Administrador	
3 EMPRESA DE...	EMPRESA DE...	La Habana	Plaza de la R...	Licencias Am...	Proceso	Julio-Septiem...	2014	Administrador		
4 EMPRESA DE...	EMPRESA DE...	La Habana	Plaza de la R...	Permisos de ...	Proceso	Octubre-Dici...	2014	Administrador		
5 EMPRESA PR...	EMPRESA PR...	La Habana	Plaza de la R...	Permisos de ...	Proceso	Enero-Marzo	2014	Administrador		
6 EMPRESA DE...	EMPRESA DE...	La Habana	Plaza de la R...	Licencias Am...	Proceso	Octubre-Dici...	2014	Administrador		
7 EMPRESA CA...	EMPRESA CA...	La Habana	Plaza de la R...	Permisos de ...	Proceso	Enero-Marzo	2014	Administrador		
8 EMPRESA GR...	EMPRESA GR...	La Habana	Plaza de la R...	Licencias Am...	Proceso	Abril-Junio	2014	Administrador		

Page 1 of 1 | 1 - 8 de 8

Figura 11 Módulo procesos

Módulo Análisis de datos

Este módulo permite la generación de tablas y gráficos de la información de los procesos insertada en el sistema para un posterior análisis.

Administrador: Administrador Administrador

Perfil de usuario | Salir

Autorizaciones | Gráficos

Adicionar | Modificar | Reporte

Ver Reporte

MINISTERIO DE CIENCIA TECNOLOGIA Y MEDIO AMBIENTE (CITMA)
Sistema de Información

AUTORIZACIONES DE LA ACTIVIDAD REGULADORA AMBIENTAL

Informe del periodo:
Trimestre: Enero-Marzo
Año: 2014

Modelo No.
Página 1 de 1
TRIMESTRAL
Unidad de medida: ENTREGAS

Denominación del informante: | Código del informante:

Solicitudes Recibidas					Estado			
Entidad	OACE	Prov.	Mpio.	Tipo de Autorización	Concedida	Denegada	En proceso	Descontinuadas.
A	B	C	D	E	F	G	H	I
EMPRESA CARNICA	EMPRESA CARNICA	La Habana	Plaza de la Revolución	Permisos de Desechos Peligrosos			X	
EMPRESA PROVINCIAL DE SERVICIOS AUTOMOTORES	EMPRESA PROVINCIAL DE SERVICIOS AUTOMOTORES	La Habana	Plaza de la Revolución	Permisos de Desechos Peligrosos			X	
MINISTERIO DE INDUSTRIAS	MINISTERIO DE INDUSTRIAS	La Habana	Plaza de la Revolución	Licencias Ambientales			X	
Total					0	0	3	0

Nombre y apellidos: | Cargo: | Fecha: | Firma:

Exportar Reporte | Cancelar



Capítulo 3. Análisis de resultados

Figura 12 Módulo Análisis de datos



Figura 13 Módulo Análisis de datos1

3.2. Pruebas al sistema

Según la bibliografía consultada la metodología utilizada requiere que se realicen pruebas al sistema, así se puede asegurar que el producto tenga éxito al realizar su entrega. Las pruebas de aceptación y pruebas unitarias serán definidas en esta sección. Las pruebas tienen como objetivo verificar que los requisitos que tiene el sistema sean los deseados.

Pruebas de aceptación

La metodología XP propone que para demostrar un buen funcionamiento del sistema es necesario realizarle pruebas al mismo, la ejecución de estas permite saber si el producto no presenta errores y si es lo que el cliente esperaba. El proceso de prueba termina cuando no se puede pensar en ninguna prueba que pueda causar falla alguna en la aplicación. Cuando se realizan las pruebas debe tener en cuenta la objetividad y deben ser realizadas desde el punto de vista del cliente, es por eso que se recomienda que se realicen por otra persona que no sea el programador.



Tabla 18 Caso de Prueba de Aceptación Generar reporte

Caso de Prueba de Aceptación Generar reporte

Número HU: 13

Número: 1

Nombre: Generar Reporte.

Descripción: Prueba de la funcionalidad Generar reporte.

Condiciones de ejecución: El usuario debe estar autenticado con el rol Director.

Pasos de ejecución: En la pestaña Reportes del menú principal se selecciona la opción Inspección o Autorización en dependencia del reporte que se quiera generar, luego en la nueva ventana que el sistema genera se selecciona el trimestre y el año del que se desea ver el reporte y se presiona el botón aceptar.

Resultado esperado: Se muestra un formulario con el reporte.

Evaluación de la prueba: Prueba satisfactoria.

Tabla 19 Caso de Prueba de Aceptación Adicionar autorización.

Caso de Prueba de Aceptación Adicionar autorización

Número HU: 3

Número:1

Nombre: Gestionar autorización.

Descripción: Prueba de la funcionalidad Adicionar autorización.

Condiciones de ejecución: Debe existir un usuario autenticado.

Pasos de ejecución: En la pestaña Procesos del menú principal se selecciona la opción Autorizaciones, luego en la nueva ventana que el sistema genera, el usuario acciona el botón adicionar. En la nueva ventana se llenan todos los datos y se presiona el aceptar.



Capítulo 3. Análisis de resultados

Resultado esperado: Se adiciona la autorización

Evaluación de la prueba: Prueba satisfactoria.

Tabla 20 Caso de Prueba de Aceptación Modificar autorización.

Caso de Prueba de Aceptación Modificar autorización

Número HU: 3

Número:2

Nombre: Gestionar autorización.

Descripción: Prueba de la funcionalidad Modificar autorización.

Condiciones de ejecución: Debe existir un usuario autenticado.

Pasos de ejecución: En la pestaña Procesos del menú principal se selecciona la opción Autorizaciones, luego en la nueva ventana que el sistema genera, el usuario selecciona la autorización que desea modificar y acciona el botón modificar. En la nueva ventana se modifica los datos que se requieran y se presiona el botón aceptar.

Resultado esperado: Se modifica la autorización

Evaluación de la prueba: Prueba satisfactoria.

Tabla 21 Caso de Prueba de Aceptación Eliminar autorización.

Caso de Prueba de Aceptación Eliminar autorización

Número HU: 3

Número:3

Nombre: Gestionar autorización.

Descripción: Prueba de la funcionalidad Eliminar autorización.

Condiciones de ejecución: Debe existir un usuario autenticado.

Pasos de ejecución: En la pestaña Procesos del menú principal se selecciona la



Capítulo 3. Análisis de resultados

opción Autorizaciones, luego en la nueva ventana que el sistema genera, el usuario selecciona la autorización que desea eliminar y acciona el botón eliminar. En la nueva ventana se selecciona el botón aceptar.

Resultado esperado: Se elimina la autorización

Evaluación de la prueba: Prueba satisfactoria.

Las pruebas de aceptación se realizaron en cuatro iteraciones arrojando los siguientes resultados:

- En la primera iteración se detectaron cinco no conformidades significativas cuatro no significativas y ocho recomendaciones.
- En la segunda iteración se obtuvo tres no conformidades de tipo significativa, cinco de tipo no significativa y seis recomendaciones.
- En la tercera iteración solo se obtuvo cuatro recomendaciones.
- En la cuarta iteración se detectaron dos no conformidades significativas y tres recomendaciones.

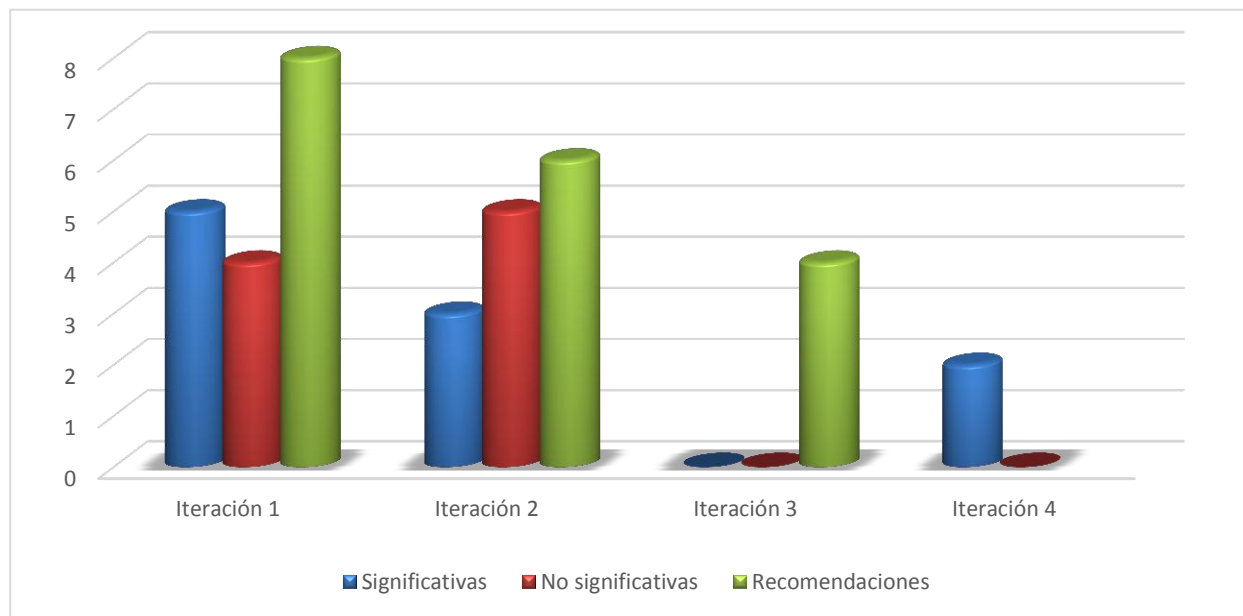
Las diez no conformidades significativas se centraron en errores en campos de selección, validaciones no realizadas e información incorrecta devuelta en campos para reportes, la corrección de estas no conformidades acarrearán modificaciones en métodos incorrectos. Uno de los errores encontrados es que al seleccionar una provincia no aparecen los municipios que le corresponden a esta. Se le dio solución agregando dichos municipios a la base de datos.

