



Universidad de las Ciencias Informáticas

Facultad 1

Módulo de gestión Comercial para el Sistema de Asesoría de la Agencia de
Supervisión e Inspección de Cargas

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

Autor: Marbier Pérez Polo

Tutores: Ing. Gregorio Ferrer Córdova
Ing. Mariela Milagros Bony Fernández

La Habana, junio de 2017
“Año 59 de la Revolución”



"Me gusta la expresión "posibilidades perdidas". Nacer significa estar obligado a elegir una época, un lugar y una vida. Existir aquí, ahora, significa perder la posibilidad de ser una infinidad de otros potenciales seres."

Hayao Miyazaki

DECLARACIÓN DE AUTORÍA

Declaro por este medio que yo Marbier Pérez Polo, con carné de identidad 89062701146 soy el autor principal del trabajo titulado “Módulo de gestión Comercial para el Sistema de Asesoría de la Agencia de Supervisión e Inspección de Cargas” y autorizo a la Universidad de las Ciencias Informáticas a hacer uso de la misma en su beneficio, así como los derechos patrimoniales con carácter exclusivo.

Para que así conste firmamos la presente a los ____ días del mes de junio del año 2017.

Autor:

Marbier Pérez Polo

Tutores:

Ing. Gregorio Ferrer Córdova

Ing. Mariela Milagros Bony Fernández

DEDICATORIA

A mi mamá que ha sacrificado todo por mí y me ha guiado siempre por el mejor camino, que siempre está ahí cuando la necesito, y a la que amo por encima de todo en este mundo.

A mi hermana, que después de mi mamá es la mujer que más amo en el mundo, a la que me ha dado su amor y su apoyo incondicional.

A mi primo Paito, que más que un primo mi padre, que sé que está feliz de ver cómo me convierto en un profesional, que siempre me ha dado todo su amor y cariño.

AGRADECIMIENTOS

A mis padres por darme la vida y en especial a mi madre por haberme dado todo.

A toda mi familia por siempre estar pendiente, en especial mi abuela, hermana y Paito, por cuidarme y sobre todo por confiar en mí.

A mis amigos por ser geniales y por siempre estar pendiente de cada paso que doy.

A mis amigos de la UCI, por acompañarme y estar ahí cuando los necesitaba.

A mis compañeros de cuarto y los que fueron también; a mis vecinas y los que fueron vecinos.

A los miembros del tribunal por cada sugerencia para que este trabajo llegara al final.

A mis profesores desde primer año hasta quinto, algunos los mejores otros quizás no tan buenos, pero todos al final aportaron en mi formación un granito de arena y a todos les debo el haber llegado hasta aquí.

A todos aquellos que no se encuentran en la UCI, pero tuvieron la dicha de compartir gratos momentos conmigo.

A todos aquellos que en algún momento me preguntaron ¿y la tesis? A todos ustedes les debo este triunfo.

Para todos ustedes llegue mis eternos agradecimientos por soportarme tal y como soy;

MUCHAS GRACIAS.

RESUMEN

El auge de las nuevas tecnologías y la aplicación de la informática en el perfeccionamiento empresarial suscitan el desarrollo creciente en la rama tecnológica en el afán de optimizar el rendimiento de las mismas. En Cuba, la Universidad de las Ciencias Informáticas, es una de las entidades especializadas en la creación de herramientas tecnológicas (software) las cuales, no solo solventan los inconvenientes prácticos que ralentizan los procesos de una empresa, sino que además aceleran el rendimiento de las entidades. El presente trabajo de diploma consiste en un módulo de gestión Comercial para el Sistema de Asesoría de la Agencia de Supervisión e Inspección de Cargas. Se realizó una revisión bibliográfica sobre sistemas de gestión Comercial que aporten elementos técnicos para la solución. Para la implementación de la propuesta de solución se utiliza como metodología de desarrollo de software *Agile Unified Process*, en su versión UCI; como lenguaje de modelado UML y como herramienta *CASE Visual Paradigm*. El desarrollo de esta investigación es importante para ASIC porque se le proporciona un módulo que contribuye al proceso de contratación de servicios mediante las funcionalidades de autenticación de los usuarios, recursos humanos, gestión de la información, además de los sub-procesos que conllevan dichas funcionalidades. Las pruebas de software aplicadas arrojaron como resultado que el módulo trabaja con buen rendimiento, es confiable y fue de agrado para el cliente.

Palabras claves:

Información, gestión, gestión Comercial, contratación, servicios.

ÍNDICE

INTRODUCCIÓN	1
CAPÍTULO 1. FUNDAMENTACIÓN TEÓRICA DEL MÓDULO DE GESTIÓN COMERCIAL PARA EL SISTEMA DE ASESORÍA DE ASIC	5
1.1. CONCEPTOS ASOCIADOS AL DOMINIO DEL PROBLEMA.....	5
1.2. ESTUDIO DE SISTEMAS HOMÓLOGOS.....	6
1.3. RESULTADOS DE LA INVESTIGACIÓN DE LOS SISTEMAS ESTUDIADOS	9
1.4. HERRAMIENTAS Y TECNOLOGÍAS	10
1.5. CONCLUSIONES PARCIALES DEL CAPÍTULO	16
CAPÍTULO 2. ANÁLISIS Y DISEÑO DEL MÓDULO DE GESTIÓN COMERCIAL PARA EL SISTEMA DE ASESORÍA DE ASIC	18
2.1. PROPUESTA DE SOLUCIÓN	18
2.2. ANÁLISIS.....	19
2.3. DISEÑO.....	26
2.4. CONCLUSIONES PARCIALES.....	46
CAPÍTULO 3. VALIDACIÓN DEL MÓDULO DE GESTIÓN COMERCIAL PARA EL SISTEMA DE ASESORÍA DE ASIC	47
3.1. ESTÁNDAR DE CODIFICACIÓN (26)	47
3.2. PRUEBA	51
3.3. CONCLUSIONES PARCIALES.....	74
CONCLUSIONES	75
RECOMENDACIONES	76
BIBLIOGRAFÍA	77

