

Universidad de las Ciencias Informáticas

Trabajo de Diploma para optar por el Título de Ingeniero en Ciencias Informáticas

Título:

Módulo de reportes para el subsistema Patrimonio de las Fuerzas Armadas Revolucionarias

Autores: Cdt. Maylenis Peña Gross

Cdt. Raúl Alex Meno Macías

Tutores: Ing. Jenny Martínez Martínez

Ing. Jorge Luis Saavedra Rubio

Ing. Asist. Yaili Ladea Velázquez

Ing. Maykel Ramírez Reyes

La Habana, 2018

Año 60 de la Revolución



“La única lucha que se pierde es la que se abandona”

Ernesto Guevara de la Serna

Declaración de Autoría

Declaramos por este medio que Maylenis Peña Gross, con carnet de identidad 95052918492, y Raúl Alex Meno Macías, con carnet de identidad 95042125542 ser los autores principales del trabajo titulado **“Módulo de reportes para el subsistema Patrimonio de las Fuerzas Armadas Revolucionarias”** y autorizo a la Empresa de Tecnologías de la Información para la Defensa a hacer uso de la misma en su beneficio, así como los derechos patrimoniales de la misma con carácter exclusivo.

Declaramos que todo lo anteriormente expuesto se ajusta a la verdad, y asumimos la responsabilidad moral y jurídica que se derive de este juramento profesional.

Y para que así conste, firmamos la presente declaración de autoría en La Habana a los ____ días del mes de _____ del año 2018.

Maylenis Peña Gross

Firma de Autor

Raúl Alex Meno Macías

Firma de Autor

Ing. Jenny Martínez Martínez

Firma de Tutor

Ing. Jorge Luis Saavedra Rubio

Firma de Tutor

Ing. Asist. Yaili Ladea Velázquez

Firma del Co-tutor

Ing. Maykel Ramírez Reyes

Firma del Co-tutor

Datos del contacto

Síntesis de los Tutores

Jenny Martínez Martínez

Graduada de Ingeniero en Ciencias Informáticas en la UCI. A partir de ese momento se incorpora como Analista de sistemas, en el centro Logística de la División de Gestión de Entidades (DGE), perteneciente a la Empresa de Tecnologías de la Información para la Defensa (XETID), donde se desempeña actualmente.

Email: jmartinezm@xetid.cu

Jorge Luis Saavedra Rubio

Graduado de Ingeniero en Ciencias Informáticas en la UCI en 2012. Después de graduado se incorpora como arquitecto en sistemas de bases de datos en el centro Logística, perteneciente a la División de Gestión de Entidades, de la Empresa de Tecnologías de la Información para la Defensa (XETID).

Email: jlsaavedra@xetid.cu

Síntesis de los Co-tutores:

Maykel Ramírez Reyes

Graduado de Ingeniero en Ciencias Informáticas en la UCI en el 2016. A partir de ese momento se encuentra preparándose como profesor de las asignaturas Gestión de Software y Componente Profesional de Ingeniería y Gestión de Software perteneciente al Departamento de Ingeniería de la Facultad 1.

Email: maykelrr@uci.cu

Yaili Ladea Velázquez

Graduada de Ingeniero en Ciencias Informáticas en la UCI en el 2008. Actualmente pertenece al Departamento de Ingeniería y Gestión de Software donde imparte las asignaturas Sistemas de Bases de Datos I y II. Desde el año 2015 posee la categoría de Asistente.

Email: yledea@uci.cu

Dedicatoria de Maylenis:

*A mis padres y mi hermano, por ser mi guía, mi luz, mi ejemplo,
por todo el amor que me han dado, por lo que soy...*

A mi novio, por su dedicación y cariño en los momentos de estrés...

A la Revolución, por brindarme esta hermosa posibilidad...

Dedicatoria de Raúl:

*A mis padres que a ellos les debo todo lo que soy, sin ellos no
hubiera podido llegar hasta aquí.*

*A mi hermano Ernestico que siempre ha sido mi ejemplo a seguir
y siempre aparece cuando más lo necesito.*

*A mi querida novia por ser mi amiga más fiel y estar a mi lado
en todo momento.*

*A todos mis amigos del 123, que han sido una gran familia
para mí y han hecho de estos 5 años los mejores de mi vida.*

Agradecimientos de Maylenis:

A mi familia, a mi mamá, mi papá y mi hermano que me han guiado en la vida y me han acompañado en el largo camino de mi educación, por sus deseos de verme como una profesional y contar con la seguridad, apoyo y confianza, por siempre estar orgullosos de mí. A ustedes que la vida sería demasiada corta para agradecerles y pagarles todo lo que han hecho por mí.

A mi novio y amigo que me ha acompañado en los últimos años y ha aguantado todas mis malcriadeces, pero sobre todo me ha ayudado en las buenas y en las malas.

A mis tutores Maykel, Jenny y Saavedra por toda la ayuda que me han brindado en este trayecto tan difícil.

A mi compañero de tesis por compartir juntos este momento.

Al profesor Denier que cada vez que lo necesitaba estaba dispuesto a ayudarme sin ser mi tutor.

A todas mis amistades más cercanas y a mis compañeros de aula por los buenos momentos que hemos compartido todo este tiempo, en especial Javier, Serguey, Dairelis, Ana Isabel, Cristy, Chirino, Yanisbel y Leiny.

A todos aquellos que de una forma u otra me han brindado su apoyo para ser hoy una profesional.

Agradecimientos de Raúl:

A toda mi familia por su apoyo y en especial a mis padres, por haberme guiado por este largo y difícil camino, por servirme de ejemplo en todo momento.

A mis tutores y compañeros de la XETiD por todo su tiempo dedicado a ayudarnos a cumplir esta meta.

A Maylenis por ser una excelente compañera de Tesis.

A mi otra familia que son mis amigos del 123, que siempre he sentido el apoyo de todos en los momentos más difíciles.

A todos los que de una forma u otra me han brindado su ayuda para poder hoy lograr este sueño.

Gracias

Resumen

La empresa de Tecnologías de la Información para la Defensa (XETID) posee un amplio catálogo de productos que garantizan el análisis, la gestión y la seguridad de los datos, como el caso del subsistema Patrimonio desarrollado por el centro de Logística. En el presente trabajo se desarrolló un módulo de reportes para Patrimonio con el objetivo de procesar la información almacenada por el sistema, de forma que facilite su análisis y sirva como herramienta de apoyo a la dirección de la organización. Se realizó una investigación de sistemas relacionados con la gestión de reportes destacándose los elementos a utilizar en el desarrollo del módulo. Para la implementación de la propuesta de solución se empleó la guía de desarrollo *Prodesoft* y la siguiente base tecnológica: el marco de trabajo *Zeolides 2.2*, *Ext.JS 3.4*, *Doctrine 1.2* y *Zend Framework* en su versión 1.7. Como sistema gestor de bases de datos *PostgreSQL* versión 9.3, para la generación de los reportes *JasperReport Server* junto al diseñador de reportes *iReport 5.6* y *Visual Paradigm 8.0* como herramienta de modelado. Para suministrarle validez a la propuesta de solución se definió una estrategia de prueba que permitió asegurar que el sistema es funcional y cumple con los requisitos identificados con el cliente. Se realizaron pruebas funcionales, de carga y estrés y de aceptación, por lo que se garantizó el correcto funcionamiento de la propuesta de solución. Al culminar este trabajo se proporcionó una herramienta ajustada a las exigencias de la empresa que permite el diseño y visualización de los reportes del subsistema Patrimonio de las FAR.

Palabras claves: generación de reportes, logística, patrimonio, reportes, sistemas de información.

Índice

Introducción 1

Capítulo 1: Fundamentación teórica. 6

Conceptos relacionados al dominio del problema 6

Sistema de información 6

Recuperación de información 6

Reportes 7

Generación de reportes 7

Estudio de los sistemas homólogos 8

El Gran Paitití 10

Módulo de reportes de Patrimonio. 10

Módulo de Reportes del SUIN 11

Módulo de reportes de la Suite de Gestión de Proyectos..... 11

Módulo de reportes de la Suite de Gestión de Proyectos..... 13

Selección del entorno de desarrollo para la construcción de la propuesta de solución.
..... 13

Conclusiones del capítulo 21

Capítulo 2: Características y diseño del módulo 22

Modelo de dominio 22

Propuesta de solución. 25

Especificación de requisitos funcionales. 25

Concepción de los modelos de procesos..... 27

Prototipos de interfaz de usuario..... 28

Especificación de requisitos no funcionales. 29

Arquitectura de la solución..... 31

Modelo del diseño..... 32

Diseño de la base de datos..... 35

Patrones de diseño	36
Conclusiones del capítulo	38
Capítulo 3: Implementación y Pruebas.	40
Diagrama de componentes	40
Diagrama de despliegue	42
Estándares de codificación y tratamiento de errores.....	43
Estándares de codificación	43
Tratamiento de errores	45
Interfaces del módulo.....	45
Pruebas	47
Pruebas Funcionales	47
Pruebas de Carga y Estrés.....	53
Validación de la idea a defender.....	56
Conclusiones del capítulo.	59
Beneficios del módulo de reportes para el subsistema Patrimonio de las FAR.	60
Conclusiones generales.....	61
Recomendaciones.	62
Bibliografía.....	63
Anexo 1: Entrevistas	69
Anexo 2: Especificación de requisitos funcionales.....	70
Anexo 3: Diagramas de estereotipos web.....	74
Anexo 4: Encuesta para evaluar la validación de la idea a defender.....	77

Índice de Tablas

Tabla 1: Resumen del análisis de los sistemas homólogos..... 12

Tabla 2: Catálogo de requisitos funcionales..... 26

Tabla 3: Estereotipos web utilizados..... 32

Tabla 4: Estereotipos usados en el diagrama de componentes 40

Tabla 5: Estereotipos utilizados en el diagrama de despliegue 42

Tabla 6: Variables empleadas en los escenarios de prueba de los requisitos
funcionales generar reporte 47

Tabla 7: Escenarios de los casos de pruebas RF 1: Generar reporte especialidad de
Campamento 49

Tabla 8: Escenarios de los casos de pruebas RF 2: Generar reporte especialidad de
Ingeniería..... 49

Tabla 9: Escenarios de los casos de pruebas RF 3: Generar reporte especialidad de
Sistema Empresarial..... 50

Tabla 10: Escenarios de los casos de pruebas RF 4: Generar reporte especialidad de
Vivienda..... 51

Tabla 11: Relación de no conformidades 53

Tabla 12: Relación de los especialistas del centro..... 56

Tabla 13: Resultado de los juicios emitidos por los encuestados..... 57

Índice de ilustraciones

Ilustración 1: Ciclo de vida de los reportes..... 9

Ilustración 2: Etapas que comprende Prodesoft..... 14

Ilustración 3: Modelo de dominio 24

Ilustración 4:Funcionamiento de la propuesta de solución 25

Ilustración 5: Especificación del RF 4: Generar reporte especialidad de Vivienda 27

Ilustración 6: PIU “Principal” 28

Ilustración 7: Arquitectura de la propuesta de solución 31

Ilustración 8: Diagrama de clases con estereotipos web del RF Generar reporte especialidad Vivienda 34

Ilustración 9: Diseño del modelo de datos..... 35

Ilustración 10: Diagrama de componentes 40

Ilustración 11: Diagrama de despliegue 42

Ilustración 12: Menú Inicio/ActivosFijos/Patrimonio/Reportes. 46

Ilustración 13: PIU funcional del módulo 46

Ilustración 14: Tiempo promedio para 25 especialistas..... 54

Ilustración 15: Tiempo promedio para 50 especialistas..... 55

Ilustración 16: Resultados de la validación 58

Introducción

Con el desarrollo de las ciencias y las tecnologías, los Sistemas de Información (SI) constituyen uno de los principales ámbitos de estudio en el área de organización de empresas, y el entorno donde las compañías desarrollan sus actividades se vuelve cada vez más complejo. La creciente globalización, el incremento de la competencia en los mercados de bienes y servicios, la rapidez en el desarrollo de las tecnologías de información, el aumento de la incertidumbre en el entorno y la reducción de los ciclos de vida de los productos originan que la información se convierta en un elemento clave para la gestión, así como para la supervivencia y crecimiento de la organización empresarial (Hernández, 2013).

El incremento exponencial de la información ha impulsado el desarrollo de herramientas de Inteligencia de Negocio o *Business Intelligence* (por sus siglas en inglés, BI) que permiten transformar datos de eventos pasados, su tendencia en información y esta, a su vez, en conocimiento. Con su uso, los datos son mostrados de manera gráfica, con reportes interactivos e indicadores que les brindan a los administradores conocimientos confiables con los que puedan anticiparse a eventos futuros, centrar los esfuerzos en los aspectos críticos y explotar los de mayor interés para la entidad (Lee, 2012).

Las organizaciones hoy en día adaptan los SI para la gestión de sus proyectos, que se deben al control de los costos y a los tiempos que permiten manejar (Yesith, 2013). Para la gestión de proyectos los SI se convierten en claves fundamentales de las tecnologías de información y juegan un papel importante (Lee, 2012).

La Empresa de Tecnologías de la Información para la Defensa (XETID), perteneciente a las Fuerzas Armadas Revolucionarias (FAR) se encarga del diseño, desarrollo y comercialización de soluciones, productos y servicios informáticos de excelencia. Tiene entre sus principales objetivos, desarrollar soluciones de código abierto, de acuerdo a su política de migración a software libre y brindar a los desarrolladores mayores conocimientos sobre las capacidades y características de los sistema que se desarrollan para garantizar la seguridad de la información que manipulan y facilitar, a su vez, la corrección de errores y modificación de funcionalidades (Garzón, y otros, 2016). Está compuesta por seis divisiones y éstas, a su vez, por centros productivos dentro de los cuales se encuentra el de Logística, donde se desarrollan soluciones para la gestión de la cadena de suministros.

Distra o Sistema de Gestión Empresarial fue desarrollado sobre el marco de trabajo *Zeolides* (MT) e integra todas las soluciones vinculadas con el Sistema de Planificación de Gestión Empresarial (ERP). El subsistema Patrimonio se encuentra vinculado a los activos fijos de la empresa y tiene como objetivo llevar el registro y control de los bienes inmuebles del patrimonio de las FAR. El sistema permite gestionar los nomencladores de objetos de obra, del documento del expediente y los nomencladores dinámicos. Además, gestiona el registro numérico y el expediente de los inmuebles y de los objetos de obra. Actualmente posee cuatro especialidades: Campamento, Vivienda, Ingeniería y Sistema Empresarial.

Para la entrega de la información del estado y avance de los inmuebles reales de las FAR se utilizan los reportes: Situación documental, Situación del registro, valor y certificado contable y Estado del registro contable que se obtienen en dependencia de la especialidad que hace uso de ellos, recopilando la información que acontece trimestralmente.

Actualmente los reportes se realizan de forma manual, con información confidencial que en muchos casos es incorrecta, antigua, discordante y eventualmente inconsistente. Esto a su vez, impacta en los resultados de la organización, al no tener la dirección de la misma información coherente y confiable que sirva de apoyo a la toma de decisiones en cada uno de los niveles. Los reportes, además, presentan problemas en la organización de los documentos, provocando pérdidas y duplicidad de la información.

Las consecuencias de esta situación se manifiestan en el aumento de los costos en desarrollo, demora en la entrega de la información por parte de las personas y poca fiabilidad en esta, debido a la posibilidad de introducción de errores humanos y pérdida de documentos afectando la toma de decisiones.

A partir de la situación problemática descrita anteriormente, se enuncia como **problema de investigación**: ¿Cómo proveer un mecanismo que procese la información almacenada por el subsistema Patrimonio de las Fuerzas Armadas Revolucionarias con el objetivo de facilitar su análisis?

Se define como **objeto de estudio** los módulos de reportes en sistemas de gestión de la información y el **campo de acción** lo constituyen el proceso de emisión de reportes en el subsistema Patrimonio de las Fuerzas Armadas Revolucionarias.

Para resolver el problema planteado se define como **objetivo general**: Desarrollar un Módulo de reportes para el subsistema Patrimonio que facilite el análisis de la

información almacenada en el subsistema Patrimonio de las Fuerzas Armadas Revolucionarias.

Por el objetivo previamente planteado se definen como **objetivos específicos** los siguientes:

1. Analizar los referentes teóricos que sustentan el desarrollo de la investigación, relacionados con la gestión de los reportes.
2. Caracterizar y diseñar el módulo de reportes para el subsistema Patrimonio.
3. Implementar el módulo de reportes para el subsistema Patrimonio.
4. Validar los resultados del módulo de reportes para el subsistema Patrimonio.

Para encaminar la investigación en vista a resolver el problema planteado se propone como **idea a defender**: El desarrollo de un módulo de reportes para el subsistema Patrimonio facilitará el análisis de la información solicitada por la dirección de las FAR.

Para dar solución a los objetivos específicos se definen las siguientes **tareas de investigación**:

1. Estudio de los principales conceptos que sustentan el desarrollo de la investigación, relacionados con los módulos de reportes.
2. Estudio del estado del arte para comprender el funcionamiento de las aplicaciones encargadas de realizar este proceso.
3. Caracterización del proceso de desarrollo, las herramientas y tecnología a utilizar.
4. Diseño de la propuesta de solución.
5. Descripción de la arquitectura y el modelado de la base de datos.
6. Definición de los patrones de diseños más adecuados para la construcción la propuesta de solución.
7. Presentación del estándar de codificación y explicación de las técnicas de programación a utilizar.
8. Desarrollo de la solución informática.
9. Definición de las pruebas a realizar a la solución y ejecución de las mismas.
10. Validación de los resultados obtenidos.

Para el desarrollo de las tareas científicas se emplearon diferentes métodos teóricos y empíricos en la búsqueda y el procesamiento de la información. Estos son:

Métodos teóricos:

Histórico - Lógico: se emplea con el propósito de constatar teóricamente cómo ha evolucionado en el tiempo la generación de reportes para el apoyo a la toma de decisiones en sistemas de gestión de información, así como las herramientas y tecnologías utilizadas en el desarrollo de aplicaciones de este tipo.

Analítico-Sintético: se emplea para el análisis de los elementos esenciales referentes a la información relacionada con los sistemas de gestión de Información para el apoyo a la toma de decisiones.

Inductivo-Deductivo: se emplea para arribar a razonamientos que puedan ser aplicables al problema a resolver, luego de adquirir una serie de elementos referentes a la generación de reportes para el apoyo a la toma de decisiones.

Modelación: se emplea a través de los diagramas del lenguaje de modelado UML; se utiliza en la representación de las características y las relaciones entre los objetos de la solución.

Métodos empíricos:

Observación: se emplea para comprender el funcionamiento real de los procesos, debido a que fue necesario observar las actividades realizadas por los especialistas del centro.

Entrevista: se emplea para obtener los requisitos funcionales de la solución propuesta y para interactuar con los especialistas que trabajan con la herramienta.

Estructura del trabajo de diploma

El documento se encuentra estructurado en tres capítulos, distribuidos de la manera siguiente:

Capítulo 1: Fundamentación teórica: Incluye un estudio acerca del estado del arte de los sistemas para el diseño y visualización de reportes para el manejo de los datos, así como los principales conceptos asociados al tema. Además, se hace referencia a los lenguajes, herramientas y metodologías que se emplean para guiar el proceso de desarrollo del Módulo de reportes para el subsistema Patrimonio de las Fuerzas Armadas Revolucionarias.

Capítulo 2: Características y diseño del módulo: Se describen las principales características a partir de artefactos de ingeniería generados, como: el modelo de dominio, los requisitos funcionales, los no funcionales y los prototipos de interfaz de

usuario. Además, todo lo referente al diseño del sistema, la arquitectura, la base de datos y los patrones de diseño.

Capítulo 3: Implementación y Pruebas: Incluye todo lo referente a la implementación de la propuesta de solución según los requisitos identificados, el diagrama de componentes y el de despliegue. Además, se definen los estándares de codificación a utilizar y se realiza la validación del módulo mediante la estrategia de pruebas definidas: Funcionales, Carga y Estrés, y validación de la idea a defender.

Capítulo 1: Fundamentación teórica.

En el presente capítulo, se define la fundamentación teórica que rige la investigación realizada acerca de las herramientas de visualización y gestión de reportes con vista al apoyo a la toma de decisiones. Se tienen presente los principales conceptos relacionados con el estudio del estado del arte y se presentan las tecnologías, lenguajes, herramientas y metodología que fueron empleados en la implementación de la propuesta de solución.

Conceptos relacionados al dominio del problema

Para lograr una mejor comprensión de la investigación, se abordan un conjunto de conceptos necesarios que están estrechamente relacionados con el dominio del problema y el objeto de estudio.

Sistema de información

(Peña, 2016) refiere que un sistema de información es un conjunto de elementos interrelacionados con el propósito de prestar atención a las demandas de información de una organización, para elevar el nivel de conocimientos que permitan un mejor apoyo a la toma de decisiones y desarrollo de acciones. Otros autores como (Peralta, 2008) de una manera más acertada definen sistema de información como: conjunto de elementos que interactúan entre sí con el fin de apoyar las actividades de una empresa o negocio, teniendo en cuenta el equipo computacional necesario para que el sistema de información pueda operar y el recurso humano que interactúa con el Sistema de Información, el cual está formado por las personas que utilizan el sistema. También, (Armando, 2010) define que un sistema de información es el sistema de personas, registros de datos y actividades que procesa los datos y la información en cierta organización, incluyendo manuales de procesos o procesos automatizados.

Por tanto, el sistema de información es el encargado de brindar información con algunos atributos importantes que ayudan a realizar alguna operación, es decir, tomar una decisión en el momento que se necesita.

Recuperación de información

La Recuperación de Información (RI) trabaja en la forma en que se representa, almacena, organiza y se accede a la información (Baeza, 1999). Otros autores plantean que la RI es dar respuesta a una pregunta formulada trayendo información relevante desde un gran archivo (Pérez-Carballo, 2000). La RI reside en establecer qué

documentos son relevantes para satisfacer las necesidades de información de un usuario determinado, expresada en un lenguaje de consulta (Tolosa, 2008).

De acuerdo con las definiciones anteriormente expuestas, los autores del presente trabajo consideran que la RI consiste en el proceso de obtener información que se encuentra almacenada y que es requerida por un usuario a través de técnicas, métodos, y herramientas de búsqueda.

Reportes

En el ámbito de la informática, los reportes son informes que organizan y exhiben la información contenida en una base de datos, según lo planteado por (Pérez-Porto, 2013). De acuerdo con las valoraciones de (Sánchez, y otros, 2016) la función de los reportes es aplicar un formato determinado a los datos para mostrarlos por medio de un diseño atractivo y que sea fácil de interpretar por los usuarios. El reporte, de esta forma, confiere una mayor utilidad a los datos. No es lo mismo trabajar con una planilla de cálculos con 10.000 campos que con un dibujo en forma de torta que presenta dichos campos de manera gráfica. (Pérez-Porto, 2013) plantea que los reportes tienen diversos niveles de complejidad, desde una lista o enumeración hasta gráficos mucho más desarrollados. Según el programa informático y la base de datos en cuestión, los reportes permiten la creación de etiquetas y la elaboración de facturas, entre otras tareas.