Las nueve no conformidades no significativas detectadas en su mayoría correspondieron a errores ortográficos en la interfaz de usuario, ejemplo el mensaje “¿Debe seleccionar una autorización de la tabla?” no presentaba el acento de la palabra “autorización” y se omitió el signo de interrogación del inicio. Se encontraron errores ortográficos en algunos campos con información cargada de la base de datos, para corregir los mismos se hizo un análisis de toda la información insertada.



Capítulo 3. Análisis de resultados

Las 21 recomendaciones se centraron en uso de colores, organización de la información en la aplicación e inclusión de algunos elementos en los campos de selección. Todas las no conformidades fueron corregidas antes de comenzar la implementación de la siguiente iteración.



Gráfica 1 No conformidades significativas, no significativas y recomendaciones.

Pruebas unitarias

Las Pruebas Unitarias son una forma de hacer pruebas a pequeñas porciones del código de un programa, estas permiten encontrar problemas cometidos por los programadores en el proceso de desarrollo de las funcionalidades, este tipo de pruebas son automatizadas y se puede decir que una prueba es completa o buena si cumple con los siguientes puntos:

- Automatizable: No requiere la intervención humana en el proceso de prueba.
- Completas: Deben abarcar la mayor cantidad de funcionalidades posibles.
- Reutilizable: Se deben crear pruebas que puedan ser utilizadas en más de una ocasión.
- Profesionales: Deben ser consideradas parte del código de la aplicación por lo que deben ser implementadas con la misma profesionalidad.
- Independiente: Cada prueba debe ejecutarse sin interferir en las otras.



Capítulo 3. Análisis de resultados

Para el proceso de prueba de la solución se utilizó el *framework* PHPUnit desarrollado sobre el lenguaje PHP y que se encuentra perfectamente integrado con Symfony 2.3.

En la siguiente imagen se muestra un ejemplo de la ejecución de las pruebas a 7 funcionalidades las cuales arrojaron los siguientes resultados:

```
FAILURES!  
Tests: 7, Assertions: 4, Errors: 3.
```

Figura 14 Prueba unitaria 1

Luego de corregir los principales problemas encontrados en la iteración anterior se ejecutaron nuevamente las pruebas y se obtuvo un resultado satisfactorio.

```
Configuration read from /var/www/Symfony/app/phpunit.xml  
.....  
Time: 5 seconds, Memory: 18.75Mb  
OK (7 tests, 7 assertions)
```

Figura 15 Prueba unitaria 2

Conclusiones parciales

El análisis de los resultados obtenidos en la implementación del sistema, tiene como objetivo probar su aplicación con datos reales, además de identificar mejoras en el proceso de gestión de la información ambiental, las contribuciones sociales y económicas asociadas a su uso, concluyendo que:

- El Módulo Administración garantizará la protección de los datos y los accesos al sistema.
- El Módulo Procesos permitirá a los expertos gestionar toda la información relacionada con los aspectos fundamentales que se manejan en cada centro nacional adscrito a la ORASEN, ya sea: autorizaciones, inspecciones, conciliaciones, desechos peligrosos, medidas o sucesos extraordinarios, contribuyendo al control y seguimiento de dicha información.



Capítulo 3. Análisis de resultados

- El Módulo Reportes permitirá a los expertos, mediante la visualización de los datos por reportes o gráficas, tener un referente comparativo de los diferentes criterios y con esto favorecer a la toma de decisiones.

La utilización del sistema servirá de apoyo a los expertos de la ORASEN y de sus centros nacionales, beneficiará a la población cubana y a la economía del país pues su desarrollo con tecnologías libres evitará gastos por conceptos de licencias de software y equipamiento.



Conclusiones generales

Al término de la investigación se arriba a las siguientes conclusiones, evidenciando el cumplimiento de los objetivos propuestos:

- El análisis bibliográfico muestra que los procesos medioambientales controlados por la ORASEN, por ser una temática propia de esta oficina y de los centros adscritos a ella, presentan una terminología en ocasiones particular de sus procesos, pero los mismos se encuentran documentados en un grupo de leyes, decretos y decretos leyes dictados por la dirección del ministerio y del país, que regulan el actuar de las partes en dichos procesos.
- El estudio del estado del arte muestra la inexistencia de una herramienta que contribuya al control y seguimiento de los procesos medioambientales que realiza la ORASEN, además que existen insuficiencias de las herramientas similares existentes para poder ser utilizadas en esta oficina. Se concluye además en la necesidad de desarrollar una aplicación web, utilizando tecnologías libres, que contribuya a la mejora de la problemática planteada en la investigación.
- La propuesta descrita se basa en dos componentes fundamentales desde el punto de vista de la funcionalidad: la gestión de la información a partir de los datos que se introducen en el sistema y el análisis que se realiza sobre los mismos, lo cual constituye el mayor aporte de la investigación desarrollada.

Se describen además los elementos necesarios para la implementación de la aplicación:

- El modelo conceptual para el acercamiento al dominio del problema, identificándose 14 conceptos.
- Las funcionalidades a implementar quedando redactadas 17 historias de usuarios, así como varios requisitos que debía cumplir el sistema desde el punto de vista no funcional.
- El diseño de la aplicación como principal entrada a la implementación de la propuesta. Se realizó el modelo de datos que consta de 27 tablas normalizadas. Además se describió la arquitectura, en este caso la arquitectura en tres capas y las ventajas de su utilización.



Conclusiones generales

- Las herramientas de visualización de la información definidas: gráficos y reportes, así como la información a mostrar en ellos que facilitarán la presentación de la información registrada en la base de datos.

- El análisis de los resultados obtenidos en la implementación del sistema, tiene como objetivo probar su aplicación con datos reales, además de identificar mejoras en el proceso de gestión de la información ambiental, las contribuciones sociales y económicas asociadas a su uso, concluyendo que:
 - El Módulo Administración garantizará la protección de los datos y los accesos al sistema.
 - El Módulo Procesos permitirá a los expertos gestionar toda la información relacionada con los aspectos fundamentales que se manejan en cada centro nacional adscrito a la ORASEN, ya sea: autorizaciones, inspecciones, conciliaciones, desechos peligrosos, medidas o sucesos extraordinarios, contribuyendo al control y seguimiento de dicha información.
 - El Módulo Reportes permitirá a los expertos, mediante la visualización de los datos por reportes o gráficas, tener un referente comparativo de los diferentes criterios y con esto favorecer a la toma de decisiones.

- La utilización del sistema servirá de apoyo a los expertos de la ORASEN y de sus centros nacionales, beneficiará a la población cubana y a la economía del país pues su desarrollo con tecnologías libres evitará gastos por conceptos de licencias de software y equipamiento.



Recomendaciones

Recomendaciones

- Aumentar el número de reportes que puedan ser de utilidad a los especialistas de la ORASEN.
- Extender el uso de la aplicación por todo el país donde la tecnología lo permita.