ÍNDICE DE TABLAS

TABLA 1: REQUISITOS FUNCIONALES. (ELABORACIÓN PROPIA).....	20
TABLA 2: HU4 GESTIONAR SOLICITUD DE SERVICIOS. (ELABORACIÓN PROPIA)	28
TABLA 3: HU8 GESTIONAR RRHH. (ELABORACIÓN PROPIA).....	31
TABLA 4: VARIABLES PARA LOS CASOS DE PRUEBA FUNCIONAL GESTIONAR SOLICITUD DE SERVICIOS. (ELABORACIÓN PROPIA)	59
TABLA 5: CASO DE PRUEBA FUNCIONAL NUEVA SOLICITUD DE SERVICIOS. (ELABORACIÓN PROPIA)	60
TABLA 6: CASO DE PRUEBA FUNCIONAL ELIMINAR SOLICITUD DE SERVICIOS. (ELABORACIÓN PROPIA)	64
TABLA 7: CASO DE PRUEBA FUNCIONAL MODIFICAR SOLICITUD DE SERVICIOS. (ELABORACIÓN PROPIA)	65
TABLA 8: CASO DE PRUEBA GESTIONAR RRHH. (ELABORACIÓN PROPIA).....	70
TABLA 9: CASO DE PRUEBA GESTIONAR SOLICITUDES DE SERVICIOS. (ELABORACIÓN PROPIA)	72

ÍNDICE DE FIGURAS

FIGURA 1: MODELO CONCEPTUAL O DIAGRAMA DEL DOMINIO. (ELABORACIÓN PROPIA).....	20
FIGURA 2: DIAGRAMA DE CLASES. (ELABORACIÓN PROPIA).....	27
FIGURA 3: ARQUITECTURA MODELO-VISTA-PLANTILLA. (22).....	35
FIGURA 4: ARQUITECTURA MODELO-VISTA-PLANTILLA DEL SOFTWARE. (ELABORACIÓN PROPIA)	36
FIGURA 5: ESTRUCTURA DE CARPETAS DEL MÓDULO. (ELABORACIÓN PROPIA)	37
FIGURA 6: VISTA DE LA PÁGINA AUTENTICAR. (ELABORACIÓN PROPIA)	38
FIGURA 7: OBTENCIÓN DE LOS DATOS PARA VERIFICACIÓN. (ELABORACIÓN PROPIA).....	38
FIGURA 8: VERIFICACIÓN DE AUTENTICACIÓN. (ELABORACIÓN PROPIA)	39
FIGURA 9: VISTA DE LA PÁGINA DE INICIO. (ELABORACIÓN PROPIA)	39
FIGURA 10: EJEMPLO DEL USO DEL PATRÓN EXPERTO. (ELABORACIÓN PROPIA)	40
FIGURA 11: CÓDIGO DEL BLOQUEO E INICIO EN LA VISTA. (ELABORACIÓN PROPIA)	41
FIGURA 12: CÓDIGO DEL SALIR EN LA VISTA. (ELABORACIÓN PROPIA)	41
FIGURA 13: VISTA DEL USO DE DECORADORES. (ELABORACIÓN PROPIA)	42
FIGURA 14: DIAGRAMA DE CLASES DEL DISEÑO: GESTIONAR SOLICITUD DE SERVICIOS. (ELABORACIÓN PROPIA)	43
FIGURA 15: MODELO DE DATOS. (ELABORACIÓN PROPIA).....	44
FIGURA 16: DIAGRAMA DE COMPONENTE. (ELABORACIÓN PROPIA)	45
FIGURA 17: MODELO O DIAGRAMA DE DESPLIEGUE. (ELABORACIÓN PROPIA)	46
FIGURA 18: INTERFAZ DEL JMeter. (ELABORACIÓN PROPIA).....	54
FIGURA 19: RESULTADO DE LA PRUEBA DE RENDIMIENTO. (ELABORACIÓN PROPIA).....	54
FIGURA 20: PRUEBA DE SEGURIDAD: 1RA ITERACIÓN. (ELABORACIÓN PROPIA)	56
FIGURA 21: PRUEBA DE SEGURIDAD: 2DA ITERACIÓN. (ELABORACIÓN PROPIA)	57
FIGURA 22: PRUEBA DE SEGURIDAD: 3RA ITERACIÓN. (ELABORACIÓN PROPIA)	57
FIGURA 23: COMPORTAMIENTO DE LAS NO CONFORMIDADES POR ITERACIÓN. (ELABORACIÓN PROPIA)	69

INTRODUCCIÓN

El escenario actual está cambiando rápidamente como consecuencia de las sustanciales transformaciones que se perciben en las nuevas tecnologías, cambios estos que a su vez modifican las maneras en el que el hombre visualiza, orienta y proyecta su existencia. En buena medida este estado de transformaciones está constituido por constantes innovaciones tecnológicas (I+D) y la acelerada adopción de nuevos dispositivos que promueve el mercado de consumo. La rápida expansión de las Tecnologías de la Información y la Comunicación (TIC) obliga a repensar prácticamente todos los ejes de desarrollo económico. Esta era marcada por la digitalización (capitalismo soportado y amplificado a través de las nuevas tecnologías digitales) y la hiper-conectividad de nuestros días, uso crítico de la información y, sobre todo, de la producción de nuevo conocimiento ha adquirido protagonismo como eje impulsor del desarrollo (*World Bank Institute, 2008*). (1)

Según plantean Syrjänen y Pathan (2008) es fundamental entender el valor estratégico del acceso a la información y con ello a nuevas posibilidades de aprendizaje. El uso apropiado de las TIC puede generar valor agregado en diversos escenarios, siempre y cuando este uso venga acompañado y enriquecido por un conjunto de habilidades, saberes y valores. (2)

Se concibe el término de Tecnologías de la Información y la Comunicación (TIC) como: “Dispositivos tecnológicos (hardware y software) que permiten editar, producir, almacenar, intercambiar y transmitir datos entre diferentes sistemas de información que cuentan con protocolos comunes. Estas aplicaciones, que integran medios de informática, telecomunicaciones y redes, posibilitan tanto la comunicación y colaboración interpersonal (persona a persona) como la multidireccional (uno a muchos o muchos a muchos). Estas herramientas desempeñan un papel sustantivo en la generación, intercambio, difusión, gestión y acceso al conocimiento. En este sentido muchas empresas se benefician de los aportes sistemáticos de la informática.” (2)

Sin embargo, en la actualidad, dichas entidades demandan continuamente determinados parámetros de rentabilidad, calidad y tecnología que contribuyan con el objetivo de tributar al desarrollo sostenible. Un Sistema de Gestión eficiente puede ayudar a convertir esas presiones en una ventaja competitiva pues las empresas buscan la evaluación y certificación independiente de sus sistemas de gestión como medio para

evidenciar el cumplimiento; además, la dirección de la empresa tiene un valor añadido en la cual una tercera parte auditara sus sistemas de gestión de forma independiente y evidenciara el resultado por medio de un certificado. (3)

Enmarcado en el estudio y uso de las TIC se encuentra la Universidad de las Ciencias Informáticas(UCI) que es un centro cubano de altos estudios con la misión de: Formar profesionales comprometidos con su Patria y altamente calificados en la rama de la Informática. Producir aplicaciones y servicios informáticos, a partir de la vinculación estudio – trabajo como modelo de formación. Servir de soporte a la industria cubana de la informática. (Portal UCI, 2010)

Para lograr esta meta la universidad cuenta con una Red de Centros de desarrollo de software entre los que se encuentra el Centro de Identificación y Seguridad Digital(CISED); al cual se le ha asignado el proyecto de realizar un sistema de gestión para la Agencia de Supervisión e Inspección de Cargas(ASIC).