Pueden clasificarse como estáticos o dinámicos. Los reportes estáticos son predefinidos, que no cambian la información que poseen, mientras que los dinámicos son los que el propio usuario puede construir mediante la utilización de los filtros de selección, permitiendo que cada uno pueda definir qué información se requiere en cada reporte, bajo un criterio de búsqueda que cumpla con los requerimientos del informe (INFODEC).

Generación de reportes

Toda empresa necesita saber cómo marchan sus negocios. La generación de reportes puede proporcionarles la información que necesitan para prosperar y crecer, puesto que le comunica el rendimiento de un modo que usted pueda entender y controlar (Copyright IBM Corporation, 2010). Está especialmente desarrollada para usuarios finales, con una interfaz amistosa, lo que les permite en poco tiempo diseñar sus reportes obteniendo la información organizada de acuerdo a lo requerido.

Una cobertura completa de generación de informes implica que un sistema flexible permita que cada grupo trabaje con la información del modo que quiera, sin que se vea abrumado por un exceso de funciones. Los receptores de los informes pueden acceder a los mismos y compartirlos con facilidad en su propio idioma de trabajo. Los directivos empresariales pueden profundizar en los informes y colaborar con otros usuarios. Los ejecutivos pueden obtener paneles de control empresariales para alcanzar resúmenes rápidos de información fundamental (Copyright IBM Corporation, 2010).

Estudio de los sistemas homólogos

Actualmente, el mercado de aplicaciones informáticas es variado y abarca utilidades para desarrollar tareas en todos los campos activos de la sociedad, donde la informática juega un papel fundamental. El desarrollo de las organizaciones demanda una enorme cantidad de información, de ahí que las empresas están obligadas a tomar decisiones cada vez más precisas y con mayor rapidez.

Muchas veces se realizan SI que se dedican a capturar datos, pero que también necesitan una manera de procesarlos y mostrarlos. Es importante mencionar que los datos almacenados son útiles en la misma medida que se convierten en información para las personas que los necesitan. Para esta tarea entran los reportes, objetos que entregan información en un formato particular y permiten realizar ciertas operaciones como imprimirlos, enviarlos por correo electrónico, guardarlos a un archivo, etc. (Viegas, 2010).

Las herramientas generadoras de reportes se componen fundamentalmente de tres elementos básicos, un diseñador o editor de informes, una herramienta de administración o gestión de los reportes y una aplicación de visualización y generación de los reportes.



Ilustración 1: Ciclo de vida de los reportes.

Fuente: (Elaboración propia)

Como muestra la Ilustración 1, para soportar los procesos de creación de informes se utiliza la aplicación Diseñador de reportes que comienza el ciclo a partir de una abstracción de la fuente de datos denominada modelo semántico, que describe la estructura de los datos mientras que se encarga de la conformación del informe. Tiene como entrada el modelo semántico y genera como salida el modelo del reporte. En la fase de administración, el Administrador de reportes permite ejecutar tareas para la gestión de los modelos semánticos y los modelos de los reportes. La última etapa en el ciclo de vida de los reportes es la entrega, donde el sistema soporta este proceso mediante el Administrador de reportes, específicamente con el procesador de programación y entrega y el Visor de reportes, el cual posibilita la visualización y exportación del informe a diferentes formatos, dígame pdf, doc, xml, html, etc.

Se conoce que varios generadores de reportes presentan solo el componente de diseño y generación o motor generador, pero no garantiza un acople con la aplicación web que se quiera desarrollar. Es por eso que resulta muy necesario realizar un estudio minucioso y detallado de reportadores realizados sobre plataforma web, con el objetivo de extraer formas de solución, mecanismos y resultados de los mismos, que puedan servir de ayuda en la realización del presente trabajo.

A continuación, se muestran varias soluciones informáticas relacionadas con el objeto de estudio de la investigación, ellas son:

- Módulo de reportes: El Gran Paitití del Servicio Nacional de Sanidad Agropecuaria e Inocuidad Alimentaria (SENAGAG) de Bolivia.

- Módulo de reportes de Patrimonio del Sistema Integrado de Gestión Administrativa (SIGA) de Perú.
- Módulo de Reportes para el Sistema Único de Identificación Nacional (SUIN) de Cuba.
- Módulo de reportes de la Suite de Gestión de Proyectos la plataforma XEDRO-GESPROD de la Universidad de las Ciencias Informáticas (UCI).

El Gran Paitití

El Módulo de Reportes El GRAN PAITITÍ es un sistema informático desarrollado para la gestión de cada uno de los servicios que brinda el Servicio Nacional Sanidad Agropecuaria e Inocuidad Alimentaria de Bolivia a nivel nacional, con el objetivo de elevar la calidad de los servicios y la atención de los usuarios que a ella acceden (SENAGAG). La aplicación posee tres reportes fundamentales dentro de los cuales se encuentran subreportes que brindan toda la información referente del reporte al que pertenecen. Este sistema permite visualizar y acceder a todos los reportes que han sido elaborados para los diferentes procesos y trámites.

El módulo se especializa en la visualización de reportes sobre unidades distritales de sanidad: vegetal, animal y alimentaria tanto de exportación como importación y no en la visualización de reportes de bienes patrimoniales. Además, presenta problemas en la impresión de certificados preimpresos; debido a la posición de los márgenes y campos a imprimir, tipos de impresora, certificados elaborados en diferentes imprentas, etc., dando lugar a una emisión de certificados un tanto defectuosa en su diseño y la exportación solamente puede hacerse en formato excel. Fue modelado y diseñado según las características propias de Bolivia por tanto es una aplicación personalizada y, además, privativa. No obstante, sirvió como guía en el desarrollo de la propuesta de solución.

Módulo de reportes de Patrimonio.

El Módulo de Patrimonio presenta sus reportes mostrando la información señalada, de acuerdo a filtros de selección (SIGA: Sistema Integrado de Gestión Administrativa, 2011). Cada uno de los reportes ofrece subreportes, que permiten trabajar con la información relacionada a cada reporte original de manera organizada. Presenta un formulario de búsqueda que posibilita filtrar la información por datos o por *ítem*.

Permite sólo la exportación de sus documentos en formato .pdf e imprimirlos directamente desde la aplicación web. Este sistema fue pensado y modelado según las especificaciones dictadas por el cliente y las particularidades de Perú. No fue elaborado para uso de manera general en cualquier ambiente, se considera personalizada y privativa. Sin embargo, las consultas realizadas al módulo de reportes en la Web devuelven como resultado un conjunto de especímenes filtrado según los criterios que el usuario haya indicado y que incluye todas las columnas que éste considere convenientes, aunque se reconoce que por la similitud de los procesos que realiza, sirvió como guía en el desarrollo de la solución propuesta.

Módulo de Reportes del SUIN

El módulo de Reportes del SUIN permite el manejo de la información de forma dinámica y facilita el control sobre actividades que conforman el sistema y las personas que interactúan con él. Es independiente de forma que puede ser de utilidad para los restantes módulos que conforman el SUIN.

Queda separado del flujo principal de los demás módulos del SUIN, pero se relaciona con todos como una herramienta que permite una mejor visualización de la información solicitada. Permite adicionar, mostrar y eliminar los reportes permitiéndole al administrador incorporar un nuevo reporte al sistema o eliminarlos.

Este sistema reconoce la generación de diferentes reportes que muestra los trámites que se llevan a cabo en las oficinas de carnet de identidad. Esta herramienta fue desarrollada sobre el *framework* .NET que al ser privativo imposibilita el acceso al código fuente. No satisface las necesidades requeridas, por lo que solo se utilizó como guía en la creación de los reportes.

Módulo de reportes de la Suite de Gestión de Proyectos.

Combina el uso de una solución informática para la dirección integrada de proyectos y un sistema de formación especializada en gestión de proyectos. Posibilita la planificación y el control y seguimiento de los proyectos y de los recursos asociados a los mismos, en alineación con la proyección estratégica de las organizaciones. El núcleo del módulo es la plataforma GESPRO, que es un producto registrado en el Centro Nacional de Derecho de Autor de Cuba, No Registro CENDA 1540-2010 (Laboratorio de Investigaciones en Gestión de Proyectos, Universidad de las Ciencias Informáticas, 2015).

Presenta un módulo de reportes que permite la integración a nivel de servicios entre diferentes componentes de GESPRO para la publicación de reportes. Permite la generación de reportes desde dos escenarios diferentes: la generación de reportes desde la plataforma PATDSI y desde Libre Office 3.5 o superior. Además, presenta en el panel lateral izquierdo los reportes generales y subreportes. Admite exportar sus documentos solo en los formatos pdf y csv y presenta una ayuda para los usuarios del sistema.

Resultados de la investigación del estado del arte

El estudio a las herramientas seleccionadas para gestionar y controlar la información de reportes, permitió realizar un resumen (Ver Tabla 1) para esto se analizaron los siguientes parámetros:

Formato de exportación de los documentos: Se refiere al tipo de formato en el que son exportados los documentos.

Tipo de Aplicación: Para definir si la herramienta a evaluar es una aplicación web o una aplicación de escritorio (*Desktop*).

Tipo de Software: Se refiere al estado jurídico de la aplicación definido por su uso, modificación y distribución, esta puede ser pública, en aquellas que no necesitan un pago para ser utilizadas; o privativa, en aquellas que si lo requieren.

Tabla 1: Resumen del análisis de los sistemas homólogos

Fuente: (Elaboración propia)

Herramienta	Formato de exportación	Tipo de Aplicación	Tipo de Software
<i>El Gran Paitití de SENAGAG</i>	Excel	Web	Privativa
<i>Módulo de reportes de Patrimonio del SIGA</i>	PDF	Web	Privativa
<i>Módulo de reportes del SUIN</i>	PDF, HTML, MHT, RTF, Excel, TXT, CSV, TIFF	Web	Privativa

<i>Módulo de reportes de la Suite de Gestión de Proyectos.</i>	PDF y CSV	Web	Privativa
--	-----------	-----	-----------

Se puede concluir que las herramientas analizadas no cumplen con el objetivo de la presente investigación. Sin embargo, el análisis realizado posibilitó la identificación de funcionalidades y tecnologías que pueden contribuir al desarrollo de la propuesta de solución como filtros de selección, la implementación de subreportes dentro de los reportes y la gestión de estos dentro del propio sistema.

Selección del entorno de desarrollo para la construcción de la propuesta de solución.

Para el desarrollo de la propuesta de solución se hace necesario la definición de las herramientas, tecnologías y metodología que se emplearán en correspondencia con las usadas en el desarrollo del subsistema de Patrimonio.

Metodología de desarrollo de software

Un proceso de desarrollo de software es un conjunto de actividades definidas, organizadas y orientadas a guiar todo el ciclo de vida del desarrollo del software. Define Quién hace Qué, Cuándo y Cómo (Somerville, 2005).

La metodología de desarrollo, en ingeniería de software, es un conjunto de herramientas, técnicas, procedimientos y soporte documental encaminados a estructurar, planificar y controlar el proceso de desarrollo de forma organizada y lógica, que tiene como objetivo apoyar a los desarrolladores en la creación de un nuevo software (Zambrano, 2013).

Para guiar el desarrollo del software se utilizará *Prodesoft v1.5*, guía para la creación de aplicaciones informáticas desarrolladas en la XETID.

Prodesoft v.1.5

El Proceso de Desarrollo y Gestión de Proyectos de Software (*Prodesoft*) es un proceso que tiene como objetivo la producción eficiente de un producto de software que satisfaga los requisitos de un cliente con una planificación y una estimación de recursos predecibles.

Utiliza específicamente la combinación entre los modelos basados en Componentes y el Iterativo e Incremental. Desarrollar un software por componentes aumenta el nivel de

reutilización, robustez y facilita el mantenimiento; además, permite a los desarrolladores modificar y actualizar los componentes sin afectar otras partes del sistema. Al ser iterativo e incremental la construcción del software se realiza mediante iteraciones, donde en cada una se reducen los riesgos más significativos para el éxito del proyecto y se obtiene una versión mejorada del producto (UCID, 2012).

Prodesoft define la composición del ciclo de vida de un proyecto en cinco fases, (Ilustración 2) que comprenden todas las actividades desde los inicios del desarrollo de un nuevo producto hasta el fin de su utilidad. Estas son:

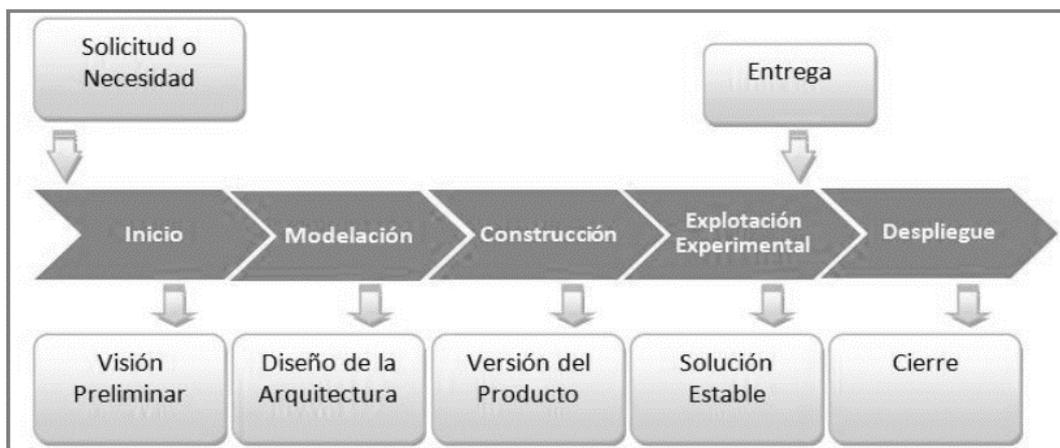


Ilustración 2: Etapas que comprende Prodesoft

Fuente: (Tomado de Prodesoft PDS UCID)

Estas fases se descomponen en iteraciones. Una iteración es un recorrido más o menos completo a través de todas las disciplinas de desarrollo (Modelación de procesos del negocio, Definición de requisitos, Diseño, Implementación y Pruebas) cada iteración va añadiendo nuevos componentes o mejorando los anteriores (UCID, 2012).

Marco de trabajo

Los *framework*¹ son de vital importancia para el desarrollo de sistemas de software orientado a objetos a gran escala. Prometen una mayor productividad y el tiempo de salida al mercado más corto a través del diseño y la reutilización del código. Un *framework* es una colección organizada de clases que constituyen un diseño reutilizable para un dominio específico de software. (GNU, 2015).

¹ Marco de trabajo

Zeolides 2.2-erp

Zeolides es un marco de trabajo que posee las características de ser libre y de código abierto, es multiplataforma, presenta un estilo arquitectónico híbrido puesto que combina varios estilos y patrones. Fomenta, además, el uso de estilos arquitectónicos como el modelo vista controlador (MVC) y el N-Capas, basada en componentes y orientadas a la nube (web). Su núcleo está desarrollado en *PHP* y *Java Script*, con algunos componentes desarrollados en *C++*, *Java* y *Python*. Utiliza como principales *frameworks*: *Ext.JS*, *Zend* y *Doctrine* (UCID). Su objetivo fundamental es el desarrollo de soluciones web de gestión empresarial.

Como el subsistema Patrimonio está desarrollado sobre el marco de trabajo *Zeolides*, el cual aporta beneficios a los usuarios por su facilidad de adaptación a los casos más complejos, se selecciona para el desarrollo de la herramienta orientada a la generación de reportes de dicho sistema.

Ext.JS v.3.4

Es un *framework JavaScript* que hace que sea relativamente fácil la creación de interfaces de usuario de estilo escritorio en una aplicación web, incluyendo varias ventanas, barras de herramientas, menús desplegables, cuadros de diálogo, etc. Tiene dos tipos de licencias, GPL y comercial, basado en componentes soportados por recursos para la programación orientada a objetos en *JavaScript*, que facilitan la implementación de extensiones y aplicaciones de gran complejidad. *ExtJS* es uno de los *frameworks* que permite flexibilizar el manejo de componentes de la página mediante el modelo de objetos para la representación de documentos (DOM, por sus siglas en inglés) y peticiones *AJAX*. Constituye una biblioteca de interfaz de usuario de cliente y además funciona como una biblioteca de desarrollo de aplicaciones (Scribd, 2016).

Doctrine v.1.2

Es un asignador objeto-relacional (ORM) para PHP, que proporciona una capa de persistencia para objetos PHP. Es una capa de abstracción que se sitúa encima de un Sistema Gestor de Base de Datos (SGBD) (Doctrine, 2006-2016). Una de las características clave de *Doctrine* es la opción de escribir las consultas de base de datos en un dialecto SQL propio orientado a objetos, llamado Lenguaje de Consulta *Doctrine* (*Doctrine Query Language* por sus siglas en inglés DQL), inspirado en *Hibernates* HQL. Además, DQL difiere ligeramente de SQL en que abstrae considerablemente la asignación entre las filas de la base de datos y objetos, permitiendo a los desarrolladores

escribir poderosas consultas de una manera sencilla y flexible (Doctrine, 2006-2016). Otra característica es el bajo nivel de configuración que necesita para empezar un proyecto. Puede generar clases a partir de una base de datos existente y después el programador puede especificar relaciones y añadir funcionalidad extra a las clases autogeneradas. No es necesario generar o mantener complejos esquemas XML de base de datos como en otros *frameworks* (Doctrine, 2006-2016).

Zend Framework v.1.7

Es un *framework* de código abierto para desarrollar aplicaciones web y servicios web con PHP 5. Cada componente está construido con una baja dependencia de otros componentes. Esta arquitectura débilmente acoplada permite a los desarrolladores utilizar los componentes por separado (Zend Technologies, 2006-2016). Ofrece un gran rendimiento, una robusta implementación MVC y una abstracción de base de datos fácil de utilizar. Empresas como *Google*, *Microsoft* y *Strikelron* se han asociado con *Zend* para proporcionar interfaces de servicios web y otras tecnologías. Cuenta con buena documentación, una comunidad muy activa, gran cantidad de componentes, incluidos algunos de *Microsoft*, *Google* y *Adobe*; además soporta *PHP 5* (Zend Technologies, 2006-2016).

Lenguaje de modelado unificado (UML)

El Lenguaje de Modelado Unificado (UML, por sus siglas en inglés) es utilizado para visualizar, construir y documentar los artefactos de un sistema, así como aspectos conceptuales: procesos de negocios o funciones del sistema y aspectos concretos como expresiones de lenguajes de programación o esquemas de bases de datos. Constituye un lenguaje de propósito general, lo que proporciona una gran flexibilidad y expresividad durante el modelado (OMG, 1997-2016).

PHP v.5.5.9

Hypertext preprocessor (PHP por sus siglas en inglés) es un lenguaje multiplataforma, de alto nivel e interpretado, compatible con la mayoría de los gestores de base de datos que se utilizan en la actualidad, es un lenguaje de programación del lado del servidor utilizado para generar páginas dinámicas. Se distribuye bajo licencia libre, por lo que se presenta como una alternativa de fácil acceso para todos. No requiere definición de tipos de variables y cuenta con manejo de excepciones (García, 2010). La sintaxis de PHP es similar a la del lenguaje C, lo cual permite su rápida ejecución y consumiendo poca

memoria. Está compuesto de un sistema principal, un conjunto de módulos y una variedad de extensiones de código (Marino, 2008).

Java Script

Java Script es un lenguaje robusto, ligero e interpretado, orientado a objetos con funciones de primera clase, más conocido como el lenguaje de *script* para páginas web, pero también usado en muchos entornos sin navegador, tales como *node.js* o *Apache CouchDB*. Es un lenguaje *script* multi-paradigma, basado en prototipos, dinámico, soporta estilos de programación funcional, orientada a objetos e imperativa. (PROJECT, 2013).

Herramientas para la generación y diseño del reporte.

Estas herramientas requieren de un componente que se integre a la plataforma *Distra* dentro del marco de trabajo de la empresa *Zeolides* y no como una herramienta independiente a esta, pudiéndose desarrollar soluciones bajo los principios, estándares y el entorno que controlan y guían el desarrollo de software.

JasperReport Server v.5.6.0

Es un motor de reportes que puede ser embebido en todo tipo de aplicaciones, desde las que generan reportes a partir de una plantilla o modelo predeterminado, hasta las que brindan más libertad al usuario para diseñar sus propios reportes y ejecutar otras operaciones complejas. Es una librería implementada completamente en *Java*, brinda el máximo nivel de portabilidad y con un amplio y expandible grupo de posibles fuentes de datos. Posee además una gran variedad de formatos de salida y exportación, así como una gran comunidad que mantiene y desarrolla la librería. Genera informes para formatos de impresión predeterminados existentes o para reportes continuos a ser visualizados en la web. Los reportes pueden ser exportados a formatos como: PDF, XML, HTML, CSV, XLS, RTF, TXT.

A través de los subreportes es posible manipular y diseñar reportes con un diseño altamente complejo. Soporta a la misma vez varios orígenes de datos incluso aunque sean de tipos distintos. Es posible enviar parámetros desde una aplicación a *JasperReports*; esto es muy simple de implementar y brinda una herramienta muy poderosa ya que permite clasificar, restringir o mejorar los datos que son enviados al usuario basándose en las condiciones de tiempo de ejecución (Jasperforge.org, 2008).

Puede ser utilizado en cualquier entorno o sistema operativo siempre que exista una implementación de la máquina virtual de *Java* para dicho entorno, las únicas dependencias que deben satisfacerse son: *Java Development Kit* (JDK) 1.3 o superior y JDBC 2.0 en caso que se utilicen fuentes de datos relacionales.

iReport v.5.6.0

La herramienta *iReport* es un constructor / diseñador de informes visual, poderoso, intuitivo y fácil de usar para *JasperReports* escrito en *Java*. Este instrumento permite que los usuarios corrijan visualmente informes complejos con cartas, imágenes, subinformes, etc. Los datos para imprimir pueden ser recuperados por varios caminos incluso múltiples uniones JDBC, *Table Models*, *JavaBeans*, XML, etc. (Herrera).

Características de *iReport*:

La lista siguiente describe algunas de las características importantes de *iReport*:

- 100% escrito en *java* y además *open source* y gratuito.
- Maneja el 98% de las etiquetas de *JasperReports*.
- Permite diseñar con sus propias herramientas: rectángulos, líneas, elipses, campos de los *textfield*s, cartas, subreportes.
- Soporta internacionalización nativamente.
- *Browser* de la estructura del documento.
- Recopilador y exportador integrados.
- Soporta JDBC.
- Soporta *JavaBeans* como orígenes de datos (éstos deben implementar la interface *JRDataSource*).
- Incluye asistentes para crear automáticamente informes.
- Tiene asistentes para generar los subreportes
- Tiene asistentes para las plantillas.
- Facilidad de instalación.