Bibliografías

1. Castro Ruz. Discurso pronunciado en la Velada Solemne con Motivo del XXV Aniversario del Triunfo de la Revolución y la Entrega del Título Honorífico de “Héroe de la Republica de Cuba” y la Orden “Antonio Maceo” a la Ciudad de Santiago de Cuba. [online]. 1 January 1984. Santiago de Cuba. Available from: <http://www.cuba.cu/gobierno/discursos/1984/esp/f010184e.html>
2. Parrado Alvarez Oscar L. Concepción acerca del medioambiente y la dimensión ambiental orientada al desarrollo sostenible y su repercusión en la formación de los profesionales. [online]. P. 25. Available from: http://www.ucp.cm.rimed.cu/uproduct/italler/htdocs/docs/panel_preevent/2/oscar-l-parrado.pdf
3. Desarrollo sostenible. EcuRed [online]. Available from: http://www.ecured.cu/index.php/Desarrollo_sostenible
4. Ministerio de Ciencia Tecnología y Medio Ambiente. [online]. Available from: <http://www.medioambiente.cu/organigrama.asp#top>
5. Gestión de la información. [online]. [no date]. Available from: http://www.ecured.cu/index.php/Gesti%C3%B3n_de_la_Informaci%C3%B3n
6. Sonora. introduccion sistemas. [online]. Available from: http://biblioteca.itson.mx/oa/dip_ago/introduccion_sistemas/p3.htm.
7. Medio Ambiente. [online]. Available from: <http://www.medioambiente.cu/>
8. Instituto Nacional de Ecología. [online]. Available from: <http://www2.inecc.gob.mx/publicaciones/libros/47/cap2.html>
9. Legislación Ambiental. Ingeniería y consultoría ambiental [online]. Available from: <http://www.ecolaningenieria.com/ingenieria-ambiental/legislacion-ambiental>
10. Manuel Antonio Sánchez Zuraty. Diccionario Básico de Derecho. Quito, Ecuador : Editorial Jurídica de Ecuador, 2000.



11. Guillermo Caballenas de Torre. Diccionario Jurídico Elemental. [no date].
12. Castro Ortiz. El principio de proporcionalidad en la Aplicación de las penas y las contravenciones muy graves en el Juzgado primero de Tránsito de la Provincia de Tungurahua. [online]. Ecuador : Universidad Técnica de Ambato,, 2013. Available from: <http://repo.uta.edu.ec/handle/123456789/6285>
13. Ecosystem. [online]. Available from: <http://www.aplicacionesgml.com/Ecosystem/default.aspx>
14. Ekomanager. [online]. Available from: <http://www.aplicacionesgml.com/Ecosystem/default.aspx>
15. Ing. Andrés Orlando González Hernández. SISTEMA DE Información AMBIENTAL para la EMPRESA DE PRODUCTOS LÁCTEOS de VILLA CLARA. Santa Clara, Villa Clara, Cuba : UNIVERSIDAD CENTRAL "MARTA ABREU" DE LAS VILLAS FACULTAD DE MATEMÁTICA, FÍSICA Y COMPUTACIÓN, [no date].
16. Conceptos básicos: funcionamiento de la programación. Microsoft Developer Network MSDN [online]. Available from: <http://msdn.microsoft.com/es-es/library/ms172579%28v=vs.80%29.aspx>
17. Ureña Almagro Carlos. Lenguajes de Programación [online]. Universidad de Granada, España, 2011. Available from: <http://lsi.ugr.es/curena/doce/lp/tr-11-12/lp-c01-impr.pdf>
18. PHP. Hypertext Preprocessor [online]. Available from: <http://php.net>
19. Rubén Alvarez. Inicio de la programación en ASP. Desarrollo web.com [online]. Available from: <http://www.desarrolloweb.com/articulos/247.php>
20. Qué es PHP. Desarrollo web, tu mejor ayuda para aprender ha hacer webs [online]. Available from: <http://www.desarrolloweb.com/articulos/392.php>
21. Java Script. Sun Microsystems [online]. Available from: <http://www.sun.com/suntrademarks/>



22. Eguíluz Pérez. Introducción a JavaScript [online]. [no date]. Available from: <http://www.librosweb.es/javascript/>.
23. Definición ABC. [online]. Available from: <http://www.definicionabc.com/tecnologia/html.php>
24. OTROS, Adams, Cameron. The Art & Science Of CSS. 2007. [no date].
25. KENNEDY, BILL, Musciano, Chuck. HTML & XHTML: The Definitive Guide. s.l.: O'Reilly. [no date].
26. Lab Ellis. CodeIgniter. [online]. Available from: <http://ellislab.com/codeigniter>
27. Potencier. Referencia de licencia. [online]. 2004. Available from: <http://symfony.com/license>
28. Propel ORM. [online]. Available from: <http://www.propelorm.org>
29. Doctrine Project. [online]. Available from: www.doctrine-project.org
30. John Resig. JQuery. [online]. Available from: <http://jquery.com/>
31. STEVE BLADES, Shea Frederick and Colin Ramsay. Learnig Ext JS. Uk : Published by Packt Publishing Ltd. 32 Lincoln Road Olton Birmingham, B27 6PA, UK., 2008. ISBN 978-1-847195-14-2.
32. Sencha. Ext JS 4.2. [online]. Available from: <http://www.sencha.com/>
33. Garcia. Ext JS in Action. MEAP Edition Manning Early Access Program, 2009.
34. RUMBAUGH, JAMES., Jacobson, Ivar and BOOCH, GRADY. El proceso unificado de desarrollo de software. Madrid, [no date]. ISBN 9788478290369.
35. SCRUM. [online]. Available from: [ttp://scrum-ing-software.blogspot.com/](http://scrum-ing-software.blogspot.com/)
36. SCRUM 1. [online]. Available from: <http://www.proyectosagiles.org/que-es-scrum>



37. Optimus Software y Consultoria C.A. [online]. Available from: <http://www.optimus-software.com/noticias/2011/11/28/desventajas-del-scrum/>
38. SIERRA, ALEJANDRO, Fowler, Martin. La Nueva Metodología. [online]. Available from: <http://www.programacionextrema.org/articulos/newMethodology.es.html>.
39. Sommerville. Ingeniería del Software. Madrid, 2005. ISBN 84-7829-074-5.
40. INTECO. [online]. Available from: <http://es.scribd.com/doc/62931905/45/Extreme-Programming-XP>.
41. JOSE ARTURO MORA SOTO, MORA SOTO, J. A. Herramientas UML... ¿cuál utilizar? [online]. Available from: <http://www.jams.name/2010/04/18/herramientas-uml-cual-utilizar/>.
42. VP. Visual Paradigm [online]. [no date]. Available from: http://www.visual-paradigm.com/support/documents/vpumuserguide/12/13/7572_licensing.html.
43. Grupo de Soluciones Innova. Rational Rose. [online]. Available from: <http://www.rational.com.ar/herramientas/roseenterprise.html>.
44. MySQL. The world's most popular open source database. [online]. Available from: <http://www.mysql.com/products/enterprise/database/>.
45. PostgreSQL. [online]. Available from: http://www.postgresql.org/es/sobre_postgresql.
46. PostgreSQL. [online]. Available from: <http://www.postgresql.org/about/>.
47. Zend: The php company. [online]. Available from: <http://www.zend.com/en/products/studio/>.
48. NetBeans IDE - The Smarter and Faster Way to Code. NetBeans [online]. 2013. Available from: <https://netbeans.org>
49. Información del servidor web IIS. [online]. Available from: <http://technet.microsoft.com/es-es/library/hh831725.aspx>
50. Apache. [online]. Available from: <http://httpd.apache.org/>



51. GNU Image Manipulation Program [online]. Proyecto GNU, [no date]. Available from: <http://www.gimp.org/>
52. JEFFRIES, R., ANDERSON, A., HENDRICKSON, C., Addison-Wesley. Extreme Programming Installed. 2001.
53. ADDISON-WESLEY. "Extreme Programming in Practice". 2001.
54. A laboratory for teaching object-oriented thinking. [online]. Available from: <http://c2.com/doc/oopsla89/paper.html#cards>
55. Metodología XP. [online]. Available from: <https://sites.google.com/site/xpmetodologia/marco-teorico/funcionamiento>
56. Craig Larman. UML y Patrones Introducción al análisis y diseño orientado a objetos. [no date].
57. Roger S. Pressman. Ingeniería de Software Un enfoque práctico. [no date].



Anexo 1 Historias de usuario

Tabla 22 Historia de usuario Gestionar usuario

Historia de usuario	
No. 1	Nombre: Gestionar usuario
Usuario: Administrador	
Modificación de HU: Ninguna	
Programador: Gisela Sagarra Mendoza – Edry Garcia Ramos	
Prioridad de Negocio: Baja	Puntos Estimados: 1
Nivel de Complejidad: Baja	Iteración Asignada: 1
Descripción: Permite crear, modificar y eliminar usuarios en la aplicación	
Observaciones:	

Tabla 23 Historia de usuario Gestionar Inspección

Historia de usuario	
No. 5	Nombre: Gestionar inspección
Usuario: Usuario, Director, Administrador, Experto	
Modificación de HU: Ninguna	
Programador: Gisela Sagarra Mendoza – Edry Garcia Ramos	
Prioridad de Negocio: Alta	Puntos Estimados: 1
Nivel de Complejidad: Baja	Iteración Asignada: 2
Descripción: Permite crear, modificar y eliminar inspecciones en la aplicación	
Observaciones:	