La Agencia de Supervisión e Inspección de Cargas ofrece servicios de inspección y evaluación de cargas en varios lugares del país, lo cual conlleva una alta demanda de sus esfuerzos, incluida la gestión de los servicios que brindan. El proceso de levantamiento de información durante la contratación de sus clientes y proveedores no se realiza de forma eficiente, por lo que se genera un gran cúmulo de documentación que debe ser manejada en formato duro, aumentando el riesgo de deterioro y/o pérdida de algún documento. Lo mismo sucede con la ejecución de los servicios, ya que deben elaborarse solicitudes de servicios, contratos, órdenes de trabajo y un registro de evaluación de la satisfacción de los clientes atendidos. A pesar de que cada servicio es facturado, la gestión económica, de clientes, proveedores y servicios, se lleva a cabo por vías distintas, por lo que una misma información necesita ser cotejada en varios documentos al mismo tiempo. Esta misma problemática se refleja en las demás instancias de la agencia distribuidas a lo largo del país, por lo que otra de las necesidades de la agencia es la unificación del proceso de contratación a nivel nacional.

En correspondencia con lo antes expuesto se plantea como **problema de investigación**: ¿Cómo contribuir a la gestión Comercial en el proceso de contratación de servicios de la Agencia de Supervisión e Inspección de Cargas? Se define como **objeto de estudio**: La gestión de información en el proceso de contratación de servicios; como **campo de acción**: El proceso de contratación de servicios en la Agencia de Supervisión e Inspección de Carga.

Para dar solución al problema existente la presente investigación se traza como **objetivo general**: Desarrollar un módulo de gestión comercial para el proceso de contratación de servicios de la Agencia de Supervisión e Inspección de Cargas.

Para alcanzar este objetivo general se han establecido los siguientes **objetivos específicos**:

1. Elaborar el marco teórico-conceptual de la investigación a partir a partir de los supuestos teóricos que respaldan la investigación.
2. Definir las tecnologías, las herramientas y la metodología de desarrollo necesarias para la implementación del módulo de gestión Comercial.
3. Diseñar el módulo de gestión Comercial para el desarrollo del módulo de contratación de servicios.
4. Implementar el módulo de gestión Comercial para el Sistema de Asesoría de la Agencia de Supervisión e Inspección de Cargas.
5. Validar mediante pruebas funcionales los resultados obtenidos con la solución para su posterior despliegue.

Preguntas científicas:

1. ¿Cuáles son los presupuestos teóricos que sustentan la implementación del módulo de gestión Comercial para el Sistema de Asesoría de la Agencia de Supervisión e Inspección de Cargas?
2. ¿Cuáles tecnologías, herramientas y metodología se requieren implementar para el módulo de gestión Comercial para el Sistema de Asesoría de la Agencia de Supervisión e Inspección de Cargas?
3. ¿Cuáles artefactos que se requieren por la metodología de desarrollo seleccionada?
4. ¿Qué concepción debe tener el módulo de gestión Comercial para el Sistema de Asesoría de la Agencia de Supervisión e Inspección de Cargas?
5. ¿Cómo validar la contribución del módulo de gestión Comercial para el Sistema de Asesoría de la Agencia de Supervisión e Inspección de Cargas?

Para ello se plantean las siguientes **tareas de la investigación**:

1. Presentación del diseño teórico-metodológico de la investigación.

2. Análisis de las tendencias actuales relacionadas con los sistemas de gestión/contratación de servicios.
3. Selección de las tecnologías, las herramientas y la metodología necesarias para implementar el módulo de gestión Comercial para el Sistema de Asesoría de la Agencia de Supervisión e Inspección de Cargas.
4. Diseño de los artefactos requeridos por la metodología de desarrollo seleccionada.
5. Implementación del módulo de gestión Comercial para el Sistema de Asesoría de la Agencia de Supervisión e Inspección de Cargas.
6. Ejecución de pruebas para validar el sistema desarrollado.

Para el desarrollo de esta investigación se utilizan los siguientes **métodos teóricos**:

- ✚ Modelación: Para la modelación de cada uno de las plantillas o interfaces, realización de diagramas, tablas, graficas que permiten la abstracción de la realidad al modelo informático y de ahí buscar la manera de implementarlo, concretando las características de cada uno de los contenidos seleccionados.
- ✚ Histórico Lógico: Se utiliza en la búsqueda de conocimiento antepuestos sobre el tema solución y se formula una solución lógica utilizando la base del conocimiento antes ya existentes sobre la temática que se está investigando.
- ✚ Sistematización: Se fructifica el conocimiento acumulado y se usa como base para dar solución al tópico abordado en la investigación.

Además de los siguientes **métodos empíricos**:

- ✚ Técnicas de recolección de información:
 - ❖ Revisión documental: Permite analizar los documentos publicados y presentados por la ASIC en el espacio creado por el CISED y el proyecto para el caso en solución y determinar las características que deben tener el sistema para poder proporcionar una solución.

CAPÍTULO 1. FUNDAMENTACIÓN TEÓRICA DEL MÓDULO DE GESTIÓN COMERCIAL PARA EL SISTEMA DE ASESORÍA DE ASIC

En este capítulo se exponen conceptos fundamentales asociados al dominio del problema planteado. Se presenta una síntesis de los principales estudios acerca de la utilización de los sistemas de gestión Comercial y logística para la solución del problema presentado. Por último, se analizan las metodologías, tecnologías y herramientas utilizadas durante el ciclo de desarrollo de la solución que se propone.