Control de versiones *TortoiseSVN* v.1.10.0

Es un cliente gratuito de código abierto para el sistema de control de versiones *Subversion*. Esto significa que *TortoiseSVN* administra archivos y directorios a lo largo del tiempo. Le permite al usuario recuperar versiones antiguas de sus archivos y examinar la historia de cómo y cuándo cambiaron sus datos, y quién hizo el cambio. Es un proyecto de Código Abierto desarrollado bajo la licencia GNU *General Public License* (GLP) (Küng, y otros, 2018).

Se encuentra disponible en 28 idiomas y puede ser usado sin un entorno de desarrollo. Está basado en *Apache* y proporciona una interfaz sencilla y fácil para *Subversion*. El código fuente también está disponible de manera gratuita.

Herramientas de modelado.

El uso de una eficiente herramienta de modelado ofrece todos los elementos necesarios para la realización de un trabajo eficiente. Permiten capturar, guardar, rechazar, integrar automáticamente información y diseñar la documentación: la creación de diagramas y el diseño de modelos para garantizar el éxito del proyecto de desarrollo de software.

Visual Paradigm v.8.0

Las herramientas CASE (*Computer Aided Software Engineering*, Ingeniería de Software Asistida por Ordenador) se utiliza para una rápida construcción de aplicaciones. Permite construir diagramas como el diagrama de proceso del negocio y el diagrama de despliegue. Se considera muy completa y fácil de usar, con soporte multiplataforma (Díaz, 2009).

El empleo de herramientas CASE permite integrar durante todo el proceso de ciclo de vida del proyecto aspectos como:

- Análisis de datos y procesos integrados mediante un repositorio.
- Generación de interfaces entre el análisis y el diseño.
- Generación del código a partir del diseño.
- Control de mantenimiento.

Netbeans v.8.0

Entorno de desarrollo de código abierto y gratuito disponible para *Windows*, *Mac*, *Linux* y *Solaris*, que permite a los programadores escribir, compilar, depurar y ejecutar programas. Brinda soporte a tecnologías como: *Java*, *PHP*, *Groovy*, *C/C++*, *HTML5*. Suele dar soporte a casi todas las novedades en el lenguaje *Java*. Permite la creación y configuración de distintos proyectos, incluida la elección de algunos *frameworks*. Desde el propio *Netbeans* se puede conectar a distintos sistemas gestores de bases de datos, como pueden ser *Oracle*, *MySQL*, *PostgreSQL* y demás, ver las tablas, realizar consultas y modificaciones y todo ello integrado en el propio IDE. Se integra con diversos servidores de aplicaciones como *Apache Tomcat*, *GlassFish*, *JBoss*, *WebLogic*, *Sailfin*, *Sun Java System Application Server* y otros. Permite que las aplicaciones sean desarrolladas a partir de un conjunto de componentes modulares de

software llamados módulos y provee soporte para aplicaciones de Arquitectura Orientada a Servicios (SOA, por sus siglas en inglés) (PostgreSQL-Es, 2013).

Por todas las características anteriormente mencionadas es el IDE escogido para el desarrollo de la aplicación.

Java SE Development Kit (JDK) v.1.7.0

Java Development Kit (JDK) es un software que provee herramientas de desarrollo para la creación de programas en *Java* (BYSPEL, 2017). Puede instalarse en una computadora local o en una unidad de red. Es la máquina virtual sin la cual *Netbeans* no funcionaría.

PostgreSQL v.9.4

Es un sistema de gestión de bases de datos objeto-relacional, distribuido bajo licencia BSD y con su código fuente disponible libremente. Es el sistema de gestión de bases de datos de código abierto más potente del mercado y en sus últimas versiones no tiene nada que envidiarles a otras bases de datos comerciales, utiliza un modelo cliente/servidor para garantizar la estabilidad del sistema. Un fallo en uno de los procesos no afectará el resto y el sistema continuará funcionando (Object Management Group, 2015). Se usa este gestor de base de datos porque se asimila como tecnología del subsistema Patrimonio.

pgAdmin III v.1.18.1

Es una herramienta de código abierto para la administración de bases de datos *PostgreSQL*. Incluye una interfaz administrativa gráfica y herramientas de consulta SQL, así como un editor de código procedural. Está diseñado para responder a las necesidades de la mayoría de los usuarios, desde escribir simples consultas SQL hasta desarrollar bases de datos complejas. La interfaz gráfica soporta todas las características de *PostgreSQL* y hace simple la administración. Está disponible en más de una docena de lenguajes y para varios sistemas operativos, incluyendo *Microsoft Windows*, *Linux*, *Mac* y *Solaris* (PostgreSQL tools).

Apache HTTP Server v.2.3.1

Es un servidor web HTTP de código abierto, que publica su código fuente permitiendo a cualquiera modificarlo y colaborar así a su desarrollo. Está diseñado para ser potente y flexible y que funcione para una amplia variedad de plataformas, *Unix* (GNU/Linux), *Microsoft Windows*, *Macintosh* y otras que implementa el protocolo HTTP. Trabaja con

lenguajes como *Perl*, *PHP* y otros. Tiene capacidad para servir páginas tanto de contenido estático como de contenido dinámico a través de otras herramientas soportadas (Alan, 2015).

Apache JMeter v.2.12

El *Apache JMeter* es un software de código abierto, una aplicación diseñada totalmente en *JAVA* para medir el rendimiento y comportamiento de servidores mediante pruebas. Se utiliza para probar el rendimiento tanto de los recursos estáticos y dinámicos archivos, *Servlets*, *scripts* de *Perl*, objetos *Java*, bases de datos - consultas, servidores FTP y mucho más). Además, se utiliza para simular una carga pesada en un servidor, la red o un objeto para poner a prueba su resistencia o para analizar el rendimiento global en diferentes tipos de carga. Se usa, también, para hacer un análisis gráfico de rendimiento o para probar el comportamiento de diferentes elementos con un gran volumen de carga y concurrencia.

Conclusiones del capítulo

En este capítulo se trataron los elementos teóricos que dan sustento a la propuesta de solución del problema planteado, arribando a las siguientes conclusiones:

- La investigación realizada al entorno del problema científico permitió conocer los principales conceptos asociados al objeto de estudio.
- Con el estudio de las soluciones existentes se determinaron las principales funcionalidades para el desarrollo del módulo de reportes para el subsistema Patrimonio de las FAR.
- Las herramientas, tecnologías y lenguajes permiten sentar las bases para el desarrollo de la propuesta de solución.

Capítulo 2: Características y diseño del módulo

En el presente capítulo se realiza un análisis de las características y diseño del módulo a desarrollar a partir de la problemática planteada. Se describen los principales artefactos de la ingeniería de software como el modelo del dominio, la propuesta de solución, la especificación de los requisitos funcionales y no funcionales. También se presentan el modelo de dominio, la arquitectura, los patrones de diseño y la base de datos.

Modelo de dominio

La propuesta de solución va dirigida a beneficiar al MINFAR. El proceso de negocio de los reportes no está bien definido, pues no se ven claramente las fronteras del mismo, así como las personas que lo inician, por tanto, se arriba a la conclusión de que no necesita de un modelado completo, por lo que se propone un modelo de dominio, para ayudar a comprender los conceptos que trabajan los especialistas y con los que deberá trabajar la aplicación.

A continuación, se definen todos los conceptos a utilizar en el diagrama como primer paso para la modelación del modelo de dominio:

MINFAR: Ministerio de las Fuerzas Armadas Revolucionarias que atiende todos los procesos del patrimonio de las FAR.

Ejército: Entidad que supervisa todos los procesos desarrollados con el patrimonio de las Regiones Militares subordinadas a estos.

Región_Militar: Entidad que supervisa todos los procesos desarrollados con el patrimonio de las Unidades Militares subordinadas a estos.

Unidad_Militar: Entidad que supervisa todos los procesos desarrollados con el patrimonio en cada una de sus áreas.

XETID: Empresa que lleva el control del patrimonio de las FAR.

Área_Logística: Área que controla todos los procesos logísticos de la empresa.

Distra: Plataforma de planificación y gestión de los procesos de la empresa.

Subsistema_Activos_Fijos: Sistema que controla y registra los activos fijos de la empresa.

Gestión_documento: Se encarga de la gestión de los documentos de los inmuebles y los objetos de obra.

Gestión_inmueble: Se encarga de la gestión de los inmuebles y los objetos de obra.

Subsistema_Patrimonio: Sistema que controla y registra los inmuebles del patrimonio de las FAR.

Módulo_reportes: guarda todos los reportes de patrimonio.

Sistema_Empresarial: Especialidad que contiene todos los inmuebles y objetos de obra de sistema empresarial.

Vivienda: Especialidad que contiene todos los inmuebles y objetos de obra de vivienda.

Campamento: Especialidad que contiene todos los inmuebles y objetos de obra de construcción y alojamiento.

Ingeniería: Especialidad que contiene todos los inmuebles y objetos de obra de ingeniería.

Especialista_XETID: Encargados de confeccionar los reportes.

Especialista_MINFAR: Encargados de solicitar los reportes.

Reporte: Documento generado por el sistema.

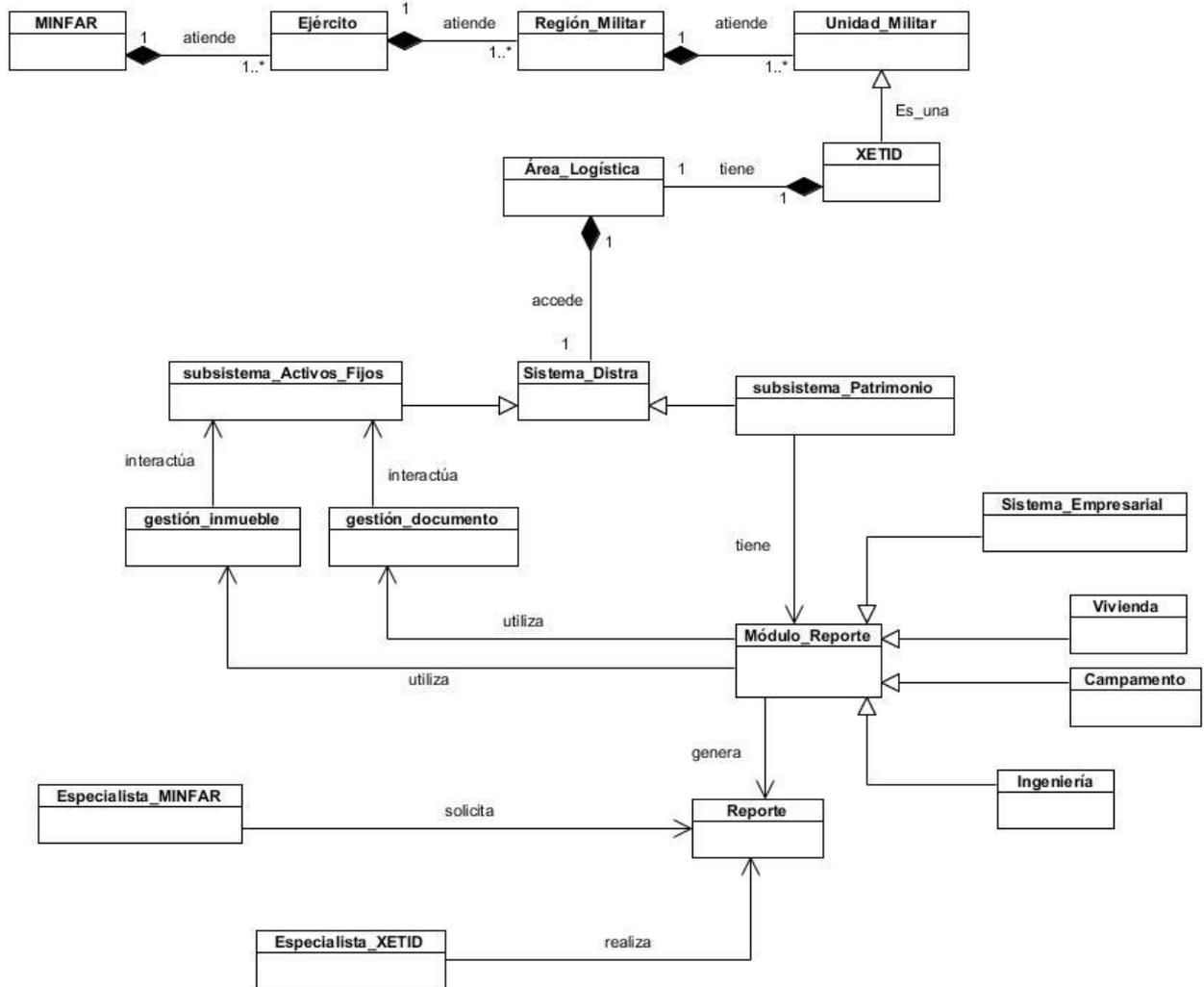


Ilustración 3: Modelo de dominio

Fuente: (Elaboración propia)

Descripción del modelo de dominio.

El MINFAR atiende los Ejércitos, el Ejército a las Regiones Militares subordinadas y las Regiones Militares a las Unidades Militares. La XETID es una unidad militar que se encarga de informatizar todo lo relacionado con el control del patrimonio de las FAR. Contiene un área de Logística, donde se desarrollan soluciones para la cadena de suministros y desde la cual se accede a Distra, la plataforma de gestión y planificación empresarial. Dentro de Distra se pueden encontrar los subsistemas Activos Fijos y Patrimonio. En el segundo se encuentra la gestión de documentos y la gestión de inmuebles que interactúan con el subsistema de activos fijos. Además, se puede encontrar el módulo de reportes que recopila la información de las especialidades: Campamento, Ingeniería, Sistema Empresarial y Vivienda utilizando la gestión de

documento y la gestión de inmuebles para generar reportes que son solicitados por los especialistas del MINFAR y confeccionados por los especialistas de la XETID.

Propuesta de solución.

La solución que se propone permitirá a los especialistas del centro realizar los reportes con mayor calidad a la hora de presentarlos al cliente. Además, recopila la información gestionada en las especialidades, haciendo un resumen de los datos solicitados. Funciona, también, como un repositorio de reportes, permitiendo almacenar los reportes para futuras solicitudes.

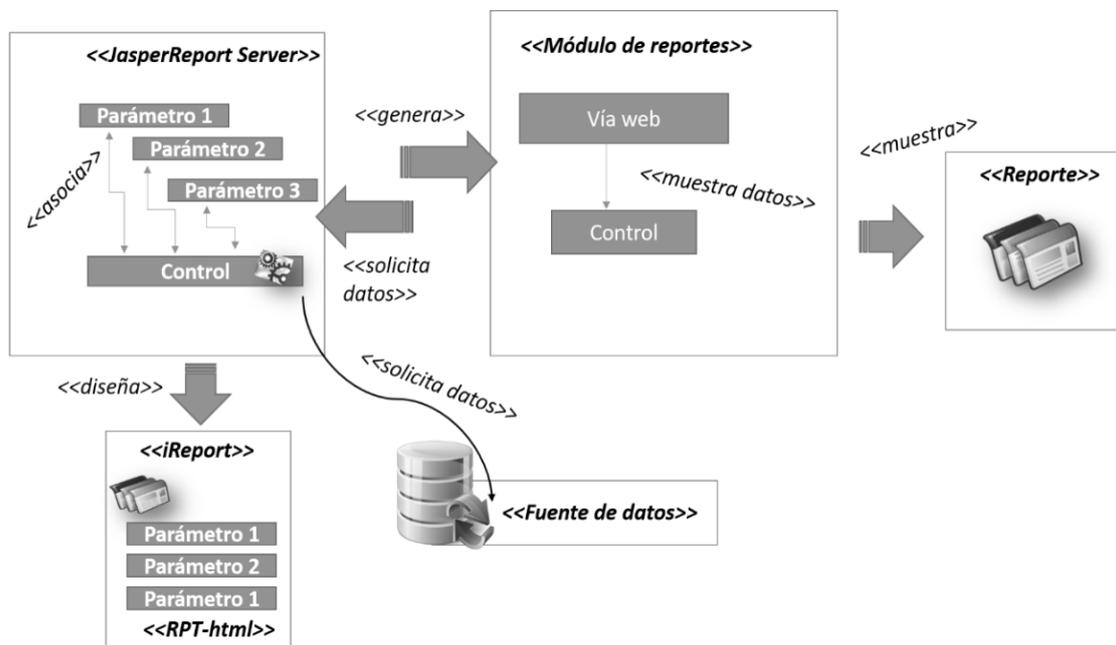


Ilustración 4: Funcionamiento de la propuesta de solución

Fuente: (Elaboración propia)

Como muestra la Ilustración 4, el módulo de los reportes solicita los datos al generador de reportes *JasperReport Server*, quien asocia cada uno de los parámetro del reporte diseñado con los datos de la fuente de datos. Luego, genera el reporte vía web al sistema y muestra, al cliente, a través del módulo, el reporte solicitado que puede ser imprimido desde la misma aplicación web y ser exportado a los formatos.pdf, .docx, .xls y .xlsx.

Especificación de requisitos funcionales.

Un requerimiento funcional, según la IEEE *Standard Glossary of Software Engineering Terminology*, se define como una “condición o capacidad que necesita un usuario para resolver un problema o lograr un objetivo” (Corporation, 2006). Para el desarrollo de la

aplicación se deben describir estas funcionalidades con el objetivo de lograr un mayor entendimiento entre clientes y desarrolladores como premisa para lograr un producto con la calidad requerida.

Tabla 2: Catálogo de requisitos funcionales

Fuente: (Elaboración propia)

Requisitos	Prioridad
RF 1: Conformar reportes especialidad de Campamento. RF 1.1: Conformar reporte Situación documental. RF 1.2: Conformar reporte Situación del registro, valor y certificado contable. RF 1.3: Conformar reporte Estado del registro contable de los bienes inmuebles de las FAR	Alta
RF 2: Conformar reportes especialidad de Ingeniería RF 2.1: Conformar reporte Situación documental. RF 2.2: Conformar reporte Situación del registro, valor y certificado contable. RF 2.3: Conformar reporte Estado del registro contable de los bienes inmuebles de las FAR	Alta
RF 3: Conformar reportes especialidad de Sistema Empresarial RF 3.1: Conformar reporte Situación documental. RF 3.2: Conformar reporte Situación del registro, valor y certificado contable. RF 3.3: Conformar reporte Estado del registro contable de los bienes inmuebles de las FAR	Alta
RF 4: Conformar reportes especialidad de Vivienda RF 4.1: Conformar reporte Situación documental. RF 4.2: Conformar reporte Situación del registro, valor y certificado contable. RF 4.3: Conformar reporte Estado del registro contable de los bienes inmuebles de las FAR.	Alta
RF 5: Mostrar reporte	Alta
RF 6: Actualizar reporte	Alto
RF 7: Exportar reporte a: -PDF	Medio

-DOCX -XLS -XLSX	
RF 8: Imprimir reporte	Medio
RF 9: Limpiar datos	Baja

Concepción de los modelos de procesos.

A través del modelado de las actividades y procesos puede lograrse un mejor entendimiento del negocio y muchas veces esto presenta la oportunidad de mejorarlos. A continuación, se muestra la especificación del RF 4: Generar reporte especialidad de Vivienda.

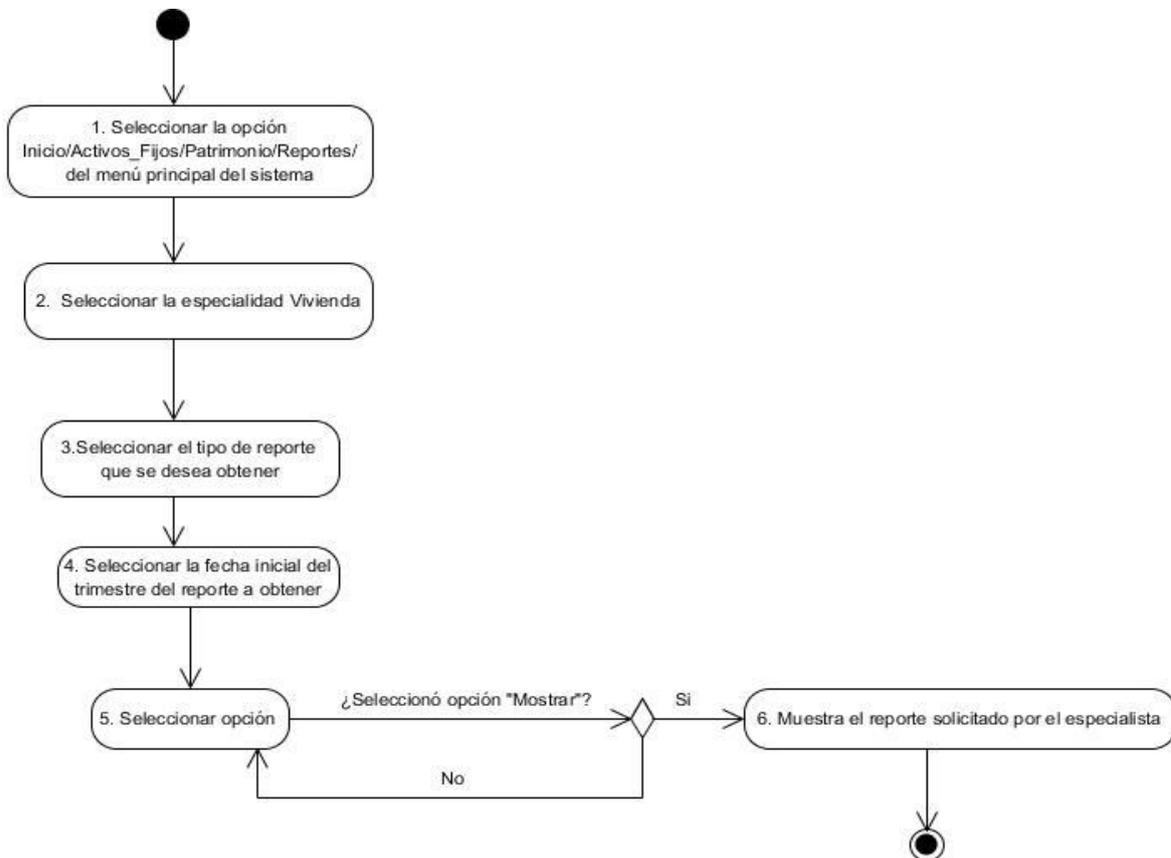


Ilustración 5: Especificación del RF 4: Generar reporte especialidad de Vivienda

Fuente: (Elaboración propia)

La Ilustración 5 muestra el flujo de actividades que se llevan a cabo en la funcionalidad Generar reporte especialidad de Vivienda. Este proceso da inicio cuando el especialista entra al sistema y selecciona la especialidad Vivienda, el tipo de reporte a obtener de dicha especialidad y la fecha inicial del trimestre. La aplicación mostrará

automáticamente la fecha final del trimestre a partir de la inicial. Luego, el especialista cuenta con dos botones: “Limpiar” y “Mostrar”. El primero le permitirá limpiar los datos seleccionados en caso de haber cometido algún error, mientras que el segundo le mostrará en el visor de reportes el deseado. Tendrá las opciones “Actualizar”: que le permitirá actualizar los datos del reporte, obteniéndolo en su forma más actualizada, “Imprimir” desde la misma aplicación web y “Exportar” los reportes en los formatos pdf, xls, xlsx y docx. Para observar los demás, ver [Anexo 2](#).