Tabla 24 Historia de usuario Gestionar desechos

Historia de usuario	
No. 11	Nombre: Gestionar desechos



Usuario: Usuario, Director, Administrador, Experto	
Modificación de HU: Ninguna	
Programador: Gisela Sagarra Mendoza – Edry Garcia Ramos	
Prioridad de Negocio: Alta	Puntos Estimados: 0.5
Nivel de Complejidad: Baja	Iteración Asignada: 2
Descripción: Permite crear, modificar y eliminar desechos en la aplicación	
Observaciones:	

Tabla 25 Historia de usuario Gestionar conciliaciones

Historia de usuario	
No. 16	Nombre: Gestionar conciliaciones
Usuario: Usuario, Director, Administrador, Experto	
Modificación de HU: Ninguna	
Programador: Gisela Sagarra Mendoza – Edry Garcia Ramos	
Prioridad de Negocio: Alta	Puntos Estimados: 1
Nivel de Complejidad: Baja	Iteración Asignada: 4
Descripción: Permite crear, modificar y eliminar conciliaciones en la aplicación	
Observaciones:	

Tabla 26 Historia de usuario Autenticar usuario

Historia de usuario	
No. 2	Nombre: Autenticar usuario
Usuario: Usuario, Director, Administrador, Experto	
Modificación de HU: Ninguna	
Programador: Gisela Sagarra Mendoza – Edry Garcia Ramos	
Descripción: Permite que el usuario se autentique para entrar en la aplicación	
Observaciones: El usuario tiene que existir en la aplicación	



Tabla 27 Historia de usuario Mostrar tipo de autorización

Historia de usuario	
No. 4	Nombre: Mostrar tipo de autorización
Usuario: Usuario, Director, Administrador, Experto	
Modificación de HU: Ninguna	
Programador: Gisela Sagarra Mendoza – Edry Garcia Ramos	
Prioridad de Negocio: Baja	Puntos Estimados: 0.5
Nivel de Complejidad: Baja	Iteración Asignada: 1
Descripción: Permite ver los tipos de autorizaciones existentes en la aplicación	
Observaciones: Deben existir datos en la aplicación	

Tabla 28 Historia de usuario Mostrar tipo de inspección

Historia de usuario	
No. 6	Nombre: Mostrar tipo de inspección
Usuario: Usuario, Director, Administrador, Experto	
Modificación de HU: Ninguna	
Programador: Gisela Sagarra Mendoza – Edry Garcia Ramos	
Prioridad de Negocio: Baja	Puntos Estimados: 0.5
Nivel de Complejidad: Baja	Iteración Asignada: 2
Descripción: Permite ver los tipos de inspecciones existentes en la aplicación	
Observaciones: Deben existir datos en la aplicación	

Tabla 29 Historia de usuario Mostrar estado de la inspección

Historia de usuario	
No. 7	Nombre: Mostrar estado de la inspección
Usuario: Usuario, Director, Administrador, Experto	
Modificación de HU: Ninguna	



Programador: Gisela Sagarra Mendoza – Edry Garcia Ramos	
Prioridad de Negocio: Baja	Puntos Estimados: 0.5
Nivel de Complejidad: Baja	Iteración Asignada: 2
Descripción: Permite ver los estados de las inspecciones existentes en la aplicación	
Observaciones: Deben existir datos en la aplicación	

Tabla 30 Historia de usuario Mostrar resultado de las medidas

Historia de usuario	
No. 9	Nombre: Mostrar resultado de las medidas
Usuario: Usuario, Director, Administrador, Experto	
Modificación de HU: Ninguna	
Programador: Gisela Sagarra Mendoza – Edry Garcia Ramos	
Prioridad de Negocio: Baja	Puntos Estimados: 0.5
Nivel de Complejidad: Baja	Iteración Asignada: 3
Descripción: Permite ver los resultados de las medidas existentes en la aplicación	
Observación:	

Tabla 31 Historia de usuario Mostrar normativa de la medida

Historia de usuario	
No. 10	Nombre: Mostrar normativa de la medida
Usuario: Usuario, Director, Administrador, Experto	
Modificación de HU: Ninguna	
Programador: Gisela Sagarra Mendoza – Edry Garcia Ramos	
Prioridad de Negocio: Baja	Puntos Estimados: 0.5
Nivel de Complejidad: Baja	Iteración Asignada: 3



Descripción: Permite ver las normativas de las medidas existentes en la aplicación
Observación:

Tabla 32 Historia de usuario Mostrar categoría de desechos

Historia de usuario	
No. 12	Nombre: Mostrar categoría de desechos
Usuario: Usuario, Director, Administrador, Experto	
Modificación de HU: Ninguna	
Programador: Gisela Sagarra Mendoza – Edry Garcia Ramos	
Prioridad de Negocio: Baja	Puntos Estimados: 0.5
Nivel de Complejidad: Baja	Iteración Asignada: 2
Descripción: Permite ver las categorías de los desechos existentes en la aplicación	
Observación:	

Tabla 33 Historia de usuario Mostrar tipo de suceso extraordinario

Historia de usuario	
No. 15	Nombre: Mostrar tipo de suceso extraordinario
Usuario: Usuario, Director, Administrador, Experto	
Modificación de HU: Ninguna	
Programador: Gisela Sagarra Mendoza – Edry Garcia Ramos	
Prioridad de Negocio: Baja	Puntos Estimados: 0.5
Nivel de Complejidad: Baja	Iteración Asignada: 4
Descripción: Permite ver los tipos de sucesos extraordinarios existentes en la aplicación	
Observación:	



Tabla 34 Historia de usuario Generar gráficos

Historia de usuario	
No. 17	Nombre: Generar gráficos
Usuario: Usuario, Director, Administrador, Experto	
Modificación de HU: Ninguna	
Programador: Gisela Sagarra Mendoza – Edry Garcia Ramos	
Prioridad de Negocio: Alta	Puntos Estimados: 0.5
Nivel de Complejidad: Baja	Iteración Asignada: 4
Descripción: Permite crear las gráficas para cada proceso	
Observación: La aplicación debe tener datos	

Anexo 2 Tareas de programación

Tabla 35 Tarea de programación 1 Crear usuario para la Historia de usuario Gestionar usuario.

Tarea: Gestionar usuario	
Número tarea: HU 1-1	Historia de usuario: HU 1 Gestionar usuario
Nombre de la tarea: Crear usuario	
Tipo de tarea: Desarrollo	
Fecha inicio: 2/12/2013	Fecha fin: 4/12/2013
Programador responsable: Edry Garcia Ramos – Gisela Sagarra Mendoza	
Descripción: Recibe como parámetros todos los datos del usuario que se quiere insertar en la aplicación	

Tabla 36 Tarea de programación 2 Modificar usuario para la Historia de usuario Gestionar usuario.

Tarea: Gestionar usuario	
Número tarea: HU 1-2	Historia de usuario: HU 1 Gestionar usuario



Nombre de la tarea: Modificar usuario	
Tipo de tarea: Desarrollo	
Fecha inicio: 4/12/2013	Fecha fin: 5/12/2013
Programador responsable: Edry Garcia Ramos – Gisela Sagarra Mendoza	
Descripción: Recibe como parámetros los datos del usuario que se quiere modificar en la aplicación	

Tabla 37 Tarea de programación 3 Eliminar usuario para la Historia de usuario Gestionar usuario.

Tarea: Gestionar usuario	
Número tarea: HU 1-3	Historia de usuario: HU 1 Gestionar usuario
Nombre de la tarea: Eliminar usuario	
Tipo de tarea: Desarrollo	
Fecha inicio: 5/12/2014	Fecha fin: 6/12/2014
Programador responsable: Edry Garcia Ramos – Gisela Sagarra Mendoza	
Descripción: Recibe el identificador de los datos del usuario que se quiere eliminar	

Tabla 38 Tarea de programación 1 Autenticar usuario para la Historia de usuario Autenticar usuario

Tarea: Autenticar usuario	
Número tarea: HU 2-1	Historia de usuario: HU 2 Gestionar usuario
Nombre de la tarea: Autenticar usuario	
Tipo de tarea: Desarrollo	
Fecha inicio: 9/12/2013	Fecha fin: 11/12/2013
Programador responsable: Edry Garcia Ramos – Gisela Sagarra Mendoza	
Descripción: Recibe como parámetros el usuario y la contraseña del usuario que quiere acceder a la aplicación	



Tabla 39 Tarea de programación 1 Crear autorización para la Historia de usuario Gestionar autorización

Tarea: Gestionar autorización	
Número tarea: HU 3-1	Historia de usuario: HU 3 Gestionar autorización
Nombre de la tarea: Crear autorización	
Tipo de tarea: Desarrollo	
Fecha inicio: 11/12/2013	Fecha fin: 13/12/2013
Programador responsable: Edry Garcia Ramos – Gisela Sagarra Mendoza	
Descripción: Recibe como parámetros todos los datos de la autorización que se quiere insertar en la aplicación	

Tabla 40 Tarea de programación 2 Modificar autorización para la Historia de usuario Gestionar autorización