1.1. Conceptos asociados al dominio del problema

Definición de contratación

La contratación es el proceso mediante el cual se realiza una transacción en la que una parte se compromete a transferir recursos económicos a cambio de la recepción de un determinado servicio. (2)

Definición de servicios

Son un conjunto de **acciones** las cuales son realizadas para **servir** a alguien, algo o alguna causa. Los servicios son funciones ejercidas por las personas hacia otras personas con la finalidad de que estas cumplan con la satisfacción de recibirlos. (5)

Definición de recursos humanos(RRHH)

Son el conjunto de empleados y colaboradores que trabajan en una empresa u organización; también se le denomina al departamento, a la persona o áreas administrativas que se encarga de seleccionar, contratar y pagar a los trabajadores. (6)

Sistemas de gestión

Los sistemas de gestión se definen como el conjunto interrelacionado de elementos (métodos, procedimientos, instrucciones, etcétera), mediante los que la organización planifica, ejecuta y controla determinadas actividades relacionadas con los objetivos que desea alcanzar. (7)

Gestión Comercial

Según (8) la gestión Comercial es la que lleva a cabo la relación de intercambio de la empresa con el mercado. (9)

Sistema de gestión Comercial(CGM)

Son un enfoque estratégico de gestionar el valor y las experiencias con los clientes. Su tecnología es un facilitador de los beneficios que llegarán de aplicar una estrategia comercial adecuada, una gran capacidad analítica y unas políticas innovadoras. (3)

1.2. Estudio de sistemas homólogos

comforce® (Contract Management & Tracking Software)



Esta herramienta presenta un sistema de gestión de contratos que realiza administración de documentos, genera alarmas o alertas para vencimientos de contratos. Presenta un filtro de búsqueda de contratos y permite un control de los usuarios, guardando nombres y fechas; además, de un control de las cuentas y costos de los contratos.

El sistema a pesar de poseer una administración de servicios no presenta un manejo de facturas, ordenes de trabajo, registro de satisfacción del cliente. Estos aspectos son necesarios para la propuesta solución. Esta herramienta no posee un gestor de recursos humanos y medios.

Ctaimacae.NET



Es una aplicación *online* que facilita el trabajo de empresas contratistas y ayuda a la eficacia y eficiencia de las actividades de las empresas. Presenta módulos de gestión documental, contratos, recursos humanos y medios. La herramienta presenta la ventaja que maneja de forma eficaz la información generada en las contrataciones, presenta manejo de fechas de contratos, además de alertas de proximidad de vencimiento de los contratos. Se gestionan los documentos e informaciones que se generan de los contratos.

La aplicación no presenta un manejo de las ordenes de trabajo, aunque si de las tareas a realizar por cada contrato, no ostenta una gestión de las facturas y cuentas. Además, no se gestiona la satisfacción del cliente.

EASY Software



Esta herramienta presenta un módulo de contratación llamado EASY Contract. Este módulo presenta una gama de beneficios en pos de satisfacer las necesidades de los clientes y proveedores:

- ✓ Gestión de todo tipo de contrato
- ✓ Manejo fácil de la documentación e información.
- ✓ Reduce errores y tiempos del proceso de creación de los contratos.
- ✓ Presenta un informe detallado de cada contrato.
- ✓ Manejo de los costos.
- ✓ Manejo de los límites de vencimiento de los contratos a través de alertas o alarmas de aviso.
- ✓ Presenta un control de usuarios.

Se presencia una gran gama de beneficios, pero no se cumplen con todas las especificaciones de la empresa ASIC, ya que no presenta un control de facturas, ordenes de trabajo, gestión de recursos humanos y medios.

PACTUM



Pactum es el software en la nube, producido por *Skina IT Solutions*, orientado a llevar una trazabilidad sobre los contratos de una empresa de cualquier tamaño. Pactum apoya la gestión de un contrato y brinda trazabilidad durante todo el proceso de contratación. El software presenta un almacenamiento centralizado y organizado, gestión de contratos, alertas ante vencimiento de contratos, aviso a supervisores cuando haya retrasos. Disminución al máximo del uso del papel.

El software no ofrece una solución viable a las exigencias de la empresa ASIC, pues la misma no presenta un soporte de nube, además el software no presente un manejo de los costos, facturas, ordenes de trabajo, RRHH, medios y registros de satisfacción.

ServiceTonic



ServiceTonic es un software de *Help Desk* para la gestión de activos y automatización de servicios. Una plataforma de software potente, flexible y fácil de usar que le ayudará a proporcionar a sus clientes un servicio excelente de una forma eficiente.

Presenta las funcionalidades de gestión de contrato, registro de satisfacción del cliente, presenta un gestor de RRHH, gestión de vencimiento de contratos. La potencia, flexibilidad y capacidad multiservicio de servicetonic le permite definir y gestionar cualquier servicio con su propia configuración. Esta herramienta no presenta un manejo de órdenes de trabajo, de costos, de medios, facturas.

Signaturit



Signaturit es una herramienta digital online utilizada para la gestión y contratación de recursos humanos y algunos servicios. Esta herramienta consta con un proceso netamente digital, es decir, tanto como documentación, información, firmas, contactos, seguimientos y otras opciones de realizan de forma electrónica.

Esta herramienta no es factible para la solución de la agencia ya que presenta en valor de soporte, mantenimiento, precios, muy altos costos; además, no hay un manejo ni control en el proceso de

contratación de servicios ya que no contienen manejo de facturas, cuentas, registro de satisfacción del cliente, no se manejan los medios.

Cezanne HR (*Human Resources*)



Cezanne HR es una herramienta para la gestión y contratación de recursos humanos de forma online. Puedes añadir módulos en cualquier momento y utilizar las API's (*Application Programming Interfaces*¹) para conectarte con otros sistemas. Contiene una gestión documental, presenta un manejo de anuncios de ofertas de trabajo. Este sistema puede ser alojado o utilizado utilizando las tecnologías de la nube.

El sistema presenta una buena gestión y contratación de los recursos humanos, pero carece de la gestión de la contratación de servicios y por ende los procesos que conllevan dicho asunto. Además, no presenta una gestión de los medios; también utiliza tecnologías muy complejas de altos costos de soporte y mantenimiento.