Prototipos de interfaz de usuario.

El prototipo de interfaz de usuario (PIU) se desarrolla a partir de la especificación de requisitos. Constituye una representación previa del software. En este se definen las vistas, los iconos, el estilo de las ventanas y menús, así como las acciones de la interfaz y los objetos, de manera que se muestra al usuario el cual podrá comunicar sus inconformidades.

A continuación, se muestra el prototipo de interfaz de usuario no funcional PIU “Principal” (Ilustración 6).

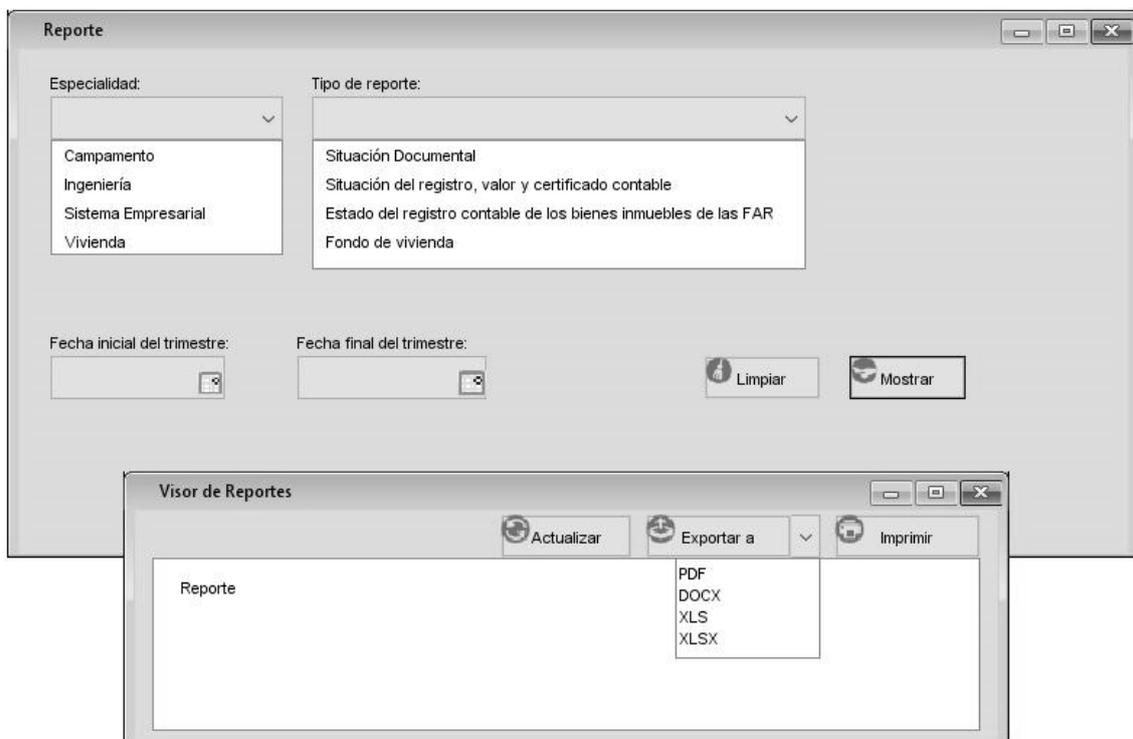


Ilustración 6: PIU “Principal”

Fuente: (Elaboración propia)

Especificación de requisitos no funcionales.

Los requisitos no funcionales (RNF) son propiedades que hacen al producto atractivo, usable, rápido y confiable. Son los encargados de darle el acabado al producto para que esté listo para usarse (Metodología Gestión de Requerimientos, 2010).

Se identificaron un total de 12 requisitos no funcionales enmarcados en grupos, que a continuación se presentan:

1. Rendimiento.

RNF 1: A partir de la arquitectura sobre la que se diseña el producto, el tiempo de respuesta del módulo y la velocidad de procesamiento de la información se minimiza.

2. Soporte.

RNF 2: Se utiliza como servidor Web *Apache*, para soportar grandes volúmenes de información el sistema gestor de base de datos *PostgreSQL* y como servidor de reportes *JasperReport Server*. Debe elaborarse un paquete de instalación de los nuevos componentes.

3. Restricciones de diseño.

RNF 3: El producto de software final se diseña sobre la arquitectura cliente-servidor. Se emplean los estándares establecidos (diseño de interfaces, base de datos y codificación). Se emplea como lenguaje de programación del lado del servidor PHP, mientras que por el lado del cliente *JavaScript*, XHTML y CSS.

4. Interfaz.

RNF 4: El módulo cuenta con una interfaz fácil de usar, sencilla, amigable, permitiendo que los usuarios sean capaces de interactuar con la aplicación aun teniendo conocimientos básicos de informática. Está diseñada para adaptarse a la resolución del usuario, utilizando colores refrescantes, agradables y se emplean imágenes identificadas con el negocio donde éste se utiliza.

5. Portabilidad.

RNF 5: El módulo es multiplataforma y es compatible con los sistemas operativos *Windows* y *Linux*.

6. Legales.

RNF 6: El módulo se ajusta y rige por las leyes, decretos, decretos leyes, resoluciones y manuales (órdenes) establecidos, que norman los procesos que serán automatizados:

- Resolución 151 del Ministro de la FAR, 2006.
- Orden 9 del Ministro de las FAR, 2015.

7. Seguridad.

RNF 7: El usuario debe autenticarse antes de entrar al sistema.

7.1. Confiabilidad

RNF 7.1: La información que se maneje en el módulo está protegida de acceso no autorizado y divulgación, a partir de los diferentes roles de los usuarios que empleen el módulo.

7.2. Integridad:

RNF 7.2: La información manejada por el módulo es objeto de cuidadosa protección contra corrupción y estados inconsistentes, de igual manera el origen y autoridad de los datos.

7.3. Disponibilidad:

RNF 7.3: La información se encuentra disponible en todo momento para aquellos usuarios autorizados a acceder al sistema.

8. Software.

- RNF 8: Para el cliente:
 1. Navegador Mozilla Firefox 52.0 o superior.
- RNF 9: Para el servidor:
 1. Servidor web *Apache* v2.0 o superior con *PHP* 5.5.9 instalado disponible, debe estar configurado con las extensiones *PDO*, *PDO_pgsql*, *pgsql* y *soap*.
 2. Gestor de base de datos: *PostgreSQL* v9.3 o superior.

9. Hardware.

- RNF 10: Para el cliente:

1. Tarjeta de red.
 2. Impresora.
 3. Procesador Pentium III a 1.33GHz con 512 MB de memoria RAM como mínimo.
- RNF 11: Para el servidor:
 1. Procesador Intel Xeon E5-2620 a 2.4 GHz 8 GB de memoria RAM virtualizado como mínimo.
 2. Al menos 40GB de espacio libre en disco duro.

10. Aplicación de Estándares.

RNF 12: Se utilizan los estándares de codificación, estándares de diseño para la base de datos y mecanismos de diseño definidos por la XETID.

Arquitectura de la solución

El Módulo de reportes del subsistema Patrimonio al estar desarrollado sobre el marco de trabajo *Zeolides* hereda la arquitectura N-Capas, definiéndose 4 capas que a continuación se explican.

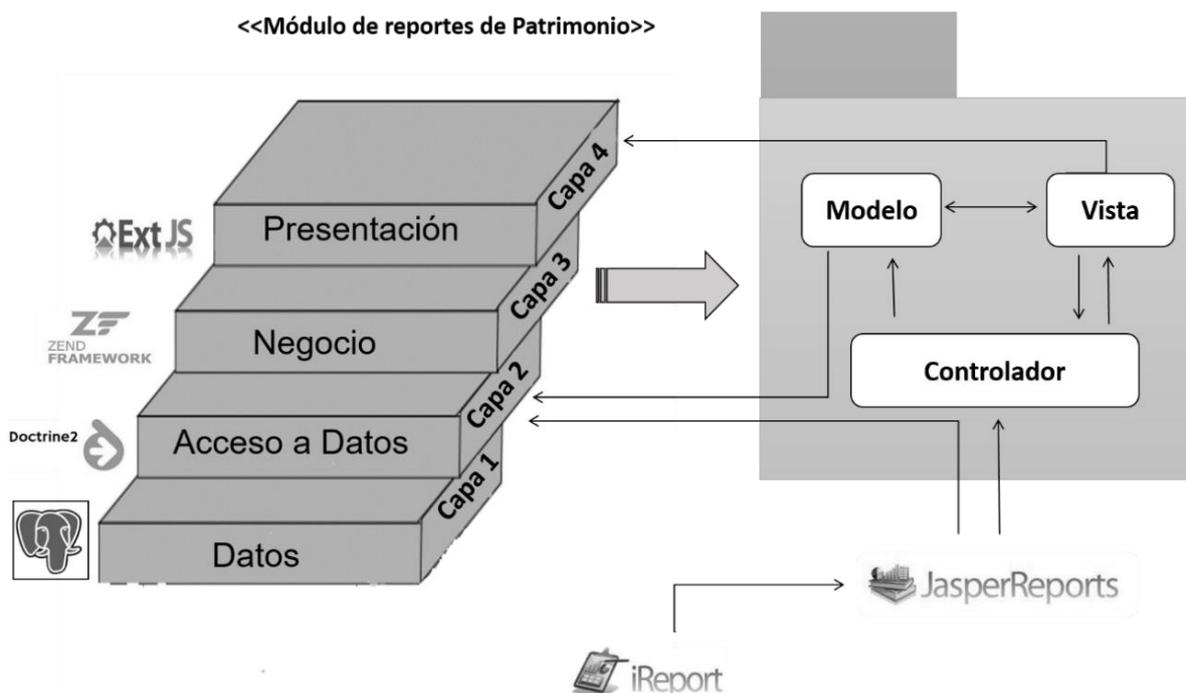


Ilustración 7: Arquitectura de la propuesta de solución

Fuente: (Elaboración propia)

Una primera capa para los datos maneja toda la información de la fuente de datos. La segunda capa se encarga del acceso a los datos mediante *Doctrine*. La tercera capa es la encargada de los procesos del negocio donde se desarrolla, a través de un MVC y de *Zend Framework*, la gestión de los reportes y una cuarta capa de presentación donde se visualiza el módulo mediante *EXT.JS*. La herramienta *JasperReport* se conecta con el controlador en el MVC en la capa de negocio y con la capa de acceso a los datos que permite el diseño de los reportes a partir de *iReport*. En el modelo se puede encontrar la clase *ReportesModel.php* que se conecta con la capa de acceso a datos. En la vista se encuentran las clases *reportes.principal.js* y *reportes.js* y se conecta con la capa de presentación y en el controlador la clase *ReportesController.php*.

Modelo del diseño

El modelo del diseño se centra en cómo los requisitos funcionales y no funcionales junto con otras restricciones relacionadas con el entorno de implementación tienen impacto en el módulo a considerar. Además, sirve de abstracción en la implementación del módulo, de ese modo es utilizada como una entrada fundamental de las actividades de implementación (Somerville, 2005).

Diagramas de clase del diseño

El diagrama de clases del diseño describe gráficamente las especificaciones de las clases de software y de las interfaces en una aplicación (Larman, 1999). A continuación, se presenta el diagrama de clases del diseño RF 1: Generar reporte especialidad de Vivienda (Ilustración 8) y los estereotipos web (Tabla 3) seleccionados:

Tabla 3: Estereotipos web utilizados

Fuente: (Elaboración propia)

Estereotipos para las clases	
Estereotipos	Descripción
 Client Page	Representan páginas que son dibujadas por el navegador web y pueden ser una combinación de algún o algunos lenguajes de marcado, <i>scripts</i> del lado del cliente, islas de datos, etc.

 Server Page	Representa una página web que tiene scripts ejecutados por el servidor. Estos <i>scripts</i> interactúan con los recursos que se encuentran al alcance del servidor. Sólo puede mantener relaciones con objetos que se encuentren en el servidor.
 Form	Representa una colección de campos de entrada que forman parte con una página del lado cliente (<i>Client Page</i>). Tiene una correspondencia directa con la etiqueta <FORM> de HTML.
Estereotipos para las Relaciones entre las Clases	
Link	Representa un apuntador desde una " <i>client page</i> " hacia una " <i>client page</i> " o " <i>server page</i> ". Corresponde directamente con una etiqueta <a> de HTML
Submit	Esta relación siempre se da entre una " <i>form</i> " y una " <i>server page</i> ", por supuesto, la " <i>server page</i> " procesa los datos que la " <i>form</i> " le envía (<i>submits</i>)
Build	Sirve para identificar cuales " <i>server page</i> " son responsables de la creación de una " <i>client page</i> ". Una " <i>server page</i> " puede crear varias " <i>client page</i> ", pero una " <i>client page</i> " sólo puede ser creada por una sola " <i>server page</i> ". Esta relación siempre es unidireccional

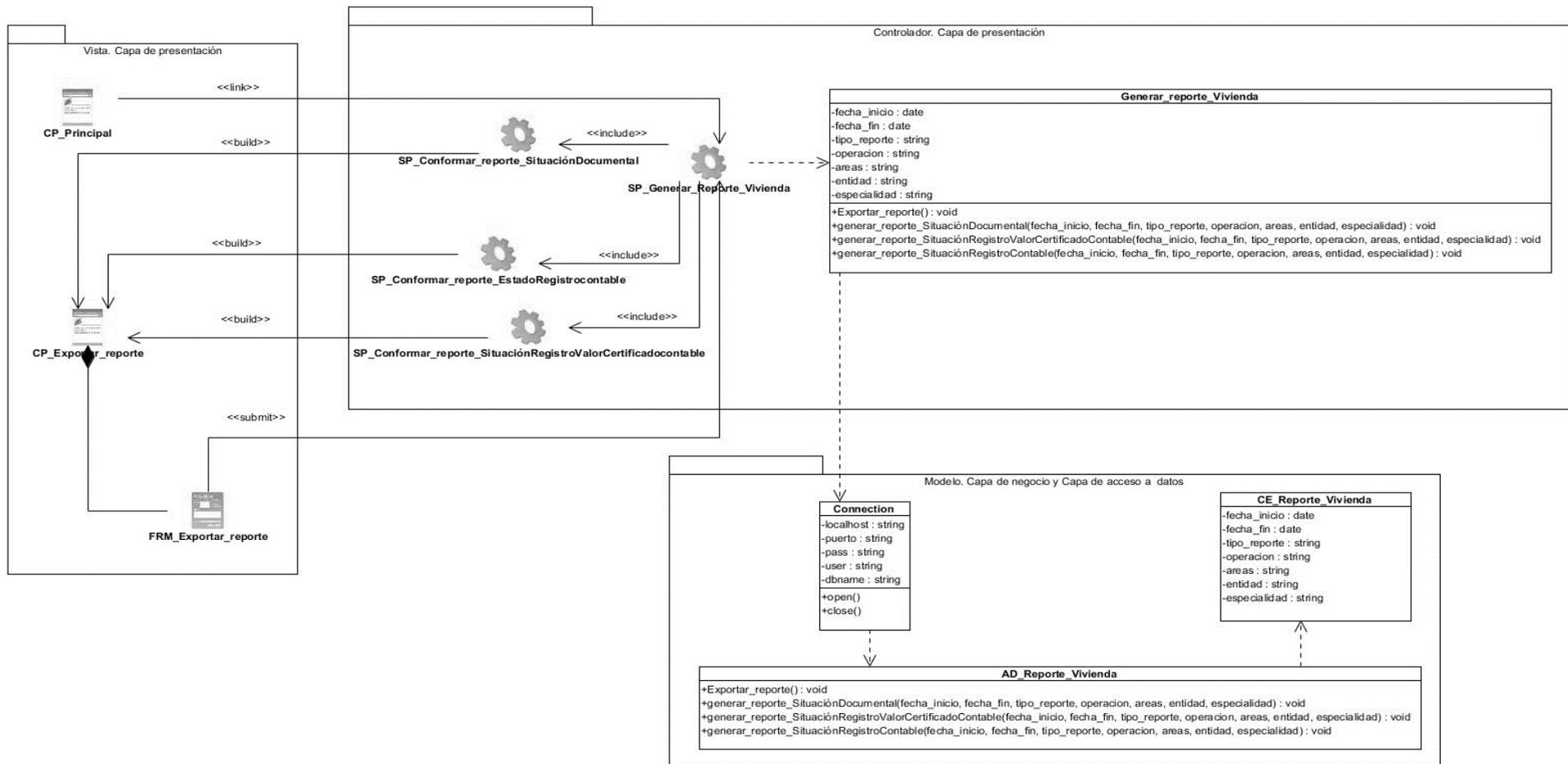


Ilustración 8: Diagrama de clases con estereotipos web del RF Generar reporte especialidad Vivienda

Fuente: (Elaboración propia)

La siguiente Ilustración 8 muestra el diagrama de clases con estereotipos web del RF Generar reporte especialidad Vivienda donde se presentan los estereotipos web utilizados. Para observar los diagramas de clases con estereotipos web ver [Anexo 3](#).

Diseño de la base de datos

Como método de diseño de base de datos se elabora el modelo de datos, el cual representa la información persistente del software mediante entidades definidas por atributos y las relaciones existentes entre ellas.

El modelo de datos es una colección de herramientas conceptuales que se emplean para especificar datos, las relaciones entre ellos, su semántica asociada y las restricciones de integridad (Zorrilla, 2014). Permite describir la base de datos, determinar su estructura lógica, sus relaciones y sus tipos, mostrando la realidad deseada. La Ilustración 9 representa un fragmento del modelo de datos del módulo. Indirectamente se relacionan con la aplicación todas las entidades pertenecientes al subsistema Patrimonio, de donde se obtendrán los datos para mostrar los reportes.

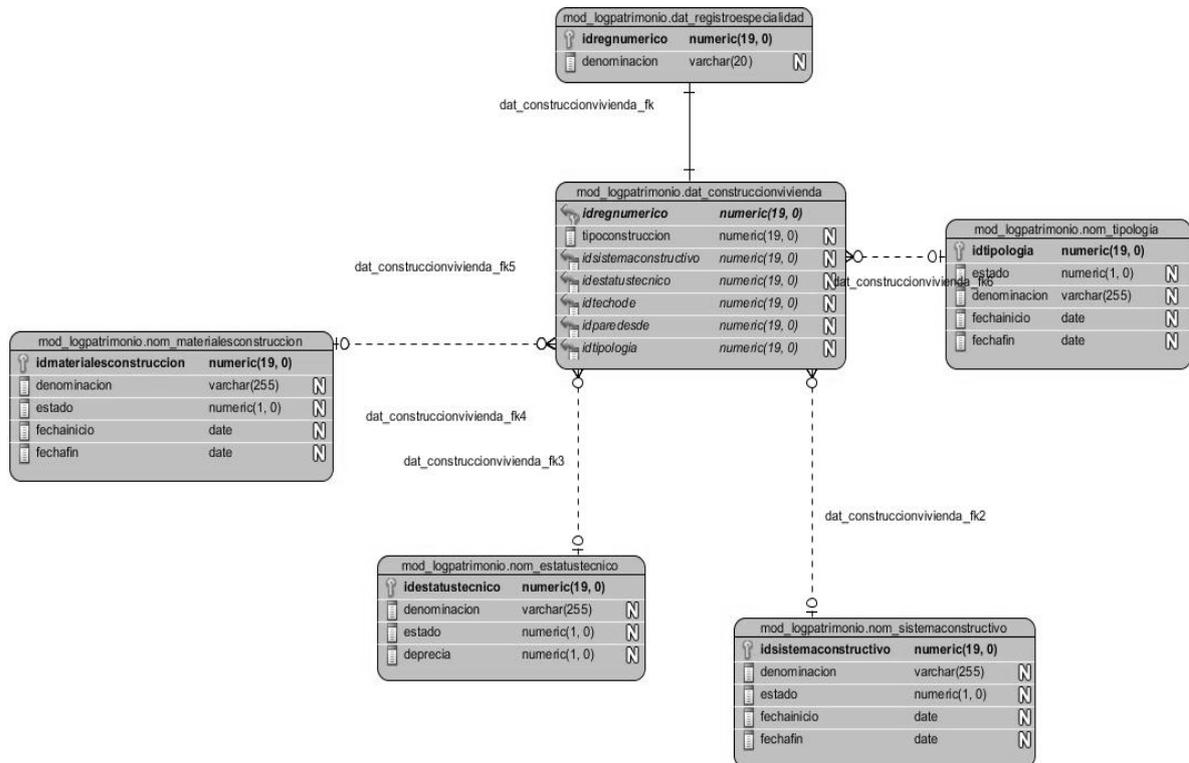


Ilustración 9: Diseño del modelo de datos

Fuente: (Tomado de la base de datos de Patrimonio)

Patrones de diseño

El desarrollo de un sistema conlleva, en la mayoría de las ocasiones, a darle solución a problemas muy complejos que ya alguien más ha resuelto. Uno de los pasos a tener en cuenta cuando se decide desarrollar un proyecto de software es identificar que patrones pueden ser utilizados. Entiéndase por patrón como una solución estándar para un problema común de programación.

Los patrones de diseño son la base para la búsqueda de soluciones a problemas comunes en el desarrollo de software (Figueroa, 2016).