Tarea: Gestionar autorización	
Número tarea: HU 3-2	Historia de usuario: HU 3 Gestionar autorización
Nombre de la tarea: Modificar autorización	
Tipo de tarea: Desarrollo	
Fecha inicio: 9/12/2013	Fecha fin: 10/12/2013
Programador responsable: Edry Garcia Ramos – Gisela Sagarra Mendoza	
Descripción: Recibe como parámetros los datos de la autorización que se quiere modificar en la aplicación	

Tabla 41 Tarea de programación 3 Eliminar autorización para la Historia de usuario Gestionar autorización

Tarea: Gestionar autorización	
Número tarea: HU 3-3	Historia de usuario: HU 3 Gestionar autorización
Nombre de la tarea: Eliminar autorización	
Tipo de tarea: Desarrollo	



Fecha inicio: 10/12/2013	Fecha fin: 11/12/2013
Programador responsable: Edry Garcia Ramos – Gisela Sagarra Mendoza	
Descripción: Recibe como parámetros el identificador de la autorización que se quiere eliminar en la aplicación	

Tabla 42 Tarea de programación 1 Mostrar tipo de autorización para la Historia de usuario
Mostrar tipo de autorización

Tarea: Mostrar tipo de autorización	
Número tarea: HU 4-1	Historia de usuario: HU 4 Mostrar tipo de autorización
Nombre de la tarea: Mostrar tipo de autorización	
Tipo de tarea: Desarrollo	
Fecha inicio: 11/12/2013	Fecha fin: 13/12/2013
Programador responsable: Edry Garcia Ramos – Gisela Sagarra Mendoza	
Descripción: Recibe como parámetros el identificador del tipo de la autorización que se quiere mostrar en la aplicación	

Tabla 43 Tarea de programación 1 Crear inspección para la Historia de usuario Gestionar inspección

Tarea: Gestionar inspección	
Número tarea: HU 5-1	Historia de usuario: HU 5 Gestionar inspección
Nombre de la tarea: Crear inspección	
Tipo de tarea: Desarrollo	
Fecha inicio: 3/2/2014	Fecha fin: 5/2/2014
Programador responsable: Edry Garcia Ramos – Gisela Sagarra Mendoza	
Descripción: Recibe como parámetros todos los datos de la inspección que se quiere insertar en la aplicación	



Tabla 44 Tarea de programación 2 Modificar inspección para la Historia de usuario Gestionar inspección

Tarea: Gestionar inspección	
Número tarea: HU 5-2	Historia de usuario: HU 5 Gestionar inspección
Nombre de la tarea: Modificar inspección	
Tipo de tarea: Desarrollo	
Fecha inicio: 5/2/2014	Fecha fin: 6/2/2014
Programador responsable: Edry Garcia Ramos – Gisela Sagarra Mendoza	
Descripción: Recibe como parámetros el identificador de la inspección que se quiere modificar en la aplicación	

Tabla 45 Tarea de programación 3 Eliminar inspección para la Historia de usuario Gestionar inspección

Tarea: Gestionar inspección	
Número tarea: HU 5-3	Historia de usuario: HU 5 Gestionar inspección
Nombre de la tarea: Eliminar inspección	
Tipo de tarea: Desarrollo	
Fecha inicio: 6/2/2014	Fecha fin: 7/2/2014
Programador responsable: Edry Garcia Ramos – Gisela Sagarra Mendoza	
Descripción: Recibe como parámetros el identificador de la inspección que se quiere eliminar en la aplicación	

Tabla 46 Tarea de programación 1 Mostar tipo de inspección para la Historia de usuario Mostar tipo de inspección

Tarea: Mostar tipo de inspección	
Número tarea: HU 6-1	Historia de usuario: HU 6 Mostar tipo de inspección
Nombre de la tarea: Mostar tipo de inspección	
Tipo de tarea: Desarrollo	



Fecha inicio: 10/2/2014	Fecha fin: 12/2/2014
Programador responsable: Edry Garcia Ramos – Gisela Sagarra Mendoza	
Descripción: Recibe como parámetros el identificador del tipo de inspección que se quiere mostrar en la aplicación	

Tabla 47 Tarea de programación 1 Mostrar estado de la inspección para la Historia de usuario
Mostrar estado de la inspección

Tarea: Mostrar estado de la inspección	
Número tarea: HU 7-1	Historia de usuario: HU 7 Mostrar estado de la inspección
Nombre de la tarea: Mostrar estado de la inspección	
Tipo de tarea: Desarrollo	
Fecha inicio: 12/2/2014	Fecha fin: 14/2/2014
Programador responsable: Edry Garcia Ramos – Gisela Sagarra Mendoza	
Descripción: Recibe como parámetros el identificador de la inspección que se quiere mostrar su estado en la aplicación	

Tabla 48 Tarea de programación 1 Crear desechos para la Historia de usuario Gestionar desechos

Tarea: Gestionar desechos	
Número tarea: HU 11-1	Historia de usuario: HU 11 Gestionar desechos
Nombre de la tarea: Crear desechos	
Tipo de tarea: Desarrollo	
Fecha inicio: 17/2/2014	Fecha fin: 17/2/2014
Programador responsable: Edry Garcia Ramos – Gisela Sagarra Mendoza	
Descripción: Recibe como parámetros todos los datos del desecho que se quiere insertar en la aplicación	



Tabla 49 Tarea de programación 2 Modificar desechos para la Historia de usuario Gestionar desechos

Tarea: Gestionar desechos	
Número tarea: HU 11-2	Historia de usuario: HU 11 Gestionar desechos
Nombre de la tarea: Modificar desechos	
Tipo de tarea: Desarrollo	
Fecha inicio: 18/2/2014	Fecha fin: 18/2/2014
Programador responsable: Edry Garcia Ramos – Gisela Sagarra Mendoza	
Descripción: Recibe como parámetros todos los datos del desecho que se quiere modificar en la aplicación	

Tabla 50 Tarea de programación 3 Eliminar desechos para la Historia de usuario Gestionar desechos

Tarea: Gestionar desechos	
Número tarea: HU 11-3	Historia de usuario: HU 11 Gestionar desechos
Nombre de la tarea: Eliminar desechos	
Tipo de tarea: Desarrollo	
Fecha inicio: 19/2/2014	Fecha fin: 20/2/2014
Programador responsable: Edry Garcia Ramos – Gisela Sagarra Mendoza	
Descripción: Recibe como parámetros el identificador del desecho que se quiere eliminar en la aplicación	

Tabla 51 Tarea de programación 1 Mostar categoría de desechos para la Historia de usuario Mostar categoría de desechos

Tarea: Mostar categoría de desechos	
Número tarea: HU 12-1	Historia de usuario: HU 12 Mostar categoría de desechos
Nombre de la tarea: Mostar categoría de desechos	
Tipo de tarea: Desarrollo	



Fecha inicio: 20/2/2014	Fecha fin: 21/2/2014
Programador responsable: Edry Garcia Ramos – Gisela Sagarra Mendoza	
Descripción: Recibe como parámetros el identificador del desecho que se quiere mostrar su categoría en la aplicación	

Tabla 52 Tarea de programación 1 Crear medida para la Historia de usuario Gestionar medida

Tarea: Gestionar medida	
Número tarea: HU 8-1	Historia de usuario: HU 8 Gestionar medida
Nombre de la tarea: Crear medida	
Tipo de tarea: Desarrollo	
Fecha inicio: 10/3/2014	Fecha fin: 12/3/2014
Programador responsable: Edry Garcia Ramos – Gisela Sagarra Mendoza	
Descripción: Recibe como parámetros todos los datos de la medida que se quiere insertar en la aplicación	

Tabla 53 Tarea de programación 2 Modificar medida para la Historia de usuario Gestionar medida

Tarea: Gestionar medida	
Número tarea: HU 8-2	Historia de usuario: HU 8 Gestionar medida
Nombre de la tarea: Modificar medida	
Tipo de tarea: Desarrollo	
Fecha inicio: 12/3/2014	Fecha fin: 13/3/2014
Programador responsable: Edry Garcia Ramos – Gisela Sagarra Mendoza	
Descripción: Recibe como parámetros los datos de la medida que se quiere modificar en la aplicación	



Tabla 54 Tarea de programación 3 Eliminar medida para la Historia de usuario Gestionar medida

Tarea: Gestionar medida	
Número tarea: HU 8-3	Historia de usuario: HU 8 Gestionar medida
Nombre de la tarea: Eliminar medida	
Tipo de tarea: Desarrollo	
Fecha inicio: 13/3/2014	Fecha fin: 14/3/2014
Programador responsable: Edry Garcia Ramos – Gisela Sagarra Mendoza	
Descripción: Recibe como parámetros el identificador de la medida que se quiere eliminar en la aplicación	

Tabla 55 Tarea de programación 1 Mostar resultado de la medida para la Historia de usuario Mostar resultado de la medida