1.3. Resultados de la investigación de los sistemas estudiados

Los sistemas estudiados arrojaron los siguientes resultados:

- ✚ Los sistemas **Conforce**, **Ctaimacae**, **EasySoftware** y **Pactum** presentan una parcial solución a lo planteado por ASIC, por lo que dificulta a la hora de su uso como solución.
- ✚ Los sistemas **Signaturit** y **Cezanne** presentan altos grados de tecnologías y necesitan de altos costos de soporte y mantenimiento, ya que utilizan como base tecnológica el uso de la nube y herramientas con licencias privativas de altos costos; además, no presentan una completa solución a lo propuesto por ASIC.

¹ Interfaces de programación de aplicaciones.

- ✚ EL sistema **ServiceTonic** presenta los módulos de contratación de servicios y recursos humanos, pero no cumplen con las especificaciones del cliente ya que no cuentan en su totalidad con los procesos planteados por el cliente.

1.4. Herramientas y tecnologías

1.4.1. Metodología de desarrollo de software

Marco conceptual

El Proceso Unificado Ágil de Scott Ambler o *Agile Unified Process* (AUP) en inglés es una versión simplificada del Proceso Unificado de Rational (RUP, por sus siglas en inglés). Este describe de una manera simple y fácil de entender la forma de desarrollar aplicaciones de software de negocio usando técnicas ágiles y conceptos que aún se mantienen válidos en RUP. El AUP aplica técnicas ágiles incluyendo:

- ✚ Desarrollo Dirigido por Pruebas²
- ✚ Modelado ágil
- ✚ Gestión de Cambios ágil

Refactorización de Base de Datos para mejorar la productividad.

Al igual que en RUP, en AUP se establecen cuatro fases que transcurren de manera consecutiva.

Fases de AUP

1. Inicio: El objetivo de esta fase es obtener una comprensión común cliente-equipo de desarrollo del alcance del nuevo sistema y definir una o varias arquitecturas candidatas para el mismo.
2. Elaboración: El objetivo es que el equipo de desarrollo profundice en la comprensión de los requisitos del sistema y en validar la arquitectura.
3. Construcción: Durante la fase de construcción el sistema es desarrollado y probado al completo en el ambiente de desarrollo.

² *Test Driven Development* (TDD, en inglés)

4. Transición: El sistema se lleva a los entornos de preproducción donde se somete a pruebas de validación y aceptación y finalmente se despliega en los sistemas de producción.

Variación de AUP para la UCI

No existe una metodología de software universal, ya que toda metodología debe ser adaptada a las características de cada proyecto (equipo de desarrollo, recursos, etc.) exigiéndose así que el proceso sea configurable. Se decide hacer una variación de la metodología AUP, de forma tal que se adapte al ciclo de vida definido para la actividad productiva de la UCI. Una metodología de desarrollo de software tiene entre sus objetivos aumentar la calidad del software que se produce, de ahí la importancia de aplicar buenas prácticas, para ello nos apoyaremos en el Modelo CMMI-DEV v1.3. El cual constituye una guía para aplicar las mejores prácticas en una entidad desarrolladora. Estas prácticas se centran en el desarrollo de productos y servicios de calidad.

1.4.2. Lenguaje de modelado

El modelado constituye una simplificación de la realidad donde se define lo esencial para la construcción del software con los objetivos de comunicar la estructura de un sistema complejo, especificar el comportamiento deseado del sistema, comprender mejor lo que se está desarrollando y descubrir oportunidades de simplificación y reutilización (10).

UML es un lenguaje de modelado orientado a objetos que permite representar gráficamente los elementos estáticos y dinámicos de una aplicación software. (11)

Las ventajas del lenguaje UML son:

- ✚ Es estándar, lo que significa que facilita la comunicación.
- ✚ Está basado en el meta modelo con una semántica bien definida.
- ✚ Se basa en una notación gráfica concisa y fácil de aprender y utilizar.
- ✚ Se puede utilizar para modelar software en distintos dominios: sistemas de información empresariales, sistemas Web, sistemas críticos y de tiempo real, etc.
- ✚ Es fácilmente extensible.

1.4.3. Lenguajes de programación

Se utilizará Python en su versión 3.4, pues fue orientada por el proyecto y este lenguaje posee las siguientes ventajas:

- ✚ Propósito general: Se pueden crear distintos tipos de programas.

- ✚ Multiplataforma: Hay versiones disponibles de Python en muchos sistemas informáticos distintos.
- ✚ Interpretado: No se debe compilar el código antes de su ejecución. En realidad, sí que se realiza una compilación, pero esta se realiza de manera transparente para el programador.
- ✚ Interactivo: Dispone de un intérprete por línea de comandos en el que se pueden introducir sentencias.
- ✚ Orientado a objetos: La programación orientada a objetos está soportada en Python y ofrece en muchos casos una manera sencilla de crear programas con componentes reutilizables.
- ✚ Funciones y librerías: Dispone de muchas funciones incorporadas en el propio lenguaje, para el tratamiento de strings, números, archivos, etc. Además, existen muchas librerías que se pueden importar en los programas para tratar temas específicos.
- ✚ Sintaxis clara: Tiene una sintaxis muy visual, gracias a una notación indentada (con márgenes) de obligado cumplimiento.
- ✚ Mixto: Se puede integrar de manera "fácil" con otros lenguajes de programación.
- ✚ Gratuito: Una ventaja fundamental de Python es la gratuidad de su intérprete, se puede descargar desde la página web: <http://www.python.org>.

1.4.4. Base de Datos

Se utilizará PostgreSQL en su versión 9.4 ya que se propuso por parte del centro y por el proyecto; además, cuenta con un rico conjunto de tipos de datos, permitiendo además su extensión mediante tipos y operadores definidos y programados por el usuario. Presenta una API de acceso al SGBD disponible en Python. Su administración se basa en usuarios y privilegios. Implementación de algunas extensiones de orientación a objetos. En PostgreSQL es posible definir un nuevo tipo de tabla a partir de otra previamente definida.

1.4.5. Herramienta CASE

Visual Paradigm for uml es una herramienta de Ingeniería de Software Asistida por Computación (CASE, por sus siglas en inglés). La misma propicia un conjunto de ayudas para el desarrollo de programas informáticos, desde la planificación, pasando por el análisis y el diseño, hasta la generación del código fuente de los programas y la documentación (12).

Se utilizará Visual Paradigm CE en su versión 13.0 pues fue orientada por el proyecto y, además, posee las cualidades de diseñar y modelar de una forma más clara el negocio en cuestión para una mejor comprensión del mismo.