En el mundo del desarrollo de software, los patrones constituyen un elemento fundamental a la hora de diseñar. Permiten usar soluciones que ya están probadas, reutilizar conocimiento y aprovechar experiencias que logran ser documentadas y catalogadas. Además, con el empleo de los patrones se logra una mejor comunicación entre diseñadores y facilitan el aprendizaje a los diseñadores inexpertos. A continuación, se muestran varios de los patrones aplicados en el módulo de generación de reportes:

Patrones GRASP

Los patrones *GRASP*², como expresan sus siglas en inglés, son patrones generales de software para asignación de responsabilidades a objetos. (Microsoft, 2016). Los patrones *GRASP* que se utilizaron son:

Experto: Es el principio básico de asignación de responsabilidades. Se conserva el encapsulamiento ya que los objetos se valen de su propia información para hacer lo que se les pide. Esto soporta un bajo acoplamiento, lo que favorece el hecho de tener sistemas más robustos y de fácil mantenimiento. El comportamiento se distribuye entre las clases que cuentan con la información requerida, alentando con ello definiciones de clase “sencillas” y más cohesivas que son más fáciles de comprender y de mantener.

```
class ReportesModel extends ZendExt_Model { //Se utiliza esta clase en Distra
}
    public function ReportesModel() {
        parent::ZendExt_Model();
    }
}
```

Creador: Guía la asignación de responsabilidades relacionadas con la creación de objetos, tarea muy frecuente en los sistemas orientados a objetos. La nueva instancia deberá ser creada por la clase que contenga la información necesaria para realizar la

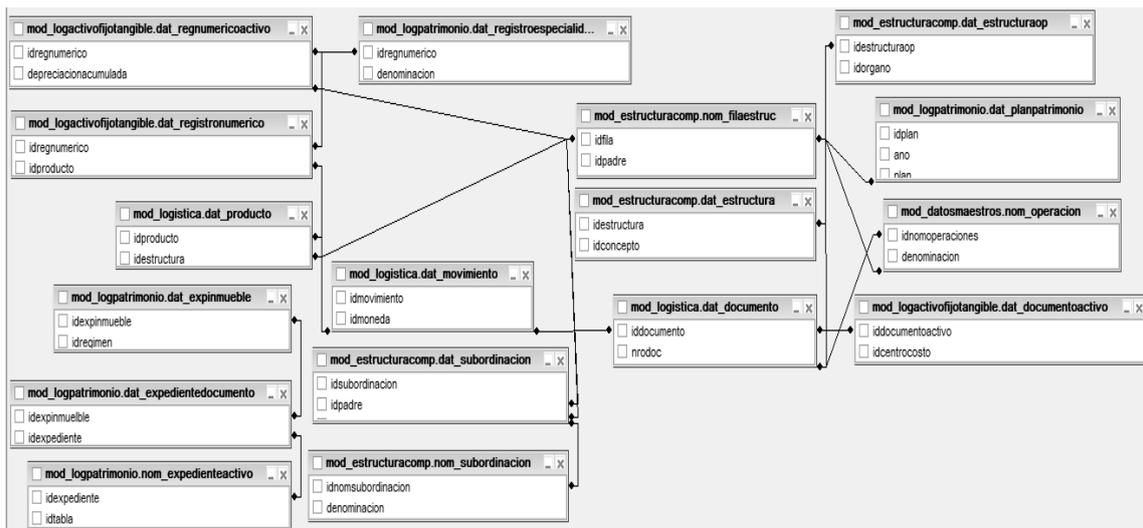
² Por sus siglas en inglés: General Responsibility Assignment Software Patterns.

creación del objeto, use directamente las instancias creadas del objeto, almacene o maneje varias instancias de la clase o contenga o agregue la clase.

```

this.cmbReporte = new Ext.form.ComboBox({
    name: 'idespecialidad',
    fieldLabel: _this.idioma.lbReporte,
    valueField: 'idreporte',
    displayField: 'denreporte',
    store: store,
    triggerAction: 'all',
    forceSelection: true,
    typeAhead: true,
    editable: false,
    emptyText: _this.idioma.lbCbVacio,
    mode: 'local',
    width: 350,
    selectOnFocus: true,
    listeners: {
        'select': function(combo, record) {
            if (combo.getValue() == 3 || combo.getValue() == 4) {
                _this.Reporte = false;
                if (!_this.Especialidad) {
                    _this.FechaDesde.reset();
                    _this.FechaHasta.reset();
                    _this.FechaDesde.disable();
                    _this.FechaHasta.disable();
                }
            } else {
                _this.FechaDesde.enable();
                _this.FechaHasta.enable();
            }
        }
    }
});
    
```

Alta cohesión: Este patrón nos dice que la información que almacena una clase debe de ser coherente y debe estar (en la medida de lo posible) relacionada con la clase. Una clase con mucha cohesión es útil porque es fácil darle mantenimiento, entenderla y reutilizarla.



Controlador: sirve como intermediario entre una determinada interfaz y el algoritmo que la implementa, de tal forma que es la que recibe los datos del usuario y la que los envía a las distintas clases.

```

class ReportesController extends BaseController { //Se usa esta clase en Distra

    public function init() {
        parent::init('log.aft.recuperacionespatrimonio');
        $this->view->arrayJs = array(/*$this->integrator->ayuda->devolverJS(),
        $this->pIntegrator->utilesGeneral->getJsReporte() */);
        $this->areaSeleccionada = $this->global->Estructura->idestructura;
    }
}

```

Patrones GoF

Los patrones *GoF*³, describen las formas comunes en que diferentes tipos de objetos pueden ser organizados para trabajar unos con otros. Tratan la relación entre clases, la combinación de clases y la formación de estructuras de mayor complejidad. Permiten crear grupos de objetos para ayudar a realizar tareas complejas (Márquez, 2013).

Patrones de Creación

El objetivo de estos patrones es de abstraer el proceso de instanciación y ocultar los detalles de cómo son creados o inicializados los objetos.

- *Prototype* (Prototipo): permite crear nuevos objetos clonándolos de una instancia ya existente, copiando el prototipo de esta clase, es utilizado por el marco de trabajo.
- *Singleton* (Instancia única): Garantiza la existencia de una única instancia para una clase y la creación de un mecanismo de acceso global a dicha instancia, es utilizado por el marco de trabajo.

Conclusiones del capítulo

Después de analizar las características del módulo se llega a las siguientes conclusiones:

- Generar los artefactos que dispone *Prodesoft* permitió la identificación de los requisitos funcionales y no funcionales del módulo.
- La elaboración del modelo de dominio permitió una mayor comprensión del negocio.
- Con la identificación de la arquitectura, la propuesta de solución cuenta con un alto grado de resistencia ante posibles modificaciones.

³ Banda de los cuatro, compuesta por los siguientes autores: Erich Gamma, Richard Helm, Ralph Johnson, John Vlissides.

- Mediante el diseño de la base de datos se facilita la visión en cuanto a composición física y lógica del módulo.

Capítulo 3: Implementación y Pruebas.

A partir de los resultados obtenidos del diseño, se comienza la implementación y las pruebas del módulo. Se muestra el diagrama de despliegue y de componentes, y para realizar la validación de la implementación se aplican pruebas de caja negra y caja blanca con las diferentes estrategias que define Pressman, las cuales permiten probar las funcionalidades del componente para detectar posibles errores, posibilitando mejorar su calidad.

Diagrama de componentes

Un componente es una parte física de un sistema (módulo, base de datos, programa ejecutable). Se puede decir que un componente es la materialización de una o más clases (Hernández, 2013).

Seguidamente se presenta el diagrama de componentes correspondiente al módulo de generación de reportes del subsistema Patrimonio:

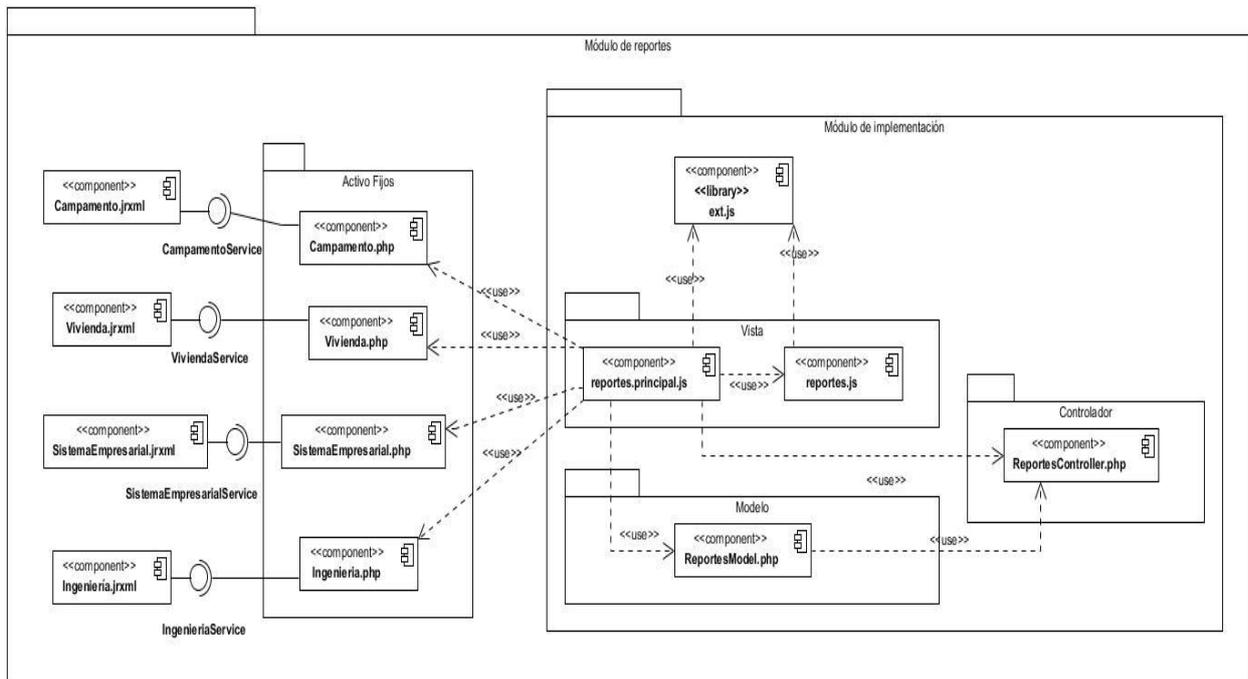


Ilustración 10: Diagrama de componentes

Fuente: (Elaboración propia)

Tabla 4: Estereotipos usados en el diagrama de componentes

Fuente: (Creación de los autores)

Componentes	Descripción
-------------	-------------

reportes.principal.js	Este componente se encarga de construir la interfaz principal donde se muestran los reportes seleccionados por el cliente, haciendo uso de la librería ext.js
reportes.js	Este componente se encarga del idioma de los contenidos de la interfaz.
ReportesController.php	Este componente se encarga de la lógica de todo el proyecto.
ReportesModel.php	Se encarga de la conexión con la fuente de datos y consulta de los datos.
Campamento.php	Contiene todos los reportes de la especialidad campamento en los activos fijos.
Vivienda.php	Contiene todos los reportes de la especialidad vivienda en los activos fijos.
SistemaEmpresarial.php	Contiene todos los reportes de la especialidad sistema empresarial en los activos fijos.
Ingenieria.php	Contiene todos los reportes de la especialidad ingeniería en los activos fijos.
Campamento.jrxml	Contiene todos los diseños de los reportes de la especialidad campamento en el <i>JasperReport Server</i>
Vivienda.jrxml	Contiene todos los diseños de los reportes de la especialidad vivienda en el <i>JasperReport Server</i>
SistemaEmpresarial.jrxml	Contiene todos los diseños de los reportes de la especialidad sistema empresarial en el <i>JasperReport Server</i>

Ingenieria.jrxml	Contiene todos los diseños de los reportes de la especialidad ingeniería en el <i>JasperReport Server</i>
------------------	---

Diagrama de despliegue

Los Diagramas de despliegue muestran las relaciones físicas de los distintos nodos que componen un sistema y el reparto de los componentes sobre los nodos. La vista de despliegue representa la disposición de las instancias de componentes de ejecución en instancias de nodos conectados por enlaces de comunicación. Un nodo es un recurso de ejecución tal como un computador, un dispositivo o memoria (Michel, 2016).

Es un tipo de diagrama UML, que se utiliza para modelar el hardware utilizado en las implementaciones de sistemas y las relaciones entre sus componentes. Los Diagramas de Despliegue muestran las relaciones físicas de los distintos nodos que componen un sistema y el reparto de los componentes sobre dichos nodos, ver Ilustración 11.

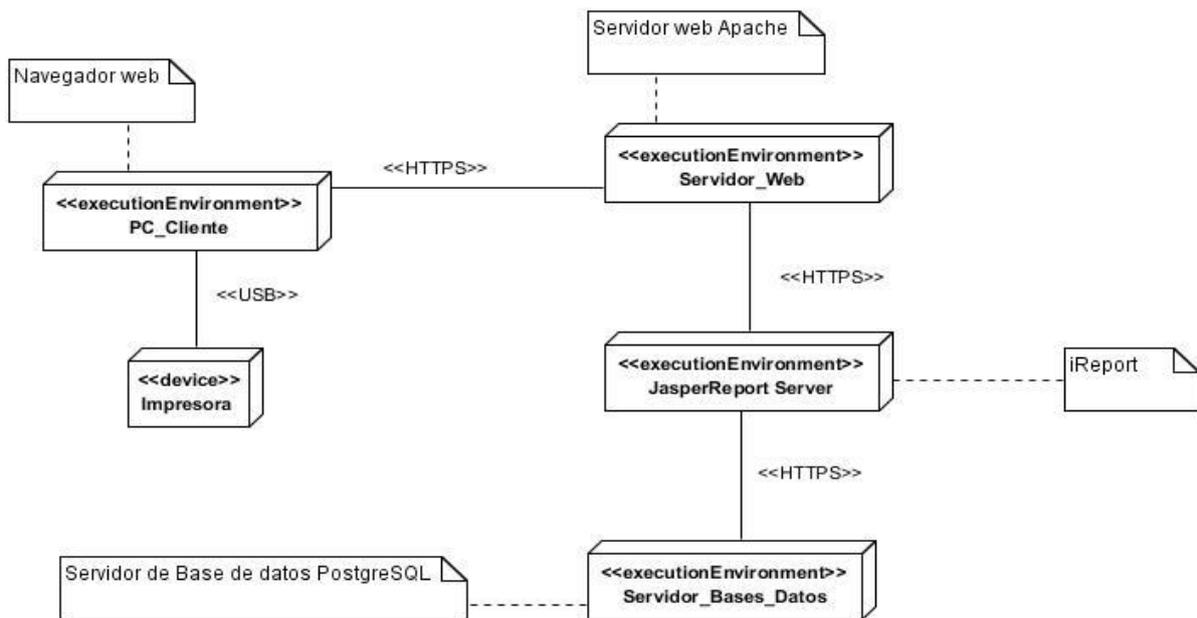


Ilustración 11: Diagrama de despliegue

Fuente: (Elaboración propia)

Tabla 5: Estereotipos utilizados en el diagrama de despliegue

Fuente: (Elaboración propia)

Estereotipos del diagrama de despliegue
--

Estereotipos	Descripción
<i>ExecutionEnvironment</i>	Ofrece un entorno para ejecutar un tipo específico de artefactos ejecutables.
<i>Device</i>	Recurso computacional físico sobre el cual puedes ser desplegados artefactos para su ejecución.
Conexión USB	Permite la comunicación entre la pc cliente y la impresora.
Conexión HTTPS	Protocolo seguro que permite la comunicación de la pc cliente con el servidor web y posibilita la comunicación entre los servidores.

Estándares de codificación y tratamiento de errores

Un estándar de codificación comprende todos los aspectos de la generación de código. Un código fuente debe reflejar un estilo armonioso, como si un único programador hubiera escrito todo el código de una sola vez. El estándar de codificación debería establecer cómo operar con la base de código (MICROSOFT, 2017). Para el desarrollo de la solución se hará uso de los estándares propuestos en los documentos “Estándar de Codificación para PHP” en su versión 2.0, para la implementación de las clases controladoras y “Estándar de Diseño de Interfaces para las aplicaciones de Gestión”, en su versión 1.0, para el diseño de las vistas, definidos por la empresa XETID.

Estándares de codificación

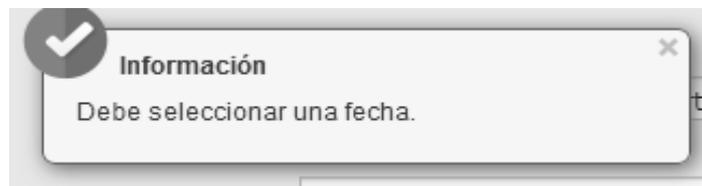
Estándar de codificación para PHP

1. En caso de que los argumentos sobrepasen el tamaño máximo de la línea, se puede dividir su declaración. Ejemplo:

```
/**
 * Bloque de documentación de la clase
 */
class ClaseEjemplo
{
    /**
     * Bloque de documentación de la función
     */
    public function bar($arg1, $arg2, $arg3,
        $arg4, $arg5, $arg6
    ) {
        // código de la función
        // debe estar indentado.
    }
}
```

Estándar de diseño de interfaces para las aplicaciones de gestión

1. Si el usuario da clic en el botón Mostrar y en el formulario hay campos no vacíos que dependen del reporte solicitado, automáticamente se mostrará un mensaje de error que dirá: “Debe seleccionar los campo(s) que faltan”.



2. Los ficheros que conforman la capa de presentación son: Archivos de tipo "*.js" (*javascript*). Estos se ejecutan en la parte del cliente, y pueden contener validaciones.

`report.view.js`

3. Botones para colocar en *topToolbar* (tbar):

Actualizar: Realiza la solicitud a la fuente de datos que determina lo que se muestra en la interfaz en un instante determinado. Se receptiona el pedido y se vuelve a recargar la interfaz con los datos actualizados.

Exportar: Dependiente del contexto. Crea un fichero con una estructura de datos determinada, y se archiva fuera de la aplicación.

Imprimir: Dependiente del contexto. Envía a la impresora una configuración determinada por una pantalla de datos para plasmarla en soporte de papel.

Mostrar: Dependiente del contexto. Muestra en una interfaz el reporte solicitado por el cliente

Tratamiento de errores

Existen mecanismos para evitar cometer errores y facilitar el tratamiento en caso de que se produzcan, garantizándose el correcto funcionamiento de la aplicación, por ejemplo:

- Validar que no se introduzcan valores incorrectos para determinados tipos de datos en la base de datos.
- Se deben introducir números enteros para el tipo de dato NUMERIC.
- El tipo de dato VARCHAR representará cadenas numéricas, con diferentes longitudes, VARCHAR (10) para las cadenas abreviadas, VARCHAR (11) para campos de identidad, VARCHAR (30-100) para campos de denominaciones y VARCHAR (255) para campos de descripciones.
- Creación de índices únicos que garanticen que no existan datos duplicados.

```
ALTER TABLE mod logpatrimonio.dat identificacioninmueblecamp  
ADD CONSTRAINT dat_identificacioninmueblecamp_pkey  
PRIMARY KEY (ididentificacioninmueble);
```

Interfaces del módulo

Una interfaz de usuario constituye el medio mediante el cual los usuarios pueden establecer comunicación con un sistema e interactuar con la información que brinda. Para lograr una buena aceptación del módulo se encaminó su diseño en el que se logró una interfaz intuitiva, amigable y fácil de asimilar por los usuarios, permitiéndoles navegar e interactuar con éste cómodamente. Además, se siguieron los estándares de diseño de interfaz de usuario y de codificación definidos en la empresa y se garantizó el soporte de los lenguajes definidos.

Para acceder al módulo desde la plataforma el usuario debe seleccionar del Menú Inicio la opción Activos Fijos/ Patrimonio/ Reportes (Ilustración 12).

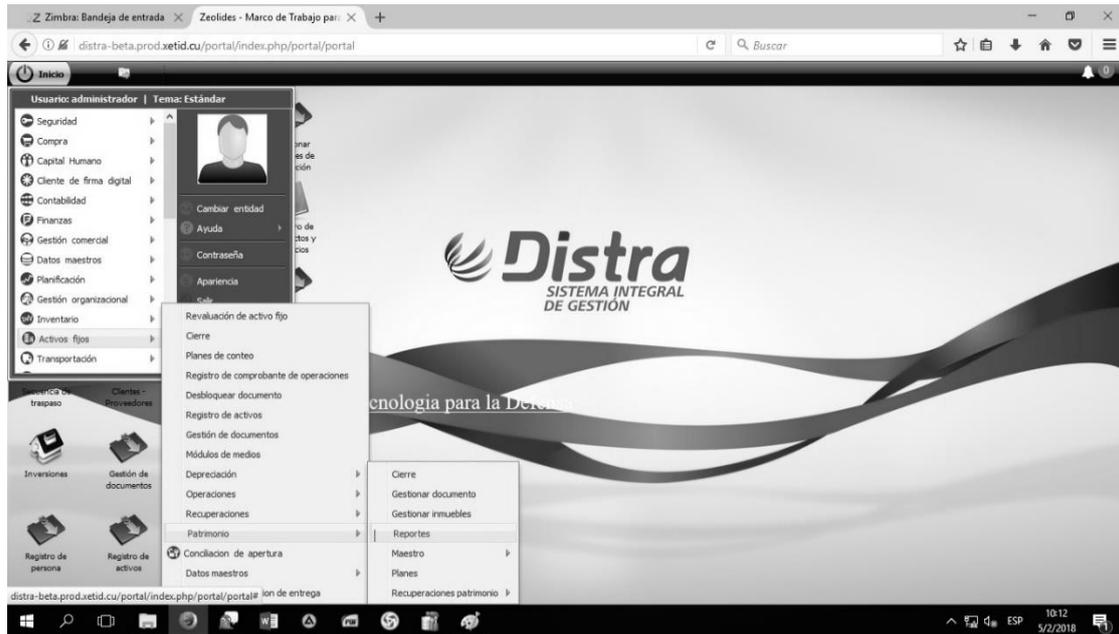


Ilustración 12: Menú Inicio/ActivosFijos/Patrimonio/Reportes.