Tarea: Mostar resultado de la medida	
Número tarea: HU 9-1	Historia de usuario: HU 9 Mostar resultado de la medida
Nombre de la tarea: Mostar resultado de la medida	
Tipo de tarea: Desarrollo	
Fecha inicio: 17/3/2014	Fecha fin: 19/3/2014
Programador responsable: Edry Garcia Ramos – Gisela Sagarra Mendoza	
Descripción: Recibe como parámetros el identificador de la medida que se quiere mostrar su resultado en la aplicación	

Tabla 56 Tarea de programación 1 Mostar normativa de la medida para la Historia de usuario Mostar normativa de la medida

Tarea: Mostar normativa de la medida	
Número tarea: HU 10-1	Historia de usuario: HU 10 Mostar normativa de la medida
Nombre de la tarea: Mostar resultado de la medida	
Tipo de tarea: Desarrollo	



Fecha inicio: 19/3/2014	Fecha fin: 21/3/2014
Programador responsable: Edry Garcia Ramos – Gisela Sagarra Mendoza	
Descripción: Recibe como parámetros el identificador de la medida que se quiere mostrar su normativa en la aplicación	

Tabla 57 Tarea de programación 1 Crear suceso extraordinario para la Historia de usuario Gestionar suceso extraordinario

Tarea: Gestionar suceso extraordinario	
Número tarea: HU 14-1	Historia de usuario: HU 14 Gestionar suceso extraordinario
Nombre de la tarea: Crear suceso extraordinario	
Tipo de tarea: Desarrollo	
Fecha inicio: 7/4/2014	Fecha fin: 9/4/2014
Programador responsable: Edry Garcia Ramos – Gisela Sagarra Mendoza	
Descripción: Recibe como parámetros todos los datos del suceso extraordinario que se quiere modificar en la aplicación	

Tabla 58 Tarea de programación 2 Modificar suceso extraordinario para la Historia de usuario Gestionar suceso extraordinario

Tarea: Gestionar suceso extraordinario	
Número tarea: HU 14-2	Historia de usuario: HU 14 Gestionar suceso extraordinario
Nombre de la tarea: Modificar suceso extraordinario	
Tipo de tarea: Desarrollo	
Fecha inicio: 10/4/2014	Fecha fin: 11/4/2014
Programador responsable: Edry Garcia Ramos – Gisela Sagarra Mendoza	
Descripción: Recibe como parámetros los datos del suceso extraordinario que se quieren modificar en la aplicación	



Tabla 59 Tarea de programación 3 Eliminar suceso extraordinario para la Historia de usuario Gestionar suceso extraordinario

Tarea: Gestionar suceso extraordinario	
Número tarea: HU 14-3	Historia de usuario: HU 14 Gestionar suceso extraordinario
Nombre de la tarea: Eliminar suceso extraordinario	
Tipo de tarea: Desarrollo	
Fecha inicio: 10/4/2014	Fecha fin: 11/4/2014
Programador responsable: Edry Garcia Ramos – Gisela Sagarra Mendoza	
Descripción: Recibe como parámetros el identificador del suceso extraordinario que se quiere eliminar en la aplicación	

Tabla 60 Tarea de programación 1 Mostar tipo de suceso extraordinario para la Historia de usuario Mostar tipo de suceso extraordinario

Tarea: Mostar tipo de suceso extraordinario	
Número tarea: HU 15-1	Historia de usuario: HU 15 Mostar tipo de inspección
Nombre de la tarea: Mostar tipo de suceso extraordinario	
Tipo de tarea: Desarrollo	
Fecha inicio: 21/4/2014	Fecha fin: 23/4/2014
Programador responsable: Edry Garcia Ramos – Gisela Sagarra Mendoza	
Descripción: Recibe como parámetros el identificador del tipo de suceso extraordinario que se quiere mostrar en la aplicación	

Tabla 61 Tarea de programación 1 Generar gráficos para la Historia de usuario Generar gráficos

Tarea: Generar gráficos	
Número tarea: HU 17-1	Historia de usuario: HU 17 Generar gráficos
Nombre de la tarea: Generar gráficos	
Tipo de tarea: Desarrollo	



Fecha inicio: 30/4/2014	Fecha fin: 2/5/2014
Programador responsable: Edry Garcia Ramos – Gisela Sagarra Mendoza	
Descripción: Recibe como parámetros el tipo de proceso, el trimestre, y el año, la aplicación debe mostrar un archivo de extensión .pdf con el reporte.	

Anexo 3 Tarjetas CRC

Tabla 62 Tarjeta CRC SIGA.portal.Escritorio.

SIGA.portal.Escritorio	
Responsabilidades	Colaboradores
init	Ext.tree.TreePanel Ext.Panel

Tabla 63 Tarjeta CRC SIGA.componets.WinPersonas

SIGA.componets.WinPersonas	
Responsabilidades	Colaboradores
initComponent	Ext.Window Ext.grid.GridPane

Tabla 64 Tarjeta CRC SIGA.Lib.Util

SIGA.Lib.Util	
Responsabilidades	Colaboradores
mostarMensaje	Ext.Msg
mask	Ext.mask Ext.unmask
s_url	



Tabla 65 Tarjeta CRC SIGA.Lib.Base64

SIGA.Lib.Base64	
Responsabilidades	Colaboradores
encode	
decode	
_utf8_encode	
_utf8_decode	

Tabla 66 Tarjeta CRC Portal.acceso.Login

Portal.acceso.Login	
Responsabilidades	Colaboradores
init	

Tabla 67 Tarjeta CRC Portal.acceso.Login.View

Portal.acceso.Login.View	
Responsabilidades	Colaboradores
loginForm	Ext.form.FormPanel Ext.Window

Tabla 68 Tarjeta CRC SIGA.portal.Usuario

SIGA.portal.Usuario	
Responsabilidades	Colaboradores
init	Ext.data.Model Ext.Ajax
editUser	SIGA.Lib.Util
delUser	SIGA.Lib.Util
reportUser	SIGA.Lib.Util



Tabla 69 Tarjeta CRC SIGA.portal.Usuario.View

SIGA.portal.Usuario.View	
Responsabilidades	Colaboradores
gridUser	Ext.grid.Panel
windowsEditUser	Ext.Window Ext.form.FormPanel
windowsSaveUser	Ext.Window Ext.form.FormPanel

Tabla 70 Tarjeta CRC SIGA.portal. Sucesos

SIGA.portal. Sucesos	
Responsabilidades	Colaboradores
init	Ext.data.Model
editSuc	SIGA.Lib.Util
delSuc	SIGA.Lib.Util
reportSuc	SIGA.Lib.Util

Tabla 71 Tarjeta CRC SIGA.portal.Sucesos.View

SIGA.portal.Sucesos.View	
Responsabilidades	Colaboradores
windowsReportSuc	Ext.Window Ext.TabPanel
chartSuc	Ext.chart.Chart Ext.Panel
gridSuc	Ext.grid.Panel
windowsEditSuc	Ext.Window Ext.form.FormPanel
windowsSaveSuc	Ext.Window



Ext.form.FormPanel

Tabla 72 Tarjeta CRC SIGA.portal.Conciliaciones

SIGA.portal.Conciliaciones	
Responsabilidades	Colaboradores
init	Ext.data.Model
editConc	SIGA.Lib.Util
delConc	SIGA.Lib.Util
reportConc	SIGA.Lib.Util

Tabla 73 Tarjeta CRC SIGA.portal. Conciliaciones.View

SIGA.portal. Conciliaciones.View	
Responsabilidades	Colaboradores
windowsReportConc	Ext.Window Ext.TabPanel
chartConc	Ext.chart.Chart Ext.Panel
gridConc	Ext.grid.Panel
windowsEditConc	Ext.Window Ext.form.FormPanel
windowsSaveConc	Ext.Window Ext.form.FormPanel

Tabla 74 Tareas CRC SIGA.portal.Desechos

SIGA.portal.Desechos	
Responsabilidades	Colaboradores
Init	Ext.data.Model
editDec	SIGA.Lib.Util



delDec	SIGA.Lib.Util
reportDec	SIGA.Lib.Util

Tabla 75 Tarjeta CRC SIGA.portal.Desechos.View

SIGA.portal.Desechos.View	
Responsabilidades	Colaboradores
windowsReportDec	Ext.Window Ext.TabPanel
chartDec	Ext.chart.Chart Ext.Panel
gridDec	Ext.grid.Panel
windowsEditDec	Ext.Window Ext.form.FormPanel
windowsSaveDec	Ext.Window Ext.form.FormPanel

Tabla 76 Tarjeta CRC SIGA.portal.Medidas

SIGA.portal.Medidas	
Responsabilidades	Colaboradores
init	Ext.data.Model
editMed	SIGA.Lib.Util
delMed	SIGA.Lib.Util
reportMed	SIGA.Lib.Util

Tabla 77 Tarjeta CRC SIGA.portal.Medidas.View

SIGA.portal.Medidas.View	
Responsabilidades	Colaboradores
windowsReportMed	Ext.Window



	Ext.TabPanel
chartMed	Ext.chart.Chart Ext.Panel
gridMed	Ext.grid.Panel
windowsEditMed	Ext.Window Ext.form.FormPanel
windowsSaveMed	Ext.Window Ext.form.FormPanel

Tabla 78 Tarjeta CRC SIGA.portal.Autorizaciones

SIGA.portal.Autorizaciones	
Responsabilidades	Colaboradores
init	Ext.data.Model
editAuto	SIGA.Lib.Util
delAuto	SIGA.Lib.Util
reportAuto	SIGA.Lib.Util

Tabla 79 Tarjeta CRC SIGA.portal.Autorizaciones.View

SIGA.portal.Autorizaciones.View	
Responsabilidades	Colaboradores
windowsReport	Ext.Window Ext.TabPanel
chartAuto	Ext.chart.Chart Ext.Panel
gridAuto	Ext.grid.Panel
windowsEdit	Ext.Window Ext.form.FormPanel
windowsSave	Ext.Window Ext.form.FormPanel



Tabla 80 Tarjeta CRC SIGA.portal.Escritorio

SIGA.portal.Escritorio	
Responsabilidades	Colaboradores
init	Ext.tree.TreePanel Ext.Panel

Anexo 4 Caso de prueba

Tabla 81 Caso de Prueba de Aceptación Adicionar inspección.