1.4.6. Framework de desarrollo

Se utilizará para el desarrollo de la aplicación Django 1.8 que es un marco de trabajo de desarrollo web totalmente implementado sobre Python, con el que se pueden crear y mantener aplicaciones de alta calidad. Incluye un servidor web ligero que se puede usar mientras se desarrolla. Al mismo tiempo, Django permite trabajar fuera de su ámbito según sea necesario. Django ofrece las siguientes facilidades:

- ✚ Sistema de plantillas para separar la presentación de un documento de sus datos.
- ✚ Construcción automática de interfaces de administración.
- ✚ Vistas genéricas que recogen ciertos estilos y patrones comunes en su desarrollo y los abstraen, de modo que se puede escribir rápidamente vistas comunes de datos sin tener que escribir mucho código.
- ✚ Sistema de caché robusto y con un nivel de granularidad ajustable, que permite guardar páginas dinámicas para que no tengan que ser recalculadas cada vez que se piden.
- ✚ Integración con bases de datos y aplicaciones existentes.
- ✚ Construcción de aplicaciones multilingaje permitiendo especificar cadenas de traducción de más de cuarenta idiomas.

Se utilizará para el desarrollo de la aplicación *HTML 5* (siglas en inglés de *HyperText Markup Language*) ya que es un lenguaje de marcado predominante para la elaboración de páginas web. Es usado para describir la estructura y el contenido en forma de texto, así como para complementar el texto con objetos tales como imágenes. El HTML se escribe en forma de «etiquetas», rodeadas por corchetes angulares (<,>). Se utilizará ya que para Para publicar información para su distribución mundial, uno necesita un lenguaje universalmente conocido, una especie de lengua materna que todos los ordenadores pueden comprender potencialmente. El lenguaje de publicación usado por la *World Wide Web*(WWW) es HTML. (13)

Se utilizará para el desarrollo de la aplicación Hojas de estilo en cascada (CSS, siglas en inglés de *Cascading Stylesheets*) ya que es un lenguaje que describe la presentación de los documentos estructurados en hojas de estilo para diferentes métodos de interpretación. CSS es una especificación

desarrollada por el W3C (*World Wide Web Consortium*) para permitir la separación de los contenidos de los documentos escritos en HTML, XML, XHTML, SVG, o XUL de la presentación del documento con las hojas de estilo, incluyendo elementos tales como los colores, fondos, márgenes, bordes, tipos de letra, modificando la apariencia de una página web de una forma más sencilla, permitiendo a los desarrolladores controlar el estilo y formato de sus documentos. (13)

Se utilizará para el desarrollo de la aplicación JavaScript ya que es un lenguaje de programación que se puede utilizar para construir sitios Web y para hacerlos más interactivos. El lenguaje JavaScript puede interactuar con el código HTML, permitiendo a los programadores web utilizar contenido dinámico. El lenguaje JavaScript es *opensource*, por lo cualquier persona puede utilizarlo sin comprar una licencia. (14)

1.4.7. IDE³ de desarrollo

Se utilizará para la implementación JetBrains PyCharm en su versión 2016.3.2 que es un IDE basado en IntelliJ IDEA, que ofrece las siguientes funciones:

- ✚ Auto-completamiento de código.
- ✚ Señalamiento de errores con soluciones fáciles.
- ✚ Posibilita una fácil navegación para proyectos y código.
- ✚ Mantiene el código bajo control de chequeos, asistencia de pruebas, refactorizaciones y un conjunto de inspecciones que posibilitan codificar de forma limpia y sostenible.

1.4.8. Servidor web

Se utilizará el Servidor Web Apache en su versión 2.2 ya que es un servidor web de software libre desarrollado por *Apache Software Foundation* (ASF). Desde 1996 es el servidor más utilizado en Internet y es el utilizado en sistemas GNU/Linux. Las ventajas que brinda Apache son:

- ✚ Es robusto y con un ciclo de desarrollo muy rápido gracias a la gran cantidad de colaboradores con los que dispone.

³ Entorno integrado de desarrollo

- ✚ Es estable (impide caídas o cambios en el servidor inesperados), flexible y eficiente (es capaz de trabajar con el estándar HTTP P/1.1 (RFC2616) y con la mayor parte de las extensiones web que existen en la actualidad: PHP, SSL, Python...), extensible (dispone de una elevada cantidad de módulos que amplían su funcionalidad) y multiplataforma (está disponible para diferentes plataformas como GNU/Linux, Windows, MacOS).

1.4.9. Herramientas para la realización de pruebas

Se utilizará JMeter en su versión 3.1 pues es un proyecto de Apache que puede ser utilizado como una herramienta de prueba de carga para analizar y medir el desempeño de una variedad de servicios, con énfasis en aplicaciones web. JMeter puede ser usado como una herramienta de pruebas unitarias para conexiones de bases de datos con JDBC, FTP, LDAP, Servicios web, JMS, HTTP y conexiones TCP genéricas. JMeter puede también ser configurado como un monitor, aunque es comúnmente considerado una solución ad-hoc respecto de soluciones avanzadas de monitoreo. Mientras que JMeter es clasificado como una herramienta de "generación de carga", no es una descripción completa de la herramienta. JMeter soporta aserciones para asegurarse que los datos recibidos son correctos, por cookies de hilos, configuración de variables y una variedad de reportes (15).

1.5. Conclusiones parciales del capítulo

Se realizó un estudio de las soluciones internacionales existentes en el mundo, donde se substancia el gran auge de las tecnologías en la consumación de sistemas de gestión Comercial donde se manifiesta la gestión/contratación de servicios. Aunque estas herramientas presentan soluciones que ameritan ser usadas en la presente investigación, se necesita de la implementación de un nuevo sistema acorde a las necesidades del negocio. Dichas herramientas no satisfacen las exigencias del cliente pues no definen en su totalidad el negocio presentado por ASIC.

Una vez vistas las características, ventajas y desventajas de las herramientas y tecnologías a utilizar, el equipo de desarrollo y el centro se decidió usar: AUP en su versión UCI como metodología de desarrollo ya que la entidad se encuentra enmarcada en un cambio de tecnologías y metodologías al ostentar la

acreditación en el grado de CMMI⁴ 2. También se propuso un paquete de herramientas para dar solución al problema antes mencionado.

⁴ *Capability Maturity Model Integration.*

CAPÍTULO 2. ANÁLISIS Y DISEÑO DEL MÓDULO DE GESTIÓN COMERCIAL PARA EL SISTEMA DE ASESORÍA DE ASIC

En el presente capítulo se describe la propuesta del módulo a desarrollar, tomando como punto inicial los elementos trazados por la metodología AUP-UCI. Se dan a conocer las primordiales funcionalidades de la propuesta de solución mediante los requisitos funcionales, no funcionales y las historias de usuarios. Se revelan arquetipos de interfaz de usuario, así como los estándares de diseño que se utilizarán durante la implementación.