Fuente: (Tomado del portal Distra <http://distra-beta.prod.xetid.cu/portal/index.php/portal/portal>)

Una vez dentro del módulo de Reportes el resultado final de la interfaz de usuario funcional es la siguiente:

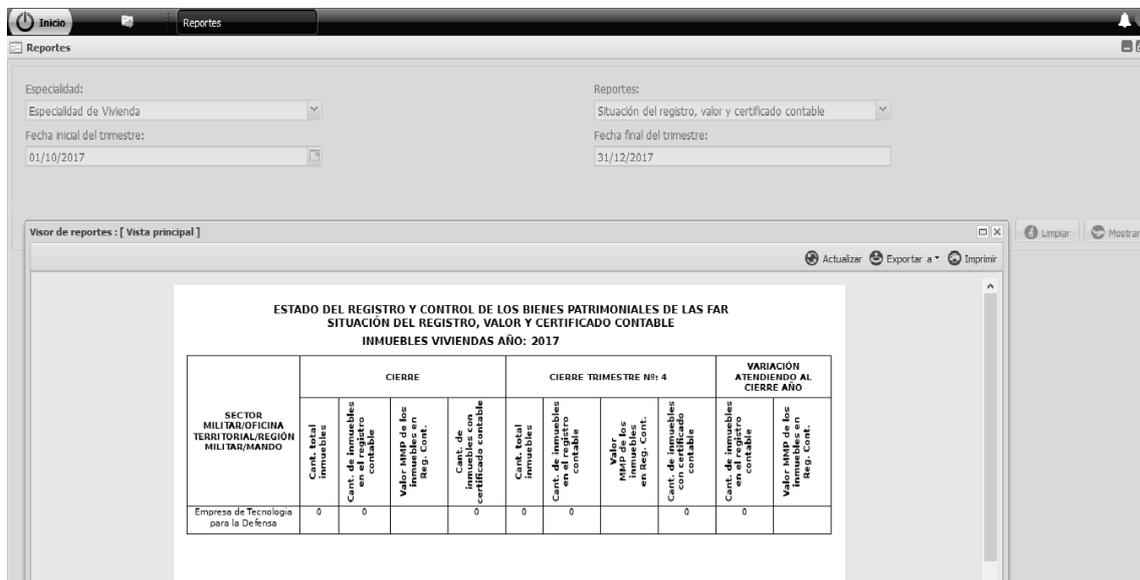


Ilustración 13: PIU funcional del módulo

Fuente: (Elaboración propia)

Pruebas

El proceso de prueba se centra en los conocimientos lógicos internos del software, donde se aseguran que todas las sentencias se han comprobado, y en los procesos externos funcionales, es decir, la realización de las pruebas para la detección de errores. Estas, además, son utilizadas para identificar posibles fallos de implementación, calidad o usabilidad de un programa (Pressman, 2010). En este epígrafe se muestran los resultados de la estrategia de prueba diseñada para la propuesta de solución, en función de garantizar y validar su calidad.

Pruebas Funcionales

Las pruebas funcionales se llevan a cabo sobre la interfaz del software sin prestar atención al código, los casos de prueba son creados con el objetivo de demostrar que la entrada es aceptada de forma adecuada y que se produce una salida correcta. El diseño de estas pruebas se realiza con la intención de detectar funciones incorrectas o ausentes, errores en accesos a bases de datos externas, errores de interfaz, errores de rendimiento, y errores de inicialización y de terminación. Dentro de la prueba se incluyen la técnica de partición de equivalencia que será la empleada en la validación. Esta técnica divide el campo de entrada de un programa en clases de datos de los que se pueden derivar casos de prueba (Pressman, 2010).

Diseño de los casos de prueba

El diseño de casos de pruebas tiene como objetivo tener la mayor probabilidad de encontrar el mayor número de errores con la mínima cantidad de esfuerzo y tiempo posible. Por cada requisito funcional que requiere de la introducción de valores, se diseñó un caso de prueba que determina si el requisito es parcial o completamente satisfactorio.

Se realizaron los diseños de casos de prueba para validar la propuesta de solución implementada para comprobar que la herramienta genere correctamente un reporte determinado. A continuación, en la Tabla 6, se muestra las variables empleadas en el caso de prueba para el escenario “Generar reporte de las distintas especialidad”

Tabla 6: Variables empleadas en los escenarios de prueba de los requisitos funcionales generar reporte

Fuente: (Elaboración propia)

Nro.	Nombre del campo.	Clasificación.	Nulo.	Descripción.
1	Especialidad. (obligatorio)	Campo de selección.	No	Contiene las diferentes especialidades del módulo.

2	Tipo de reporte. (obligatorio)	Campo de selección.	No	Contiene los tipos de reportes.
3	Fecha inicial del trimestre (obligatorio).	Campo de selección.	No	Contiene el 1er día del trimestre.
4	Fecha final del trimestre.	Campo autocompletado	No	Contiene el último día del trimestre que se autocompleta de acuerdo al 1er día seleccionado.

Tabla 7: Escenarios de los casos de pruebas RF 1: Generar reporte especialidad de Campamento

Fuente: (Elaboración propia)

Escenario.	Descripción.	V1	V2	V3	V4	Resultados.	Flujo central.
EC. 1.1. generar reporte especialidad de Campamento con los campos correctamente.	Se llenan todos los campos que muestra el sistema correctamente.	Especialidad de Campamento.	Tipo de reporte a obtener.	Rango de fecha inicial deseado.	Rango de fecha final deseado.	Muestra el reporte deseado por el especialista.	El especialista rellena los campos obligatorios de forma correcta y da clic en el botón Mostrar.
EC.1.2. generar reporte especialidad de Campamento solo con algunos campos.	Se llenan algunos campos que muestra el sistema correctamente.	Especialidad de Campamento.	Tipo de reporte a obtener.	Vacío	Vacío	Muestra el mensaje de error "Existen campos vacíos"	El especialista rellena algunos campos de forma correcta y da clic en el botón Mostrar.

Tabla 8: Escenarios de los casos de pruebas RF 2: Generar reporte especialidad de Ingeniería

Fuente: (Elaboración propia)

Escenario.	Descripción.	V1	V2	V3	V4	Resultados.	Flujo central.
------------	--------------	----	----	----	----	-------------	----------------

EC. 2.1. generar reporte especialidad de Ingeniería con los campos correctamente.	Se llenan todos los campos que muestra el sistema correctamente.	Especialidad de Ingeniería.	Tipo de reporte a obtener.	Rango de fecha inicial deseado.	Rango de fecha final deseado.	Muestra el reporte deseado por el especialista.	El especialista rellena los campos obligatorios de forma correcta y da clic en el botón Mostrar.
EC.2.2. generar reporte especialidad de Ingeniería solo con algunos campos.	Se llenan algunos campos que muestra el sistema correctamente.	Especialidad de Ingeniería.	Tipo de reporte a obtener.	Vacío	Vacío	Muestra el mensaje de error "Existen campos vacíos"	El especialista rellena algunos campos de forma correcta y da clic en el botón Mostrar.

Tabla 9: Escenarios de los casos de pruebas RF 3: Generar reporte especialidad de Sistema Empresarial

Fuente: (Elaboración propia)

Escenario.	Descripción.	V1	V2	V3	V4	Resultados.	Flujo central.
EC. 3.1. generar reporte especialidad de Sistema Empresarial con los	Se llenan todos los campos que muestra el sistema correctamente.	Especialidad de Sistema Empresarial.	Tipo de reporte a obtener.	Rango de fecha inicial deseado.	Rango de fecha final deseado.	Muestra el reporte deseado por el especialista.	El especialista rellena los campos obligatorios de forma correcta y da clic en el botón Mostrar.

campos correctamente.							
EC.3.2. generar reporte especialidad de Sistema Empresarial solo con algunos campos.	Se llenan algunos campos que muestra el sistema correctamente.	Especialidad de Sistema Empresarial.	Tipo de reporte a obtener.	Vacío	Vacío	Muestra el mensaje de error "Existen campos vacíos"	El especialista rellena algunos campos de forma correcta y da clic en el botón Mostrar.

Tabla 10: Escenarios de los casos de pruebas RF 4: Generar reporte especialidad de Vivienda

Fuente: (Elaboración propia)

Escenario.	Descripción.	V1	V2	V3	V4	Resultados.	Flujo central.
EC. 4.1. generar reporte especialidad de Vivienda con los campos correctamente.	Se llenan todos los campos que muestra el sistema correctamente.	Especialidad de Vivienda.	Tipo de reporte a obtener.	Rango de fecha inicial deseado.	Rango de fecha final deseado.	Muestra el reporte deseado por el especialista.	El especialista rellena los campos obligatorios de forma correcta y da clic en el botón Mostrar.

<p>EC.4.2. generar reporte especialidad de Vivienda solo con algunos campos.</p>	<p>Se llenan algunos campos que muestra el sistema correctamente.</p>	<p>Especialidad de Vivienda.</p>	<p>Tipo de reporte a obtener.</p>	<p>Vacío</p>	<p>Vacío</p>	<p>Muestra el mensaje de error "Existen campos vacíos"</p>	<p>El especialista rellena algunos campos de forma correcta y da clic en el botón Mostrar.</p>
--	---	----------------------------------	-----------------------------------	--------------	--------------	--	--

Las pruebas funcionales aplicadas a la propuesta de solución se basaron en el diseño de casos de pruebas, donde se recogen los escenarios correspondientes a cada requisito funcional. Se realizaron en tres (3) iteraciones. En la primera iteración se detectaron 10 no conformidades, de ellas 6 de validación, 1 de redacción y 3 de errores ortográficos, las mismas fueron analizadas y corregidas satisfactoriamente. En la segunda iteración se encontraron un total de 4 no conformidades, de ellas 3 de validación y 1 de errores ortográficos, las cuales fueron mitigadas con éxito. Se realizó una tercera iteración donde no se detectaron no conformidades como resultado final, lo que demuestra que el software funciona correctamente. A continuación, en la Tabla 11 se muestran los resultados obtenidos en cada iteración de prueba al Módulo de reportes del subsistema Patrimonio.

Tabla 11: Relación de no conformidades

Fuente: (Elaboración propia)

	Iteración 1	Iteración 2	Iteración 3
Validación	6	3	0
Redacción	1	0	0
Errores ortográficos	3	1	0
Total de no conformidades	10	4	0

Pruebas de Carga y Estrés

Las pruebas de carga se utilizan para validar y valorar la aceptabilidad de los límites operacionales de un sistema bajo carga de trabajo variable, mientras el sistema bajo prueba permanece constante. Además, se simula la carga de trabajo promedio y con picos que ocurre dentro de tolerancias operacionales normales y las pruebas de estrés están enfocadas a evaluar cómo el sistema responde bajo condiciones anormales (Berges Pedrianes, y otros, 2013).

Para la realización de las pruebas de carga y estrés se utilizó la herramienta *Apache JMeter* 2.12 y un ambiente de prueba con las siguientes características:

- ❖ Sistema Operativo: *Windows* 10
- ❖ Microprocesador: Intel(R) Core(TM) i7 CPU 4790 @3.6 GHz

- ❖ Memoria RAM: 8 GIB
- ❖ Disco Duro: 1TB

La validación realizada consistió en definir dos pruebas de 25 y 50 usuarios. Se definió una lista de enlaces a los que se simuló el acceso aleatorio, luego se recolectaron los datos necesarios para su interpretación.

El análisis

- ❖ Se analizaron los resultados a través de un intervalo de confianza 10 con un nivel de confianza al 95%.
- ❖ Para el análisis, dado que la muestra se considera grande, no se requiere hacer la suposición de que la muestra tiene una distribución normal ya que por el Teorema Central del Límite (TCL), para n grande implica que X tiene una distribución aproximadamente Normal sin importar la naturaleza de la distribución poblacional (Devore, 2010).

Para la primera prueba se configuró el entorno de prueba para 25 especialistas, cada “1 segundo”. Los valores obtenidos por la herramienta *Apache JMeter* se muestran a continuación en la Ilustración 14:

Etiqueta	# Muestras	Media	Mediana	Linea de 90%	Mín	Máx	% Error	Rendimiento	Kb/sec
606 /predefe...	25	6092	6091	9798	2215	9846	0,00%	2,3/sec	10,8
641 /predefe...	25	5080	4866	6274	4150	6492	0,00%	1,7/sec	7,3
676 /predefe...	25	4008	4106	4614	2694	5423	0,00%	1,8/sec	7,4
779 /predefe...	25	1455	1543	1923	770	2035	100,00%	2,2/sec	21,3
813 /predefe...	25	1331	1163	1957	676	2062	100,00%	2,4/sec	23,7
914 /predefe...	25	1295	1407	1988	363	2214	100,00%	2,9/sec	28,9
Total	150	3210	2214	6274	363	9846	50,00%	6,6/sec	47,1

Ilustración 14: Tiempo promedio para 25 especialistas

Fuente: (Resultado de JMeter)

Como se observa, el tiempo promedio para acceder a una página es 3,210 segundos, realizándose un total de 150 peticiones al servidor. El tiempo total utilizado para los 25 usuarios se puede calcular con la siguiente fórmula (Diaz, y otros, 2010):

$$\text{Tiempo total} = \#Muestras * Media = 150 * 3210 = 481500 \text{ milisegundos}$$

El tiempo total requerido por cada usuario, se puede calcular de la siguiente manera (Diaz, y otros, 2010):

$$((\text{Tiempo total}/1000)/60)/\text{Cantidad de especialistas} = ((481500/1000)/60)/25 = 0.321 \text{ minutos}$$

Análisis realizado

Se evaluaron los resultados obtenidos a través de un intervalo de confianza del 95% para muestras grandes. La fórmula del mismo es la siguiente (Diaz, y otros, 2010):

$$TP - z_{0.95} * S / \sqrt{n}; TP + z_{0.95} * S / \sqrt{n} \text{ donde:}$$

- Tiempo promedio (TP) de respuesta es: 10254
- Estimador de Desvío (S) es: $\sqrt{\sum_{i=1}^n (X_i - x)^2 / n - 1} = 3042$
- Tamaño de la muestra (n) es de: 150
- $z_{0.95} = 1.96$

El intervalo resultante es el siguiente:

$$[3210 - 1,96 * 3042 / \sqrt{150}; 3210 + 1,96 * 3042 / \sqrt{150}]$$

$$[3210 - 487; 3210 + 487]$$

$$[2.723; 3.697]$$

Por lo tanto, para una cantidad de 25 usuarios simultáneos realizando 150 solicitudes se espera un tiempo de respuesta promedio entre 2,723 y 3,697 segundos.

Para la segunda se configuró el entorno de prueba para 50 usuarios, cada “1 segundo”. Los valores obtenidos por la herramienta de prueba *Apache JMeter* se muestran a continuación en la Ilustración 15:

Etiqueta	# Muestras	Media	Mediana	Linea de 90%	Mín	Máx	% Error	Rendimiento	Kb/sec
606 /predefe...	50	13785	17324	19171	1551	19494	0,00%	2,5/sec	11,8
641 /predefe...	50	9630	10282	11296	1694	11618	0,00%	1,7/sec	7,4
676 /predefe...	50	7920	7873	8596	4020	9909	0,00%	1,4/sec	5,9
779 /predefe...	50	3289	3342	3855	2500	4122	100,00%	1,4/sec	13,9
813 /predefe...	50	3059	3064	3466	2234	4156	100,00%	1,5/sec	14,4
914 /predefe...	50	2805	2694	3650	1175	4016	100,00%	1,6/sec	16,1
Total	300	6748	3824	15320	1175	19494	50,00%	6,4/sec	46,1

Ilustración 15: Tiempo promedio para 50 especialistas

Fuente: (Resultado de JMeter)

El tiempo promedio para acceder a una página es 6,748 segundos, realizándose un total de 300 peticiones al servidor. El tiempo total utilizado para los 50 especialistas y el requerido para cada uno se puede calcular de la misma manera que en la primera prueba:

$$\text{Tiempo total} = \#Muestras * Media = 300 * 6748 = 2024400 \text{ milisegundos}$$

$((\text{Tiempo total}/1000) /60) /\text{Cantidad de especialistas} = ((2024400/1000) /60) /50 = 0.6748$
 minutos

Análisis realizado

La segunda prueba fue realizada de manera similar que, en la primera, usando las mismas fórmulas.

Con un intervalo de confianza al 95% para muestras grandes, el intervalo resultante es el siguiente:

$$[6748 - 1,96 * 6377/\sqrt{300}; 6748 + 1,96 * 6377/\sqrt{300}]$$

$$[6748 - 722; 6748 + 722]$$

$$[6026; 7470]$$

Esto significa que puede esperarse un tiempo de respuesta promedio entre 0.6026 y 0.7470 segundos para una cantidad de 50 usuarios simultáneos realizando 300 solicitudes.

Validación de la idea a defender.

Para validar la idea a defender, se le entregó la aplicación a un grupo de especialistas del centro, para que emitieran su criterio a partir de sus consideraciones respecto a las ventajas que ofrece el módulo y las necesidades que resuelve. Como instrumento se utilizó una encuesta (ver [Anexo 4](#)) a través de la cual se puede validar la idea a defender. La encuesta fue realizada a cinco (5) especialistas, que poseen un promedio entre 5 y 6 años de experiencia trabajando en dicho centro de desarrollo.

Tabla 12: Relación de los especialistas del centro

Fuente: (Creación de los autores)

Nro.	Nombre y Apellidos	Centro	Años de experiencia
1	Jenny Martínez Martínez	GE-Logística	5
2	Jorge Luis Saavedra Rubio	GE-Logística	6
3	Osmel Cutiño Felipe	GE-Logística	5
4	Yadrián Águila Ferrán	GE-Logística	5

5	Yoendy Martínez Contrera	GE-Logística	6
---	--------------------------	--------------	---

La Tabla 13 muestra el resumen del resultado de los juicios emitidos por los encuestados:

Tabla 13: Resultado de los juicios emitidos por los encuestados

Fuente: (Elaboración propia)

Parámetros evaluados	Nivel de valoración									
	MA		A		PA		NA		Total	
	N	%	N	%	N	%	N	%	N	%
1	5	100	0	0	0	0	0	0	5	100
2	2	40	3	60	0	0	0	0	5	100
3	3	60	2	40	0	0	0	0	5	100
4	4	80	1	20	0	0	0	0	5	100
5	5	100	0	0	0	0	0	0	5	100

Para el procesamiento y análisis de la información obtenida se analizaron las respuestas de cada uno de los parámetros que aparecen en la encuesta. De esta forma se presentan los resultados teniendo en cuenta que los niveles empleados para la valoración fueron: MA: Muy adecuado, A: Adecuado, PA: Poco adecuado y NA: No adecuado. En la tabla, N se refiere a la cantidad de encuestados que emitieron una valoración determinada y % al por ciento que representa con respecto al total de encuestados.

Los resultados mostrados anteriormente, son representados de forma gráfica a continuación:

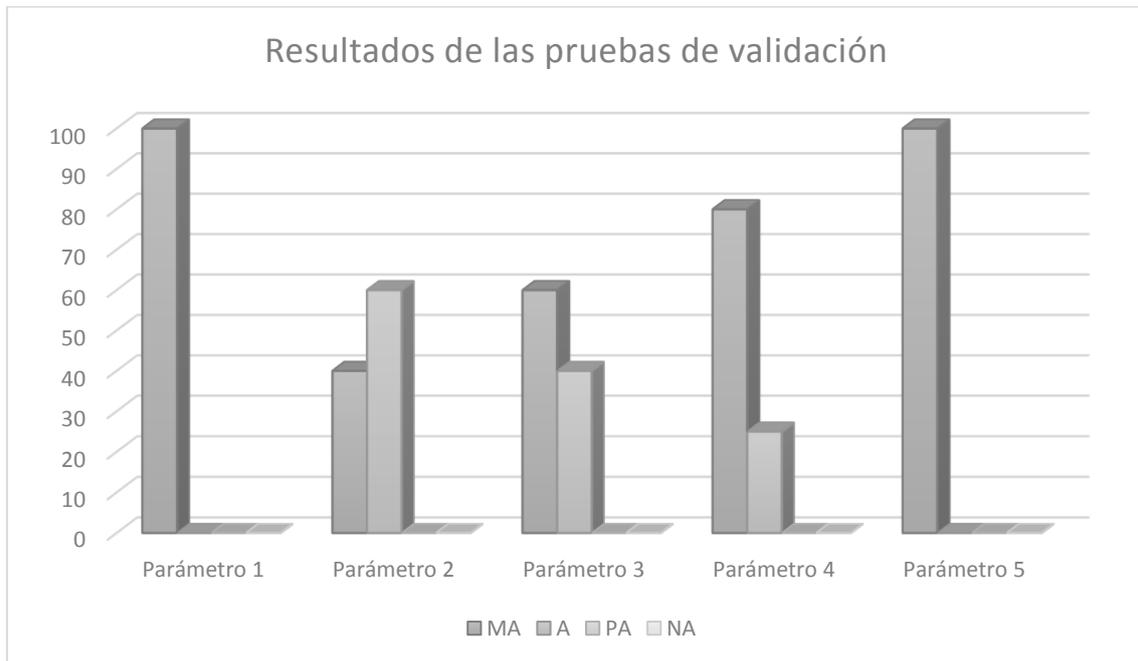


Ilustración 16: Resultados de la validación

Fuente: (Elaboración propia)

Como se observa en la Ilustración 16, al analizar los resultados, se pudo comprobar que el 100% de los encuestados consideran muy adecuado el tratamiento de los aspectos teóricos de la gestión de la información de los reportes en el sistema implementado y el nivel de usabilidad del mismo. El nivel de apoyo que brinda al especialista del centro de Logística para la gestión de los reportes del subsistema Patrimonio y su mantenimiento y el rendimiento de los datos de la base de datos fue valorado de muy adecuado por el 50% de los encuestados y el otro 50% considera que son adecuados. Por otra parte, el 75% de los encuestados considera que es muy adecuada la presentación de una interfaz agradable e intuitiva para el usuario.

A partir de los criterios y otras consideraciones de los especialistas encuestados, se evidencia que la solución implementada tiene un nivel satisfactorio de aceptación entre ellos. Todos los aspectos evaluados, que están en concordancia con el objetivo general de la investigación, fueron valorados entre los niveles de adecuado y muy adecuado. Esto demuestra el correcto cumplimiento del mismo desde el punto de vista de los expertos. Además, se obtuvo un conjunto de recomendaciones y valoraciones que aportan mejoras a la propuesta de solución, las cuales se tienen en cuenta para el resultado final de la investigación, así como futuras profundizaciones sobre la misma.

Conclusiones del capítulo.

En este capítulo se abordaron los elementos de la implementación del sistema de control y gestión de reportes para el subsistema Patrimonio, así como las pruebas realizadas al mismo y los resultados obtenidos; lo que permite concluir:

- La confección del diagrama de componentes permitió una mejor comprensión de la estructura de los componentes de la propuesta de solución.
- El empleo de los estándares de codificación permitió garantizar la alta calidad, minimización de errores y mayor limpieza en el código fuente. Lográndose así, que pueda ser mantenido fácilmente y reutilizado por otros desarrolladores que lo necesiten.
- La implementación de la propuesta de solución facilitó la obtención de una aplicación funcional lista para su uso.
- La validación de la propuesta de solución, mediante una estrategia de pruebas de software, evidenció que la misma cumple con los requisitos definidos por el cliente.

Beneficios del módulo de reportes para el subsistema Patrimonio de las FAR.

El desarrollo del módulo de reportes beneficia y mejora el trabajo que se desarrolla en el subsistema Patrimonio de las Fuerza Armadas Revolucionarias:

- Posibilita que todos los especialistas tengan acceso a la misma información, fluyendo y llegando oportunamente a quienes la necesitan de manera automatizada.
- Incrementa el desempeño y la productividad de cada persona que necesita alguna información determinada y que el sistema le permite obtenerla rápidamente.
- El subsistema de Patrimonio adquiere una capacidad de responder más rápido a las necesidades de información de las diferentes cadenas de mando de las FAR.
- Estos reportes permiten realizar análisis diarios y organizar con mayor exactitud toda la información estadística de la empresa.
- Permite que la fuerza de trabajo pueda detectar cambios y oportunidades y por supuesto, actuar o responder rápidamente e incluso anticipadamente a ellos.
- Mejora la administración de los recursos debido a que la simpleza de los reportes se convierte en una ayuda estratégica para la toma de decisiones.
- La posibilidad de un monitoreo en tiempo real, permite la corrección inmediata de posibles errores.
- Aumenta la calidad del servicio ya que cuenta con información exacta y al momento.
- Además de los beneficios intrínsecos de disponer de información válida para tomar decisiones, la gran ventaja de este módulo de reporte es la automatización de la generación de informes.