Caso de Prueba de Aceptación Adicionar inspección	
Número HU: 5	Número:1
Nombre: Gestionar inspección.	
Descripción: Prueba de la funcionalidad Adicionar inspección.	
Condiciones de ejecución: Debe existir un usuario autenticado.	
Pasos de ejecución: En la pestaña Procesos del menú principal se selecciona la opción Inspecciones, luego en la nueva ventana que el sistema genera, el usuario acciona el botón Adicionar. En la nueva ventana se llenan todos los datos y se presiona el botón aceptar.	
Resultado esperado: Se adiciona la inspección	
Evaluación de la prueba: Prueba satisfactoria.	

Tabla 82 Caso de Prueba de Aceptación Modificar inspección.

Caso de Prueba de Aceptación Modificar inspección	
Número HU: 5	Número:2
Nombre: Gestionar inspección.	
Descripción: Prueba de la funcionalidad Modificar inspección	
Condiciones de ejecución: Debe existir un usuario autenticado.	
Pasos de ejecución: En la pestaña Procesos del menú principal se selecciona la opción Inspecciones, luego en la nueva ventana que el sistema genera, el usuario selecciona la inspección que desea modificar y acciona el botón modificar. En la nueva ventana se modifica los datos que se requieran y se presiona el botón aceptar.	



Resultado esperado: Se modifica la inspección
Evaluación de la prueba: Prueba satisfactoria.

Tabla 83 Caso de Prueba de Aceptación Eliminar inspección.

Caso de Prueba de Aceptación Eliminar inspección	
Número HU: 5	Número:3
Nombre: Gestionar inspección.	
Descripción: Prueba de la funcionalidad Eliminar inspección.	
Condiciones de ejecución: Debe existir un usuario autenticado.	
Pasos de ejecución: En la pestaña Procesos del menú principal se selecciona la opción Inspecciones, luego en la nueva ventana que el sistema genera, el usuario selecciona la inspección que desea eliminar y acciona el botón eliminar. En la nueva ventana se selecciona el botón aceptar.	
Resultado esperado: Se elimina la inspección.	
Evaluación de la prueba: Prueba satisfactoria.	

Tabla 84 Caso de Prueba de Aceptación Adicionar medida.

Caso de Prueba de Aceptación Adicionar medida	
Número HU: 8	Número:1
Nombre: Gestionar medidas.	
Descripción: Prueba de la funcionalidad Adicionar medidas.	
Condiciones de ejecución: Debe existir un usuario autenticado.	
Pasos de ejecución: En la pestaña Procesos del menú principal se selecciona la opción Medidas, luego en la nueva ventana que el sistema genera, el usuario acciona el botón Adicionar. En la nueva ventana se llenan todos los datos y se presiona el botón aceptar.	
Resultado esperado: Se adiciona la medida	
Evaluación de la prueba: Prueba satisfactoria.	



Tabla 85 Caso de Prueba de Aceptación Modificar medida.

Caso de Prueba de Aceptación Modificar medida	
Número HU: 8	Número:2
Nombre: Gestionar medidas.	
Descripción: Prueba de la funcionalidad Modificar medida	
Condiciones de ejecución: Debe existir un usuario autenticado.	
Pasos de ejecución: En la pestaña Procesos del menú principal se selecciona la opción Medidas, luego en la nueva ventana que el sistema genera, el usuario selecciona la medida que desea modificar y acciona el botón modificar. En la nueva ventana se modifica los datos que se requieran y se presiona el botón aceptar.	
Resultado esperado: Se modifica la medida	
Evaluación de la prueba: Prueba satisfactoria.	

Tabla 86 Caso de Prueba de Aceptación Eliminar medida

Caso de Prueba de Aceptación Eliminar medida	
Número HU: 8	Número:3
Nombre: Gestionar medidas.	
Descripción: Prueba de la funcionalidad Eliminar medidas.	
Condiciones de ejecución: Debe existir un usuario autenticado.	
Pasos de ejecución: En la pestaña Procesos del menú principal se selecciona la opción Medidas, luego en la nueva ventana que el sistema genera, el usuario selecciona la medida que desea eliminar y acciona el botón eliminar. En la nueva ventana se selecciona el botón aceptar.	
Resultado esperado: Se elimina la medida.	
Evaluación de la prueba: Prueba satisfactoria.	

Tabla 87 Caso de Prueba de Aceptación Adicionar desecho.

Caso de Prueba de Aceptación Adicionar desecho	
Número HU: 11	Número:1
Nombre: Gestionar desecho.	
Descripción: Prueba de la funcionalidad Adicionar desecho.	



Condiciones de ejecución: Debe existir un usuario autenticado.
Pasos de ejecución: En la pestaña Procesos del menú principal se selecciona la opción Desechos Peligrosos, luego en la nueva ventana que el sistema genera, el usuario acciona el botón Adicionar. En la nueva ventana se llenan todos los datos y se presiona el botón aceptar.
Resultado esperado: Se adiciona el desecho
Evaluación de la prueba: Prueba satisfactoria.

Tabla 88 Caso de Prueba de Aceptación Modificar desechos.

Caso de Prueba de Aceptación Modificar desechos	
Número HU: 11	Número:2
Nombre: Gestionar desecho.	
Descripción: Prueba de la funcionalidad Modificar desecho	
Condiciones de ejecución: Debe existir un usuario autenticado.	
Pasos de ejecución: En la pestaña Procesos del menú principal se selecciona la opción Desechos Peligrosos, luego en la nueva ventana que el sistema genera, el usuario selecciona el desecho que desea modificar y acciona el botón modificar. En la nueva ventana se modifican los datos que se requieran y se presiona el botón aceptar.	
Resultado esperado: Se modifica el desecho.	
Evaluación de la prueba: Prueba satisfactoria.	

Tabla 89 Caso de Prueba de Aceptación Eliminar desecho.

Caso de Prueba de Aceptación Eliminar desecho	
Número HU: 11	Número:3
Nombre: Gestionar desecho.	
Descripción: Prueba de la funcionalidad Eliminar desecho.	
Condiciones de ejecución: Debe existir un usuario autenticado.	
Pasos de ejecución: En la pestaña Procesos del menú principal se selecciona la opción Desechos Peligrosos, luego en la nueva ventana que el sistema genera, el	



usuario selecciona el desecho que desea eliminar y acciona el botón eliminar. En la nueva ventana se selecciona el botón aceptar.
Resultado esperado: Se elimina el desecho.
Evaluación de la prueba: Prueba satisfactoria.

Tabla 90 Caso de Prueba de Aceptación Adicionar suceso extraordinario.

Caso de Prueba de Aceptación Adicionar suceso extraordinario	
Número HU: 14	Número: 1
Nombre: Gestionar suceso extraordinario.	
Descripción: Prueba de la funcionalidad Adicionar suceso extraordinario.	
Condiciones de ejecución: Debe existir un usuario autenticado.	
Pasos de ejecución: En la pestaña Procesos del menú principal se selecciona la opción Sucesos Extraordinarios, luego en la nueva ventana que el sistema genera, el usuario acciona el botón Adicionar. En la nueva ventana se llenan todos los datos y se presiona el botón aceptar.	
Resultado esperado: Se adiciona el suceso extraordinario.	
Evaluación de la prueba: Prueba satisfactoria.	

Tabla 91 Caso de Prueba de Aceptación Modificar suceso extraordinario.