2.1. Propuesta de solución

En la introducción se planteó las dificultades que se presentaban en la empresa ASIC. Tras la presentación de dichas no conformidades se propone la solución de la realización de un software informático, un módulo web, donde se propone la presentación del modelo de gestión Comercial para la contratación de servicio y recursos humanos, el cual está formado por procesos de gestión de contratación y facturación de servicios, y otro de gestión y contratación de recursos humanos; a su vez este módulo va a poder ser visto tanto forma local como nacional. En el primer proceso se gestionarán todos los procesos de gestión de contratos, clientes, servicios, facturas, ordenes de trabajos, proveedores y el nivel de satisfacción de los clientes; en el segundo se realizará una gestión de los medios y recursos humanos de dicha empresa.

Se contará en la aplicación con un *login* o página de autenticación donde se podrán autenticar los usuarios previamente autorizados por el administrador de la plataforma web. Ya autenticado el sistema mostrará una vista de inicio donde se encontrarán disímiles datos de importancia para el negocio. En la parte izquierda de la aplicación se encontrará el menú de navegación de las funcionalidades del sistema donde se podrán encontrar las operaciones que se realizan en la empresa ASIC, tales como: gestión de clientes, proveedores, contratos, servicios, recursos humanos, ordenes de trabajo, facturas, cuentas, usuarios, registros de satisfacción; los mismos pueden ser accedidos en correspondencia de los permisos que tenga el usuario previamente autenticado.

Se encontrará en la parte superior un menú del usuario donde podrá acceder a sus posibles configuraciones; se mostrará su nombre completo, su avatar y otro menú desplegable donde se encontrarán las opciones de perfil, bloqueo y salir. En perfil tendrá la opción de ver sus datos generales, cambiar tanto como avatar y

contraseña. La opción de bloqueo te permitirá que no trabajen sobre tu cuenta de trabajo, y por último la opción de salir que no es más que el cierre de la sección del usuario previamente conectado.

2.2. Análisis

Es una tarea de la ingeniería del software que cubre el hueco entre la definición del software a nivel sistema y el diseño del software. El análisis de requisitos permite refinar la definición del software y construir los modelos de los dominios de datos, funcional y de comportamiento a ser considerados en el producto software. El análisis de requisitos proporciona al diseñador del software los siguientes modelos: de datos, arquitectónico, de interfaz y procedimental. La especificación de requisitos proporciona al diseñador y al cliente los medios para valorar la calidad una vez que se ha construido el software (12).

2.2.1. Modelo conceptual o diagrama del dominio

Un modelo del dominio es una representación visual de las clases conceptuales u objetos del mundo real en un dominio de interés. También se les denomina modelos conceptuales, modelo de objetos del dominio y modelos de objetos de análisis⁵ (17).

⁵ También están relacionados con los modelos conceptuales entidad-relación, que son capaces de mostrar vistas de los dominios puramente conceptuales, pero que han sido ampliamente re-interpretados como modelos de datos para el diseño de bases de datos. Los modelos del dominio no son modelos de datos.

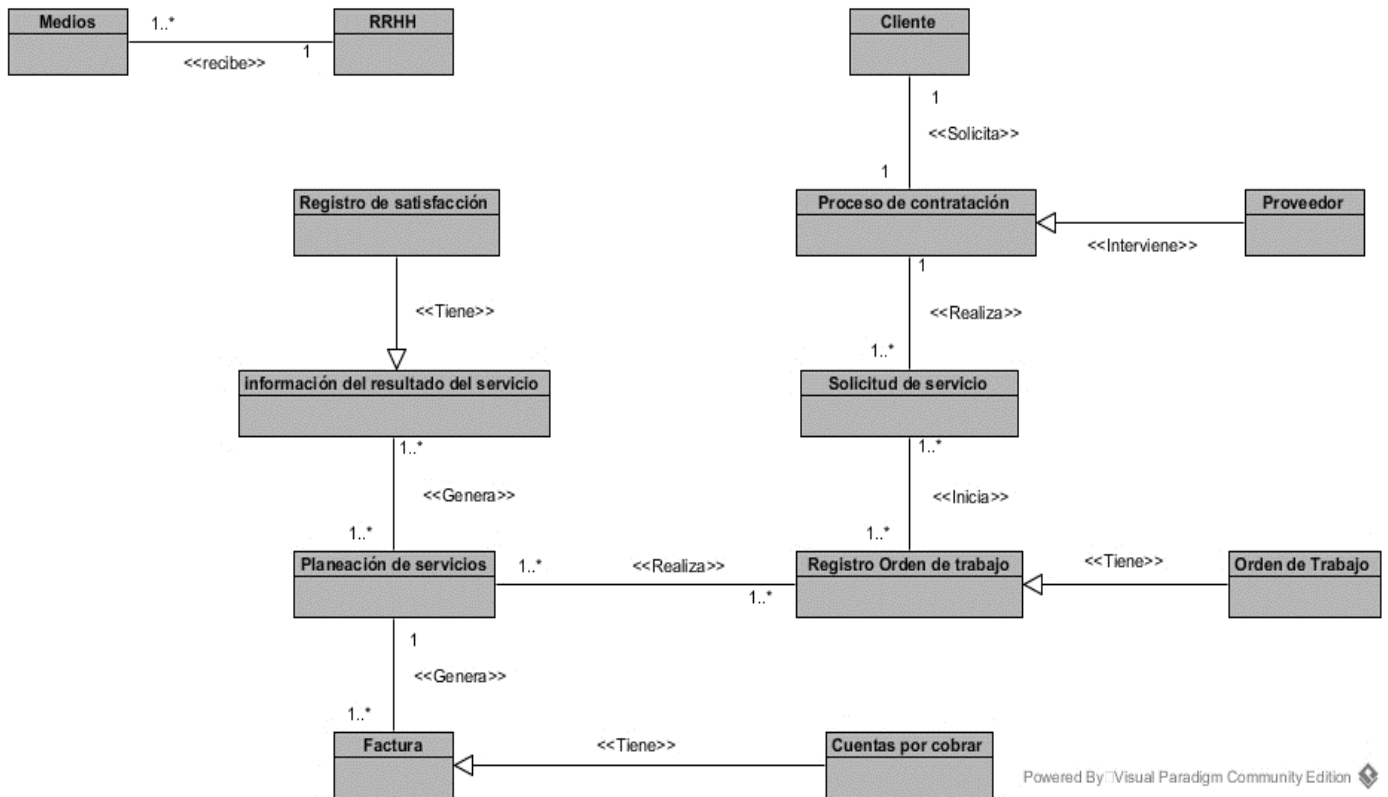


Figura 1: Modelo conceptual o diagrama del dominio. (Elaboración propia)

2.2.2. Requisitos de la propuesta solución

Los requisitos son capacidades y condiciones con las cuales debe ser conforme el sistema -y más ampliamente, el proyecto. El primer reto del trabajo de los requisitos es encontrar, comunicar y recordar (que normalmente significa registrar) lo que se necesita realmente, de manera que tenga un significado claro para el cliente y los miembros del equipo de desarrollo (17).