Conclusiones generales.

El presente Trabajo de Diploma concluye con el desarrollo de un Módulo de reportes para el subsistema Patrimonio de las FAR, el cual sirve de apoyo a la toma de decisiones. Esto se debe a que el sistema fue desarrollado como tecnología para el apoyo al trabajo de los especialistas del centro de Logística facilitando el proceso de gestión de los reportes del patrimonio de las FAR.

Se destaca, además que:

- ❖ El estudio realizado de los sistemas homólogos, evidenció la necesidad de implementar un módulo de reportes para el análisis de la información almacenada en el sistema Patrimonio y facilitar el apoyo a la toma de decisiones, definiéndose una propuesta de solución de acuerdo a las necesidades existentes.
- ❖ La selección de herramientas, lenguajes y tecnologías permitió la implementación del módulo de reportes para el subsistema Patrimonio de las Fuerzas Armadas Revolucionarias para el análisis de la información almacenada en el subsistema Patrimonio.
- ❖ La utilización de la estrategia de pruebas garantizó la identificación temprana de las deficiencias en el módulo que se desarrolló; corrigiéndose las mismas y logrando un producto más seguro y funcional.
- ❖ El desarrollo del módulo de reportes para el subsistema Patrimonio de las Fuerzas Armadas Revolucionarias contribuyó a la mejora del proceso de apoyo a la toma de decisiones.

Recomendaciones.

Con el cumplimiento del marcado objetivo en la presente investigación y luego de alcanzado el resultado esperado, se propone la siguiente recomendación:

- ❖ Crear un interfaz para configurar los reportes y/o modificar la cantidad de columnas a devolver en el mismo.

Bibliografía

- Abernethy, Michel. 2011.** ¿Simplemente qué es Node.js? Un servidor listo para codificar. [En línea] 14 de junio de 2011. [Citado el: 15 de marzo de 2018.] <https://www.ibm.com/developerworks/ssa/opensource/library/os-nodejs/>.
- Alan, Le. Softonic Bihan. 2015.** [En línea] 2015. [Citado el: 10 de enero de 2018.] <http://mozilla-firefox.softonic.com/>.
- ALEGSA.com.ar. 2016.** Diccionario de informática y tecnología. *Definición de estructura de control*. [En línea] 19 de julio de 2016. [Citado el: 21 de marzo de 2018.] http://www.alegsa.com.ar/Dic/estructura_de_control.php.
- Álvarez. 2017.** Sistemas gestores de bases de datos. Introducción a este concepto y características especiales. [En línea] 2017. [Citado el: 18 de enero de 2018.] <http://www.desarrolloweb.com/articulos/sistemas-gestores-bases-datos.html>.
- Armando, Dargel Duany. 2010.** Sistemas de información. [En línea] 24 de febrero de 2010. [Citado el: 26 de octubre de 2017.] <https://www.econlink.com.ar/sistemas-informacion/definicion/>.
- Baeza, and Rivero. 1999.** Modern Information Retrieval. [En línea] 1999. [Citado el: 27 de octubre de 2017.] <http://people.ischool.berkeley.edu/~hearst/irbook/print/chap10.pdf>.
- Berges Pedrianes, Zaidee y González Castro, Aliané Amelia. 2013.** *Desarrollo de un portal web para la Casa Editorial Abril*. La Habana : s.n., 2013.
- Briones, José. 2010.** Qué es un reporte y cómo se hace. [En línea] 2010. [Citado el: 21 de octubre de 2017.] <http://pepishighs.blogspot.com/2009/09/que-es-un-reporte-y-como-se-hace.html>.
- BYSPEL. 2017.** Que es JAVA, Características, JDK, JRE, Bytecode. [En línea] 29 de septiembre de 2017. [Citado el: 15 de marzo de 2018.] <https://byspel.com/que-es-java-caracteristicas-jdk-jre-bytecode/>.
- Castro, L. 2014.** *Guía de gestión del riesgo tecnológico para el tratamiento de la seguridad durante el proceso de desarrollo de software*. 2014.
- Clements, Mary Saw and Paul.** *A fielguide to Boxology: Preliminary classification of architectural styles for software system*. . Proceeding of the 21st International Computer Software and Applications. Conference.

Copyright IBM Corporation. 2010. IBM. *Generación de informes para empresas medianas.* [En línea] 2010. [Citado el: 27 de octubre de 2017.]

<http://ibm.com/cognos/generacion-de-informes-para-empresas-medianas.pdf>..

Corporation, Adobe Systems Incorporated: MITRE. 2006. *Reference Model for Service Oriented Architecture 1.0.* Laboratories of America. Fujitsu : OASIS Standard, 2006.

Devore, Jay. 2010. *Probabilidades y Estadísticas para Ingeniería y Ciencias.* Sexta. 2010.

Díaz. 2009. *Las metodologías ágiles como garantía de calidad del software.* 2009. págs. 40-43. Vol. 3.

Díaz, F. Javier y otros. 2010. *Usando JMeter para pruebas de rendimiento.* La Plata : s.n., 2010.

Doctrine. 2006-2016. Doctrine Team. [En línea] 2006-2016. [Citado el: 7 de enero de 2018.] <http://doctrine-orm.readthedocs.io/en/latest/tutorials/getting-started.html>..

fergarcia. 2013. Entorno de Desarrollo Integrado (IDE). [En línea] 25 de enero de 2013. [Citado el: 3 de abril de 2018.]

<https://fergarcia.wordpress.com/2013/01/25/entorno-de-desarrollo-integrado-ide/>.

Figuroa. 2016. *Introducción a patrones.* 2016.

García. 2010. PHP5 A través de ejemplos. [En línea] 2010. [Citado el: 21 de enero de 2018.] <http://somosprogramacion.blogspot.com/2015/02/php-5-traves-deejemplos-abraham-gutierrez-y-gines-brav.html>..

Garzón, Jennifer Sánchez y Machado Hernández, José Carlos. 2016.

Componentes para el diseño y visualización de reportes en la plataforma DAMIX.

Universidad de las Ciencias Informáticas. La Habana : s.n., 2016. Tesis de pregrado.

GeReport: Sistema de Gestión de Reportes Dinámicos. **Hernández, Cinthya**

Rodríguez. 2014. La Habana : Revista Cubana de Ciencias Informáticas, 2014, Vol. 8. ISSN 2227-1899.

GNU. 2015. GNUstep en español. *¿Qué es un Framework?* [En línea] 2015. [Citado el: 14 de noviembre de 2017.] <https://qnustep.wordpress.com/qnustep-a-fondo/%C2%BFque-es-un-framework-%C2%BFcomo-se-utiliza>.

- Hernández, Alejandro. 2013.** *Los sistemas de información: Evolución y Desarrollo.* Departamento de Economía y Dirección de Empresas, Universidad de Zaragoza. Zaragoza : s.n., 2013.
- Hernández, L. 2013.** Modelo de implementación. [En línea] 2013. [Citado el: 22 de marzo de 2018.] <http://ithleovi.blogspot.com/2013/06/unidad-5-modelo-deimplementacion->.
- Herrera, Cristhian.** AdiptosAlTrabajo.com. *Informes en JAVA con iReport.* [En línea] [Citado el: 12 de marzo de 2018.] <https://www.adictosaltrabajo.com/tutoriales/ireport/>.
- INFODEC.** Generación de Reportes Dinámicos. [En línea] [Citado el: 16 de febrero de 2018.] [http://www.infodeclat.com/generacion-reportes-dinamicos/..](http://www.infodeclat.com/generacion-reportes-dinamicos/)
- Jasperforge.org. 2008.** [En línea] 2008. [Citado el: 11 de febrero de 2018.] http://jasperforge.org/plugins/project/project_home.php?group_id=102.
- Laboratorio de Investigaciones en Gestión de Proyectos, Universidad de las Ciencias Informáticas. 2015.** GESPRO: Ecosistema de software para la Dirección Integrada de Proyectos. [En línea] 2015. [Citado el: 18 de mayo de 2018.] <https://gespro.cidi.prod.uci.cu/gespro/about>.
- Larman. 1999.** *UML y Patrones: Introducción al análisis y diseño orientado a objetos.* 1999. ISBN.
- Lee, S-K and Yu, J-H. 2012.** *Success model of project management information system in construction.* Department of Construction and Engineering, Kwangwoon University Automation in Construction. 2012. págs. 82-93. Vol. 25. ISBN 0926-580.
- Legna, Pablo. 2007.** *Reportes de sostenibilidad. Parte 1: concepto, beneficio y contenido.* Universidad Abierta Interamericana. Argentina : s.n., 2007.
- López, Dr. Mario Rossainz. 2016.** Cursos de Periodo de Primavera. [En línea] 2016. [Citado el: 5 de abril de 2018.] http://rossainz.cs.buap.mx/IngSw_2016/Proyecto/5_PruebasDeSw.pdf..
- Marino. 2008.** Programación en PHP5. Nivel básico. [En línea] 2008. [Citado el: 21 de enero de 2018.] http://administraciondesistemas.pbworks.com/f/Manual_PHP5_Basico.pdf..
- Márquez, Javier. 2013.** Patrones GoF. [En línea] enero de 2013. [Citado el: 5 de abril de 2018.] [https://infow.wordpress.com/category/patrones-de-disenogof/page/2/..](https://infow.wordpress.com/category/patrones-de-disenogof/page/2/)

Márquez, Santiago Solís. 2007. La web Semántica. [En línea] 2007. [Citado el: 5 de abril de 2018.]

Metodología Gestión de Requerimientos. 2010. Técnicas para Identificar Requisitos Funcionales y No Funcionales. [En línea] 2010. [Citado el: 9 de mayo de 2018.] <https://sites.google.com/site/metodologiareq/capitulo-ii/tecnicas-para-identificar-requisitos-funcionales-y-no-funcionales>.

Michel, , Limachi, and Nancy Susana. 2016. *Trabajo de investigación y exposición: Diagrama de despliegue.* 2016.

Microsoft. 2016. Microsoft. Developer Network. [En línea] 2016. [Citado el: 5 de abril de 2018.] <https://msdn.microsoft.com/es-es/library/dd409390.aspx#..>

MICROSOFT. 2017. Revisiones de código y estándares de codificación. [En línea] 2017. [Citado el: 22 de marzo de 2018.] [https://msdn.microsoft.com/es-es/library/aa291591\(v=vs.71\).aspx](https://msdn.microsoft.com/es-es/library/aa291591(v=vs.71).aspx).

Mis respuestas.com. 2015. Mis respuestas.com. [En línea] 2015. [Citado el: 10 de enero de 2018.] <http://www.misrespuestas.com/que-es-un-servidor-web.html..>

Montilva, Jonás A. 1986. *Desarrollo de Sistemas de Información: Administración, Metodología y Técnica.* Universidad de Los Andes. Venezuela : Consejo de Publicaciones, 1986. págs. 1-16.

Novatos en la Programación. 2007. Novatos en la Programación. [En línea] 2007. [Citado el: 12 de enero de 2018.] <http://escandell.cubava.cu/java-ee/>.

OMG. 1997-2016. Unified Modeling Language. [En línea] 1997-2016. [Citado el: 7 de enero de 2018.] <http://www.uml.org/>.

Open Text. 2015. Open TextAnalytics. [En línea] 2015. [Citado el: 21 de diciembre de 2017.] <http://www.actuate.com/download/brochures-datasheets/actuate-birt-datasheet.pdf...>

Pérez-Carballo, and Strzalkowski. 2000. *Natural language information retrieval: progress report information Processing and Management.* 2000. págs. 155-178.

Pérez-Porto, and Merino. 2013. Definición de reportes. [En línea] 2013. [Citado el: 27 de octubre de 2017.] [https://definicion.de/reporte/..](https://definicion.de/reporte/)

PostgreSQL tools. pgAdmin . [En línea] [Citado el: 15 de enero de 2018.] [http://www.pgadmin.org/..](http://www.pgadmin.org/)

PostgreSQL-Es. 2013. Group. The PostgreSQL Global Development. [En línea] 2013. [Citado el: 10 de enero de 2018.] <http://www.postgresql.org.es/documentacion>.

Pressman, Roger S. 2010. *Ingeniería de Software: Un enfoque práctico*. [trad.] Javier Enríquez Brito Victor Campos Olgúin. 7ma. 2010. ISBN 9788448132149.

PROJECT. 2013. JavaScript Overview Mozilla Developer Network. [En línea] 2013. [Citado el: 23 de enero de 2018.] <https://developer.mozilla.org/en-US/docs/Web/JavaScript/Overview>.

Real Academia Española. 2014. Real Academia Española. *DRAG*. [En línea] 2014. [Citado el: 24 de octubre de 2017.] <http://lema.rae.es/drae/?val=reporte..>

Sampieri, R y Fernández , C. 2010. *Metodología de la investigación*. 2010. ISBN 978-607-15-0291-9.

Sánchez, Jennifer Garzón y Machado Hernández, José Carlos. 2016. *Componente para el diseño y visualización de reportes en la plataforma DAMIX*. Centro de Inteligencia de Negocio, Empresa de Tecnologías de la Información para la Defensa (XETID). La Habana : s.n., 2016. Tesis de pregrado.

Sánchez, S. 2015. *Estrategia de soporte técnico para el proceso de migración a código abierto en los Organismos de la Administración Central del Estado*. Universidad de las Ciencias Informáticas. La Habana : s.n., 2015. Tesis de maestría.

Scribd. 2016. Scribd. [En línea] 2016. [Citado el: 7 de enero de 2018.] <https://es.scribd.com/doc/58623687/Manual-de-Extjs-con-Ejemplo..>

SENAGAG. El GRAN PAITITÍ. [En línea] Bolivia. [Citado el: 13 de MARZO de 2018.] <http://190.129.48.189/egp/u022.html>.

SIGA: Sistema Integrado de Gestión Administrativa. 2011. *Módulo de Patrimonio*. Oficina General de Informática y Estadística, Ministerio de Economía y Finanzas. Perú : s.n., 2011. pág. 148, Manual de usuario.

SMILE. Open Source Guide. [En línea] [Citado el: 7 de enero de 2018.] [http://www.open-source-guide.com/en/Solutions/Applications/Business-intelligence-reporting/Pentaho-report-designe/..](http://www.open-source-guide.com/en/Solutions/Applications/Business-intelligence-reporting/Pentaho-report-designe/)

Somerville. 2005. *Ingeniería de Software*. 2005.

SQL Manager.net. 2017. EMS SQL Manager for PostgreSQL. [En línea] 12 de noviembre de 2017. [Citado el: 22 de marzo de 2018.]

- Tolosa, and Bordignon. 2008.** Introducción a la Recuperación de Información. [En línea] 2008. [Citado el: 27 de octubre de 2017.]
<http://admodeinformacion.blogspot.com/2011/09/introduccion-la-recuperacion-de.html..>
- Torre, Anibal de la. 2006.** PHPNuke. [En línea] 2006. [Citado el: 10 de enero de 2018.]
http://www.adelat.org/media/docum/nuke_publico/lenguajes_del_lado_servidor_o_cliente.html.
- UCID.** *Manual del "Caso de Estudio Usuario" par el MT Zeolides 2.0.* Centro de Desarrollo y Asimilación de Tecnologías (CDAT), Empresa de Tecnologías de la Información para la Defensa. Guía para el trabajo con el MT Zeolides.
- **2012.** *Proceso de Desarrollo y Gestión de Proyectos de Software.* Empresa de Tecnologías de la Información para la Defensa. La Habana : s.n., 2012.
- Una visión en el proceso de la toma de decisiones en las empresas del siglo XXI desde la gestión de la información.* **Pantoja, Liliana Ayala y Guido and Revelo, Luis Carlos. 2011.** 1, 1er Semestre, 2011, Revista de la Facultad de Ciencias Económicas y Administrativas, Vol. XII, págs. 92-118.
- Universidad de las Ciencias Informáticas.** Sitio oficial de la UCI. [En línea] [Citado el: 21 de febrero de 2018.] <http://www.uci.cu/investigacion-y-desarrollo/productos>.
- Viegas, Alejandro Torres. 2010.** *Módulo de reportes del Sistema Único de Identificación Nacional.* Universidad de las Ciencias Informáticas. La Habana : s.n., 2010. pág. 93, Tesis de pregrado.
- Yesith, Jaime Valencia. 2013.** *Modelo de sistema de información para apoyar la gestión de proyectos de investigación.* Universidad Industrial de Santander. Norte de Santander, Colombia : Scientia et Technica, 2013. ISSN 0122-1701.
- Zambrano, R. A. 2013.** *Metodología de la investigación.* 2013.
- Zend Technologies. 2006-2016.** ZendFramework. [En línea] 2006-2016. [Citado el: 7 de enero de 2018.] <http://framework.zend.com/manual/2.3/en/ref/overview.html..>
- Zorrilla, Martha Elena Pantaleón. 2010-2011.** UNICAN. [En línea] 2010-2011. [Citado el: 22 de febrero de 2018.] <http://personales.unican.es/zorrillm/BasesDatos/02%20-%20Modelos%20de%20datos%20ER-UML-relacional.pdf...>
- Zorrilla, Martha Elene Pantaleón. 2014.** *Modelo de datos.* 2014.

Anexo 1: Entrevistas

Estimados especialistas: Se necesita de su cooperación en una investigación para una tesis de pregrado. Para ello, sería de gran ayuda que respondiera las siguientes preguntas:

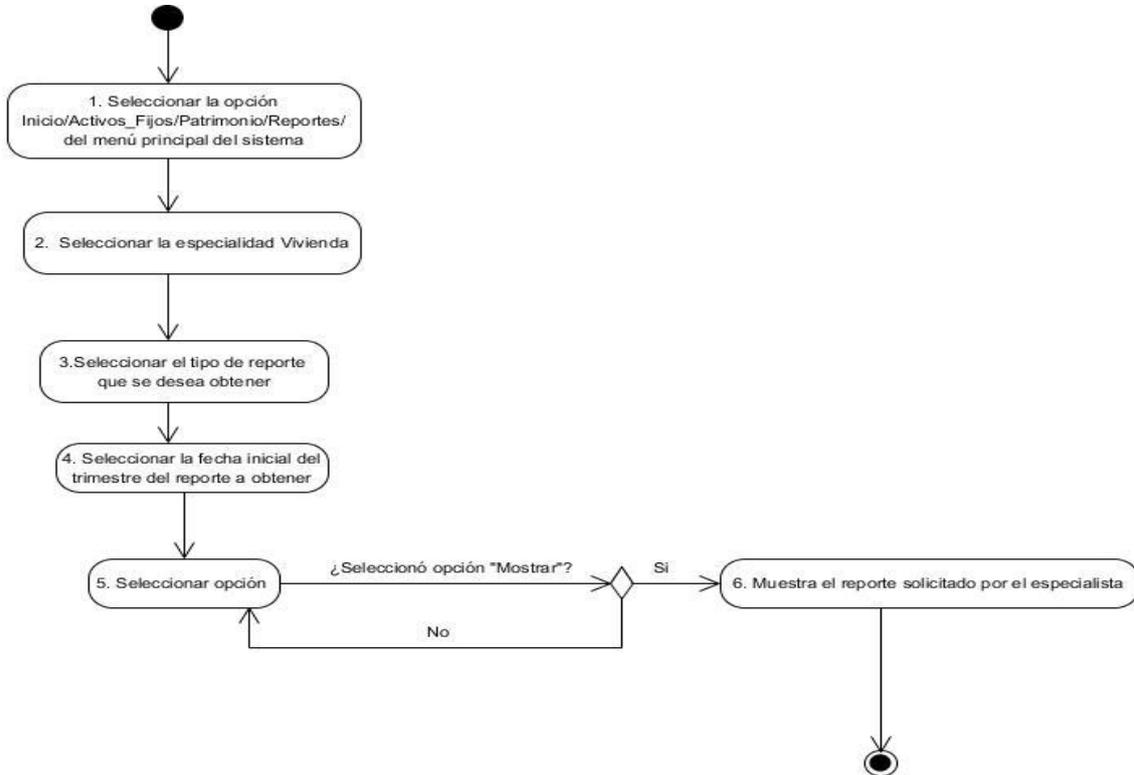
1. ¿Cómo se realizan actualmente los reportes en el subsistema?
2. ¿Qué características presentan los reportes actualmente?
3. ¿Cómo se manejan los datos del subsistema Patrimonio?
4. ¿Los procesos de los reportes son adecuados o necesitan redefinirse?
5. ¿Qué mejoras se podrían hacer para que responda aún mejor a las necesidades y deseos del cliente?

Estimado cliente: Se necesita de su cooperación para una investigación de pregrado. Para ello sería de gran ayuda que respondiera lo siguiente:

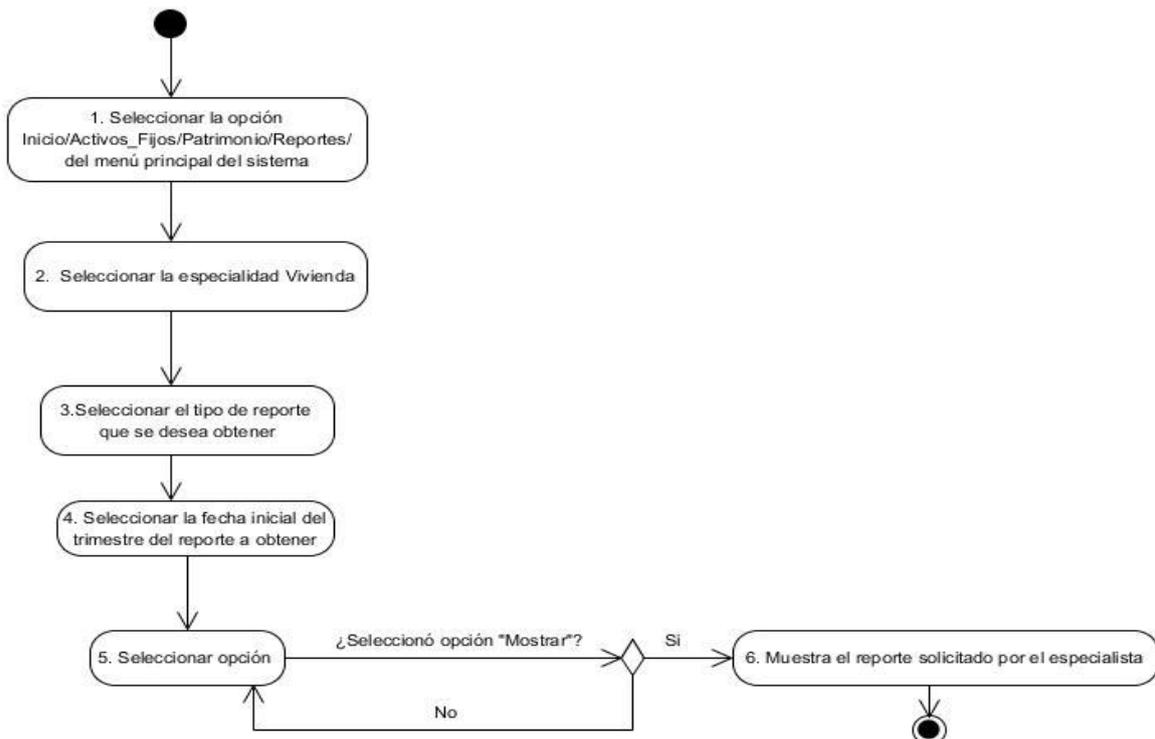
1. ¿Considera usted que el sistema responde con las demandas solicitadas?
2. ¿Con cuáles no cumple? ¿En qué aspectos se podría mejorar?
3. ¿Le aporta algún beneficio el sistema?