Caso de Prueba de Aceptación Modificar suceso extraordinario	
Número HU: 14	Número:2
Nombre: Gestionar suceso extraordinario.	
Descripción: Prueba de la funcionalidad Modificar suceso extraordinario.	
Condiciones de ejecución: Debe existir un usuario autenticado.	
Pasos de ejecución: En la pestaña Procesos del menú principal se selecciona la opción Sucesos Extraordinarios, luego en la nueva ventana que el sistema genera, el usuario selecciona el suceso extraordinario que desea modificar y acciona el botón modificar. En la nueva ventana se modifican los datos que se requieran y se presiona el botón aceptar.	
Resultado esperado: Se modifica el suceso extraordinario.	
Evaluación de la prueba: Prueba satisfactoria.	



Tabla 92 Caso de Prueba de Aceptación Eliminar suceso extraordinario.

Caso de Prueba de Aceptación Eliminar suceso extraordinario	
Número HU: 14	Número:3
Nombre: Gestionar suceso extraordinario.	
Descripción: Prueba de la funcionalidad Eliminar suceso extraordinario.	
Condiciones de ejecución: Debe existir un usuario autenticado.	
Pasos de ejecución: En la pestaña Procesos del menú principal se selecciona la opción Sucesos Extraordinarios, luego en la nueva ventana que el sistema genera, el usuario selecciona el suceso extraordinario que desea eliminar y acciona el botón eliminar. En la nueva ventana se selecciona el botón aceptar.	
Resultado esperado: Se elimina el suceso extraordinario.	
Evaluación de la prueba: Prueba satisfactoria.	

Tabla 93 Caso de Prueba de Aceptación Adicionar conciliaciones.

Caso de Prueba de Aceptación Adicionar conciliaciones	
Número HU: 16	Número: 1
Nombre: Gestionar conciliación.	
Descripción: Prueba de la funcionalidad Adicionar conciliación.	
Condiciones de ejecución: Debe existir un usuario autenticado.	
Pasos de ejecución: En la pestaña Procesos del menú principal se selecciona la opción Conciliación, luego en la nueva ventana que el sistema genera, el usuario acciona el botón Adicionar. En la nueva ventana se llenan todos los datos y se presiona el botón aceptar.	
Resultado esperado: Se adiciona la conciliación.	
Evaluación de la prueba: Prueba satisfactoria.	

Tabla 94 Caso de Prueba de Aceptación Modificar conciliación.

Caso de Prueba de Aceptación Modificar conciliación	
Número HU: 16	Número:2
Nombre: Gestionar conciliación.	
Descripción: Prueba de la funcionalidad Modificar conciliación.	



Condiciones de ejecución: Debe existir un usuario autenticado.
Pasos de ejecución: En la pestaña Procesos del menú principal se selecciona la opción Conciliación, luego en la nueva ventana que el sistema genera, el usuario selecciona la conciliación que desea modificar y acciona el botón modificar. En la nueva ventana se modifican los datos que se requieran y se presiona el botón aceptar.
Resultado esperado: Se modifica la conciliación.
Evaluación de la prueba: Prueba satisfactoria.

Tabla 95 Caso de Prueba de Aceptación Eliminar conciliación.

Caso de Prueba de Aceptación Eliminar conciliación.	
Número HU: 16	Número:3
Nombre: Gestionar conciliación.	
Descripción: Prueba de la funcionalidad Eliminar conciliación.	
Condiciones de ejecución: Debe existir un usuario autenticado.	
Pasos de ejecución: En la pestaña Procesos del menú principal se selecciona la opción Conciliación, luego en la nueva ventana que el sistema genera, el usuario selecciona el suceso extraordinario que desea eliminar y acciona el botón eliminar. En la nueva ventana se selecciona el botón aceptar.	
Resultado esperado: Se elimina la conciliación.	
Evaluación de la prueba: Prueba satisfactoria.	

Tabla 96 Caso de Prueba de Aceptación Adicionar usuario.

Caso de Prueba de Aceptación Adicionar usuario	
Número HU: 1	Número: 1
Nombre: Gestionar usuario.	
Descripción: Prueba de la funcionalidad Adicionar usuario.	
Condiciones de ejecución: Se debe estar autenticado como usuario administrador.	
Pasos de ejecución: En la pestaña Roles del menú principal se selecciona la el tipo de usuario que se desea insertar, luego en la ventana que el sistema genera, se presiona el botón Adicionar. En la nueva ventana se llenan todos los datos y se selecciona el	



botón aceptar.
Resultado esperado: Se adiciona el usuario a la aplicación.
Evaluación de la prueba: Prueba satisfactoria.

Tabla 97 Caso de Prueba de Aceptación Modificar usuario.

Caso de Prueba de Aceptación Modificar usuario	
Número HU: 1	Número: 2
Nombre: Gestionar usuario.	
Descripción: Prueba de la funcionalidad Modificar usuario.	
Condiciones de ejecución: Se debe estar autenticado como usuario administrador.	
Pasos de ejecución: En la pestaña Roles del menú principal se selecciona el rol, luego el usuario que se desea modificar, en la ventana que el sistema genera, se presiona el botón Modificar. En la nueva ventana se modifican los campos necesarios y se selecciona el botón modificar.	
Resultado esperado: Se modifica el usuario en la aplicación.	
Evaluación de la prueba: Prueba satisfactoria.	

Tabla 98 Caso de Prueba de Aceptación Eliminar usuario.

Caso de Prueba de Aceptación Eliminar usuario	
Número HU: 1	Número: 3
Nombre: Gestionar usuario.	
Descripción: Prueba de la funcionalidad Eliminar usuario.	
Condiciones de ejecución: Se debe estar autenticado como usuario administrador.	
Pasos de ejecución: En la pestaña Roles del menú principal se selecciona el rol y luego el usuario que se desea eliminar, se presiona el botón Eliminar.	
Resultado esperado: Se elimina el usuario de la aplicación.	
Evaluación de la prueba: Prueba satisfactoria.	

Tabla 99 Caso de Prueba de Aceptación Autenticar usuario.

Caso de Prueba de Aceptación Autenticar usuario	
Número HU: 2	Número: 1



Nombre: Autenticar usuario.
Descripción: Prueba de la funcionalidad Autenticar usuario.
Condiciones de ejecución: Se debe acceder a la página de autenticación
Pasos de ejecución: Se debe insertar el usuario y la contraseña correctamente.
Resultado esperado: Se muestra la página de inicio de la aplicación.
Evaluación de la prueba: Prueba satisfactoria.

Tabla 100 Caso de Prueba de Aceptación Mostar tipo de autorización.

Caso de Prueba de Aceptación Mostar tipo de autorización	
Número HU: 4	Número: 1
Nombre: Mostar tipo de autorización.	
Descripción: Prueba de la funcionalidad Mostrar tipo de autorización	
Condiciones de ejecución: Se debe acceder a la página de autenticación	
Pasos de ejecución: Se debe insertar el usuario y la contraseña correctamente.	
Resultado esperado: Se muestra la página de inicio de la aplicación.	
Evaluación de la prueba: Prueba satisfactoria.	

Anexo 5 Entrevista

Entrevista	
Entrevistados:	Mariela Marrero Garcia Yanaysa Concepción Castellón
Lugar de la entrevista:	Oficina de Regulación Ambiental y Seguridad Nuclear
Preguntas	Respuestas
Objetivos	Definir los principales problemas presentados en la actualidad en la oficina con respecto a los procesos, detectar los procesos a informatizar y definir los principales roles que se desempeñan en el flujo que recorre la información
¿Qué procesos son gestionadas por la	En la oficina se gestionan distintos tipos de



oficina?	procesos dentro de estos se pueden listar algunos como: procesos para recursos humanos, la regulación ambiental, la defensa, protección y seguridad física, entre otros.
¿De estos cuáles son los procesos que se desean informatizar?	Se desean informatizar los seis procesos relacionados con la regulación ambiental que se realizan a nivel nacional, los cuales son: autorizaciones de la actividad reguladora ambiental, inspección estatal de la actividad reguladora, medidas aplicadas a las empresas, sucesos ambientales extraordinarios, conciliaciones y desechos peligrosos manipulados o creados por las empresas.
¿Por qué es una necesidad para ustedes la informatización de esos procesos?	Se necesita la informatización de estos procesos porque la oficina cuenta con sucursales en todas las provincias del país y la información referente a estos procesos es enviada por correo en tablas Excel las cuales dificultan por la cantidad existente la revisión, realización de estadísticas, búsqueda y consulta.
¿Cómo se realizan estos procesos en la oficina?	Primeramente la información es enviada desde las provincias por correo cada trimestre, esta información es almacenada hasta que esté completa para luego ser revisada por distintos expertos para determinar si es correcta y en caso contrario corregirla, posteriormente se pasa a la etapa de procesado de la información donde se generan reportes provinciales y a nivel de



	país, además se crean gráficos con los resultados obtenidos para tener una mejor apreciación de los datos.
¿Cuántas personas participan en el tratamiento de la información?	Participan las personas que trabajan en las provincias enviando información, luego los expertos validan la calidad de esta y finalmente los directores son los encargados de crear los reportes.