Anexo 2: Especificación de requisitos funcionales

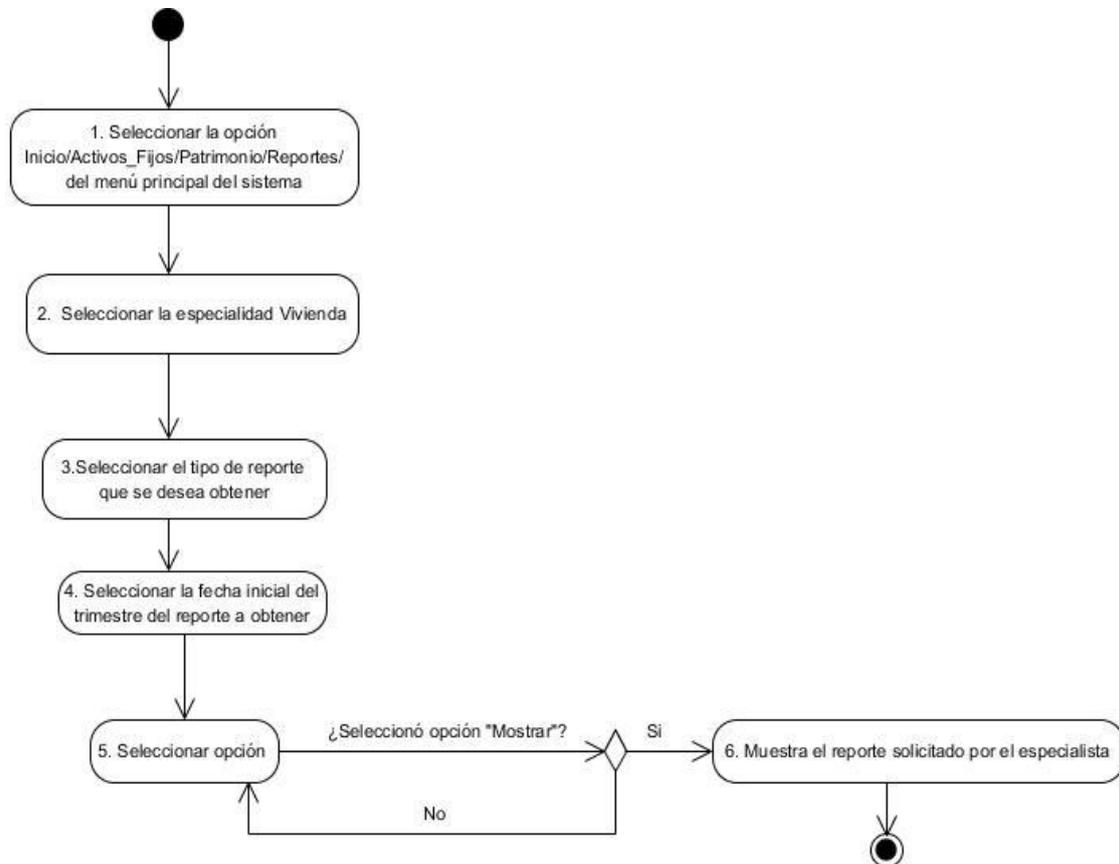
RF: Generar reporte especialidad Campamento



RF: Generar reporte especialidad Ingeniería

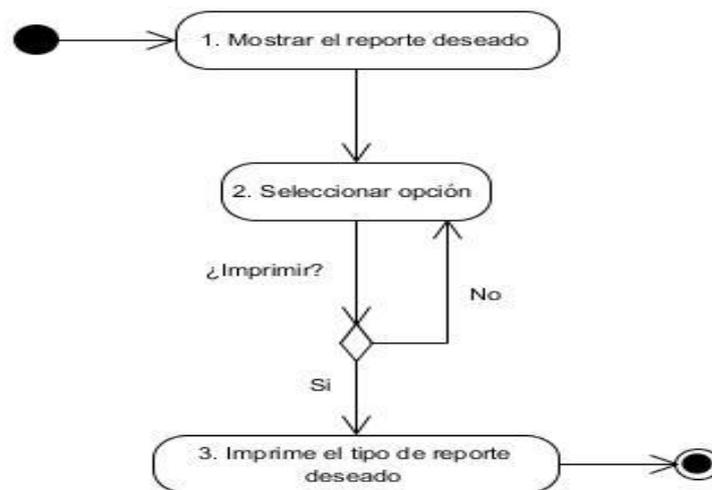


RF: Generar reporte especialidad Sistema Empresarial

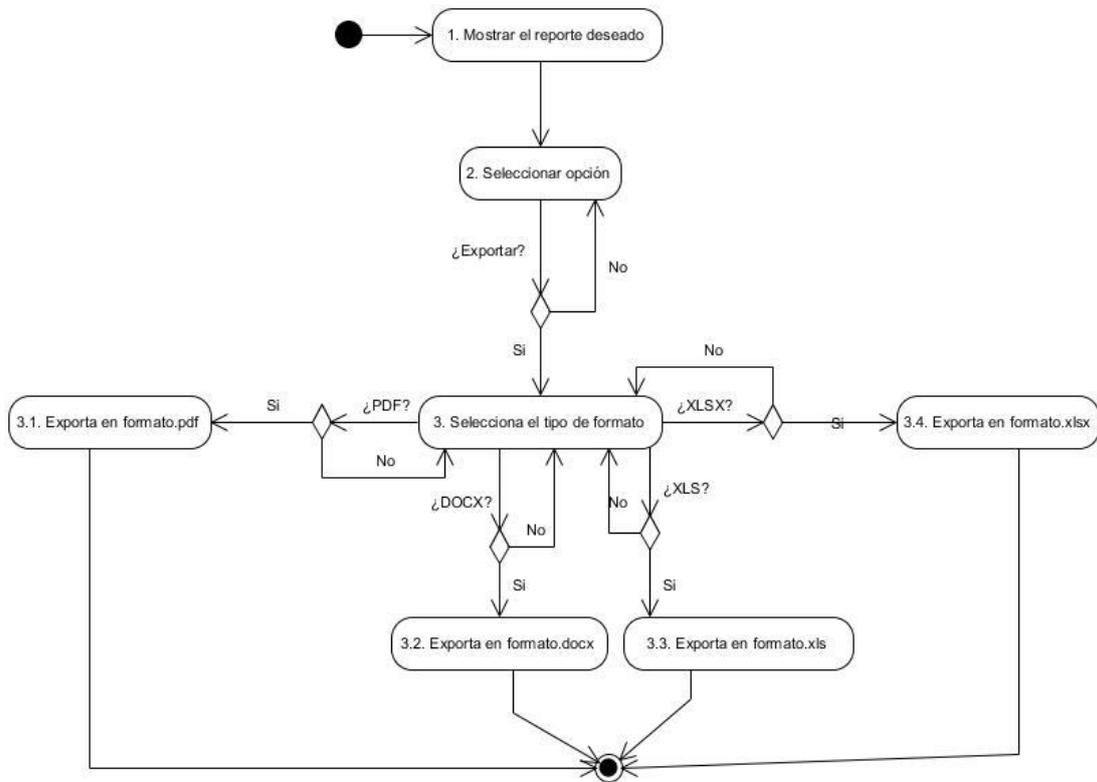


RF: Mostrar reporte

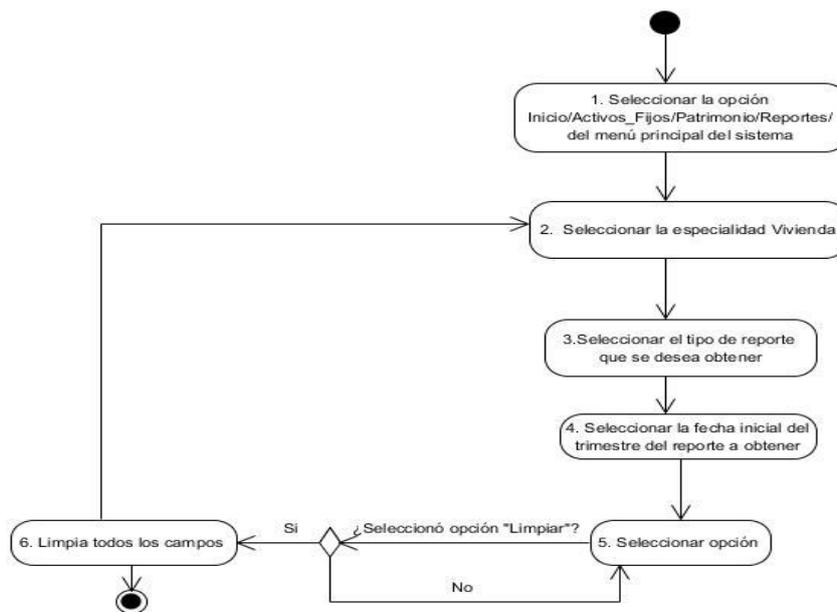


RF: Actualizar datos**RF: Imprimir reporte**

RF: Exportar reporte



RF: Limpiar datos



Anexo 3: Diagramas de estereotipos web

Diagrama de estereotipos web RF Generar reporte especialidad de Campamento

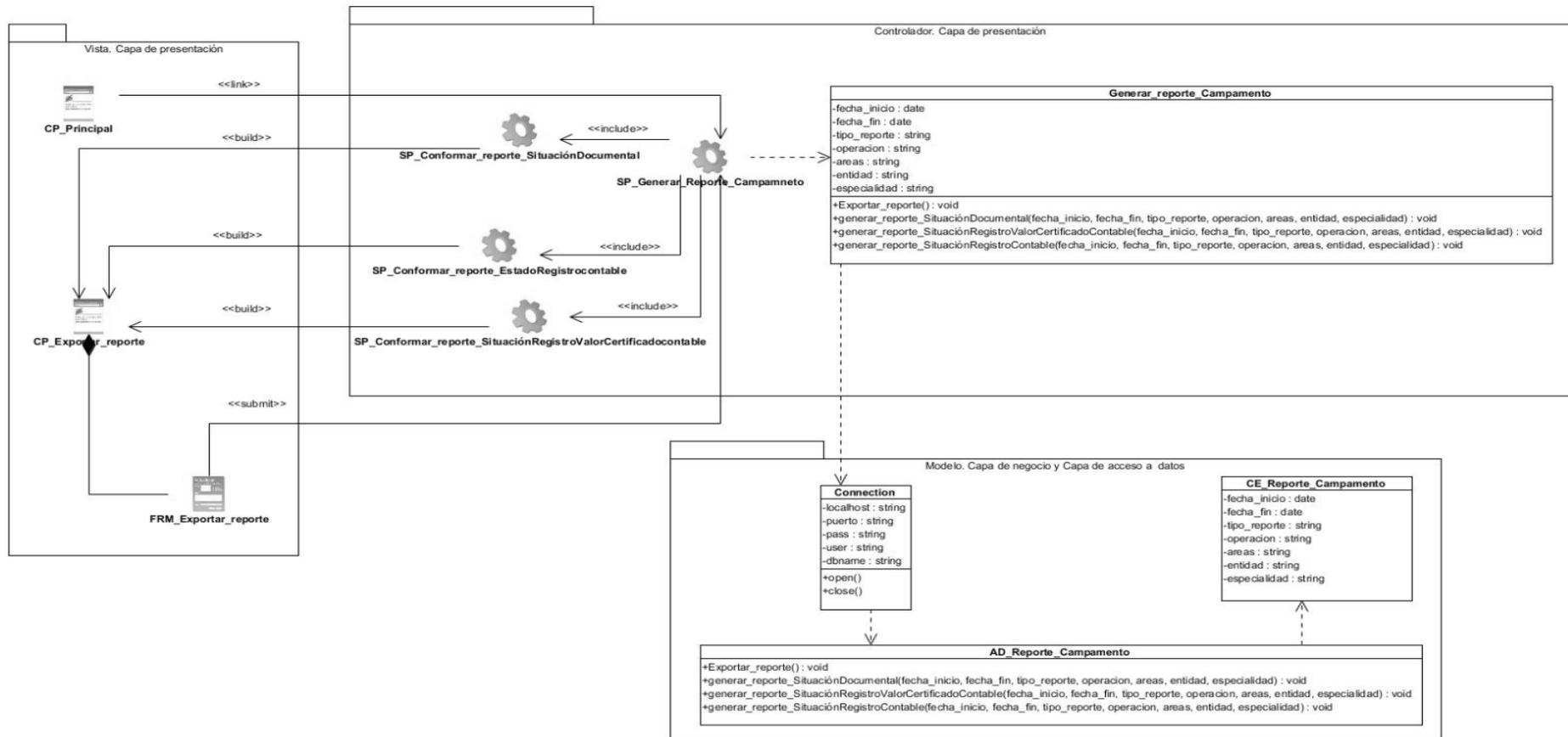


Diagrama de estereotipos web RF Generar reporte especialidad de Ingeniería

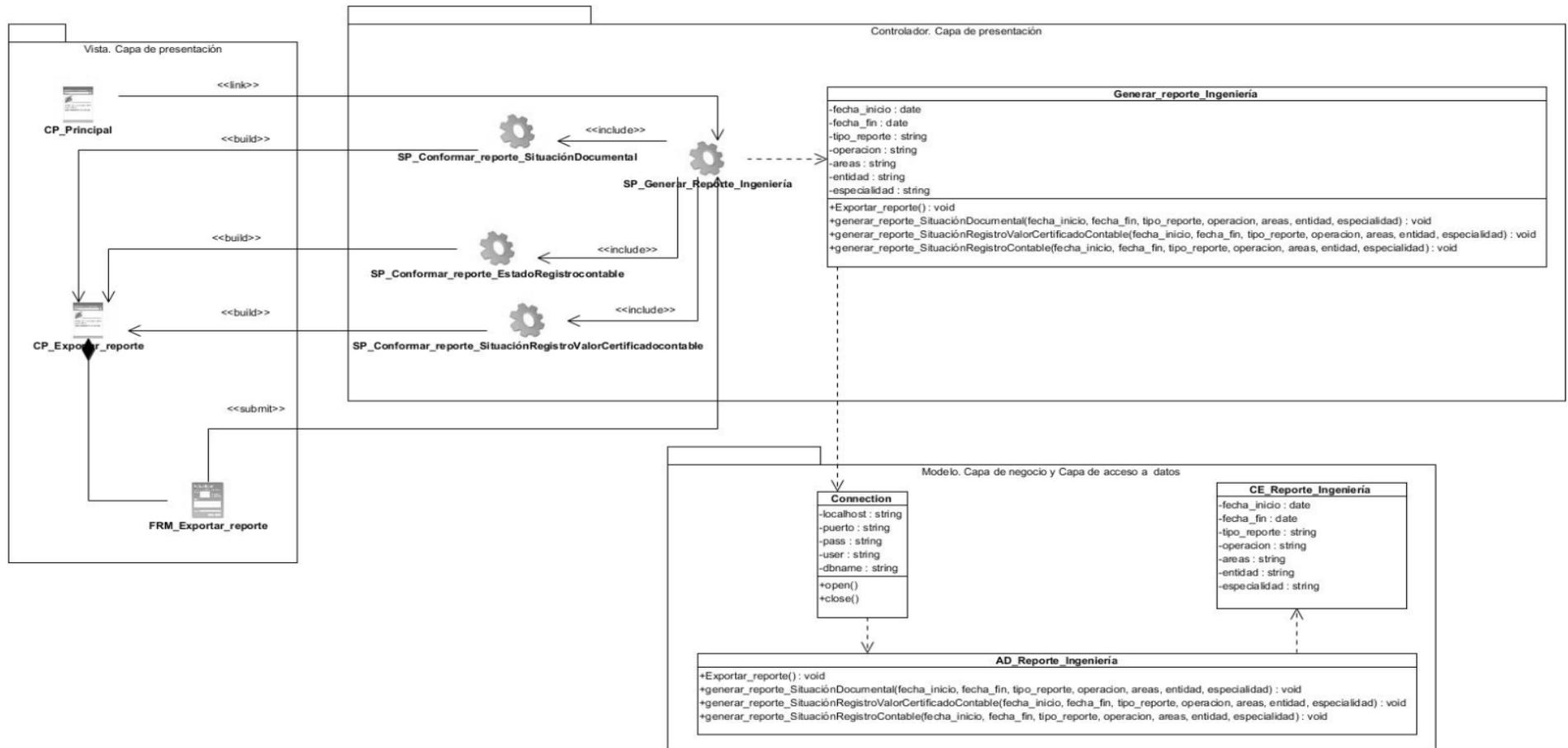
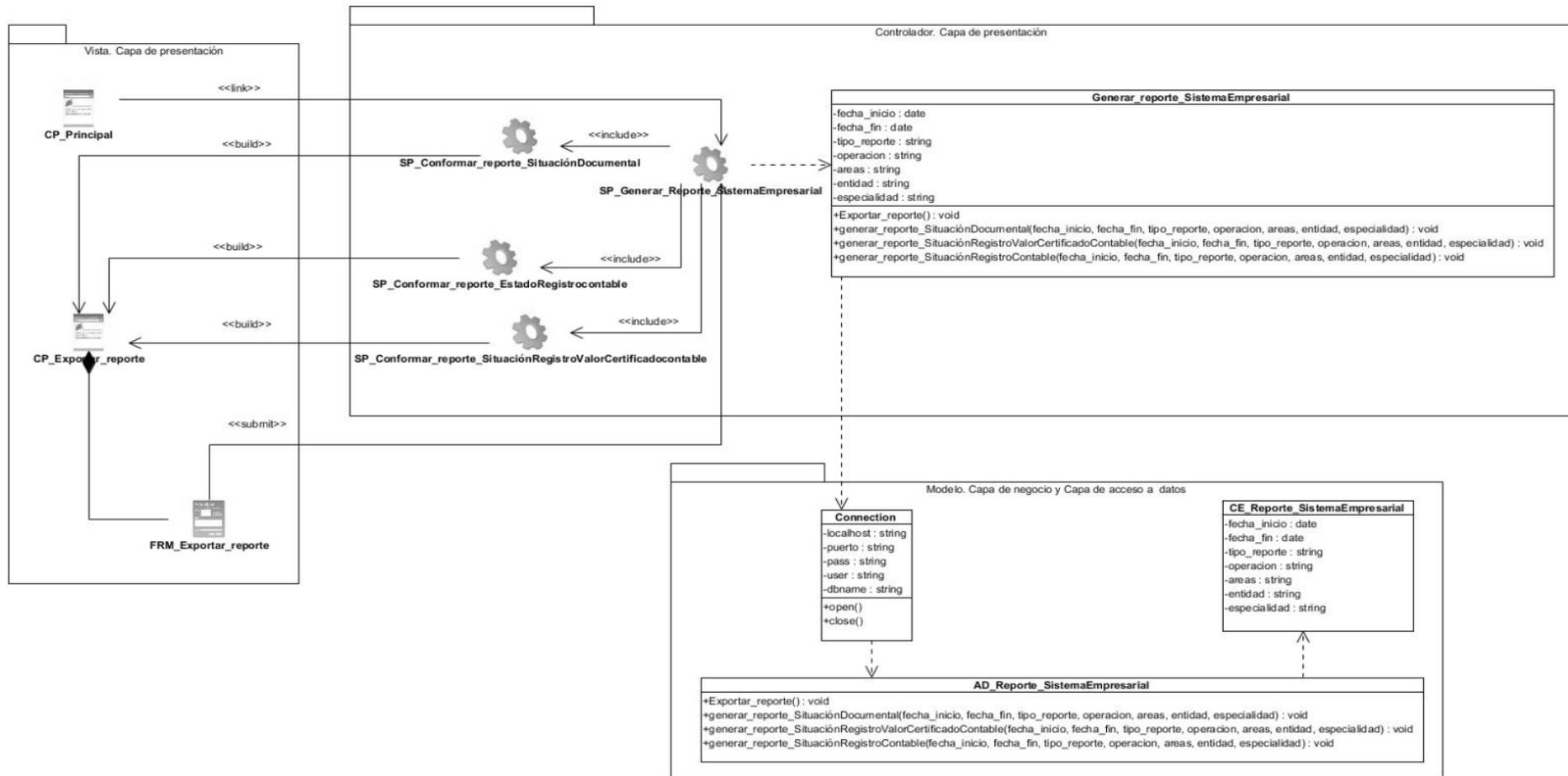


Diagrama de estereotipos web RF Generar reporte especialidad de Sistema Empresarial



Anexo 4: Encuesta para evaluar la validación de la idea a defender

Estimado especialista: Se necesita de su cooperación en una investigación para una tesis de pregrado para evaluar el nivel de aceptación del Módulo de reportes para el subsistema Patrimonio, por ello, sería de gran ayuda que respondiera lo siguiente, con la sinceridad y seriedad que el proceso necesita.

Emita su valoración del sistema respecto a los siguientes parámetros: MA, si valora el parámetro como Muy Adecuado; A, si lo valora como Adecuado; PA, si considera que es Poco Adecuado; o NA si considera que es No Adecuado. Marcar con una X.

Nro.	Parámetros	Valoración			
		MA	A	PA	NA
1	Tratamiento de los aspectos teóricos de la gestión de la información de los reportes en el sistema implementado.				
2	Nivel de apoyo que brinda a los especialistas del centro de Logística para la gestión de los reportes del subsistema Patrimonio				
3	Contribución del sistema al apoyo de la toma de decisiones.				
4	Presentación de una interfaz agradable e intuitiva para el usuario.				
5	Usabilidad del sistema.				