Requisitos funcionales

Tabla 1: Requisitos funcionales. (Elaboración propia)

Nº	Nombre	Descripción	Prioridad para el cliente	Complejidad
RF1	Insertar Cliente.	Permitirá insertar un cliente en	Alta	Media

		el sistema con sus datos correspondientes.		
RF2	Modificar Cliente.	Permitirá modificar un cliente en el sistema.	Alta	Media
RF3	Eliminar Cliente.	Permitirá eliminar un cliente en el sistema.	Alta	Media
RF4	Visualizar Cliente.	Permitirá visualizar un cliente en el sistema.	Alta	Media
RF5	Insertar Proveedor.	Permitirá insertar un proveedor en el sistema con sus datos correspondientes.	Media	Media
RF6	Modificar proveedor.	Permitirá modificar un proveedor en el sistema.	Media	Media
RF7	Eliminar proveedor.	Permitirá eliminar un proveedor en el sistema.	Media	Media
RF8	Visualizar proveedor.	Permitirá visualizar un proveedor en el sistema.	Media	Media
RF9	Insertar Contrato.	Permitirá insertar un contrato con los diferentes clientes.	Alta	Media
RF10	Modificar Contrato.	Permitirá modificar un contrato.	Alta	Media
RF11	Eliminar Contrato.	Permitirá eliminar un contrato.	Alta	Media
RF12	Visualizar Contrato.	Permitirá visualizar un contrato.	Alta	Media
RF13	Insertar solicitud de servicio.	Permitirá insertar solicitudes de servicios de los diferentes clientes.	Alta	Baja
RF14	Modificar solicitud de	Permitirá modificar solicitudes	Alta	Media

	servicio.	de servicios de los diferentes clientes.		
RF15	Eliminar solicitud de servicio.	Permitirá eliminar solicitudes de servicios de los diferentes clientes.	Alta	Media
RF16	Visualizar solicitud de servicio.	Permitirá visualizar solicitudes de servicios de los diferentes clientes.	Alta	Baja
RF17	Insertar orden de trabajo.	Permitirá insertar las diferentes órdenes de trabajo de los servicios que se brinden en la ASIC.	Alta	Baja
RF18	Modificar orden de trabajo	Permitirá modificar las diferentes órdenes de trabajo de los servicios que se brinden en la ASIC.	Alta	Media
RF19	Eliminar orden de trabajo.	Permitirá eliminar las diferentes órdenes de trabajo de los servicios que se brinden en la ASIC.	Alta	Media
RF20	Visualizar orden de trabajo.	Permitirá visualizar las diferentes órdenes de trabajo de los servicios que se brinden en la ASIC.	Alta	Baja
RF21	Insertar factura.	Permitirá insertar las diferentes facturas que se emitan en la ASIC.	Media	Baja
RF22	Modificar factura.	Permitirá modificar las diferentes facturas que se	Media	Media

		emitan en la ASIC.		
RF23	Eliminar factura.	Permitirá eliminar las diferentes facturas que se emitan en la ASIC.	Media	Media
RF24	Visualizar factura.	Permitirá visualizar los diferentes las diferentes facturas que se emitan en la ASIC.	Media	Baja
RF25	Insertar cuentas por cobrar.	Permitirá insertar las diferentes cuentas por cobrar.	Media	Baja
RF26	Modificar cuentas por cobrar.	Permitirá modificar las diferentes cuentas por cobrar.	Media	Media
RF27	Eliminar cuentas por cobrar.	Permitirá eliminar las diferentes cuentas por cobrar.	Media	Media
RF28	Visualizar cuentas por cobrar.	Permitirá visualizar las diferentes cuentas por cobrar.	Media	Baja
RF29	Insertar recursos humanos.	Permitirá insertar los recursos humanos de la ASIC.	Alta	Baja
RF30	Modificar recursos humanos.	Permitirá modificar los recursos humanos de la ASIC.	Alta	Media
RF31	Eliminar recursos humanos.	Permitirá eliminar los recursos humanos de la ASIC.	Alta	Media
RF32	Visualizar recursos humanos.	Permitirá visualizar los recursos humanos de la ASIC.	Alta	Baja
RF33	Insertar medios.	Permitirá insertar los diferentes medios que son asignados a los trabajadores.	Media	Baja
RF34	Modificar medios.	Permitirá modificar los diferentes medios que son	Media	Media

		asignados a los trabajadores.		
RF35	Eliminar medios.	Permitirá eliminar los diferentes medios que son asignados a los trabajadores.	Media	Media
RF36	Visualizar medios.	Permitirá visualizar los diferentes medios que son asignados a los trabajadores.	Media	Baja
RF37	Insertar registro de satisfacción del cliente.	Permitirá insertar las reclamaciones, quejas y sugerencias recibidas de los clientes.	Media	Baja
RF38	Modificar registro de satisfacción del cliente.	Permitirá modificar las reclamaciones, quejas y sugerencias recibidas de los clientes.	Media	Media
RF39	Eliminar registro de satisfacción del cliente.	Permitirá eliminar las reclamaciones, quejas y sugerencias recibidas de los clientes.	Media	Media
RF40	Visualizar registro de satisfacción del cliente.	Permitirá visualizar las reclamaciones, quejas y sugerencias recibidas de los clientes.	Media	Baja
RF41	Insertar usuario.	Permitirá insertar los diferentes usuarios en el sistema.	Media	Baja
RF42	Modificar usuario.	Permitirá modificar los diferentes usuarios en el sistema.	Media	Media
RF43	Eliminar usuario.	Permitirá eliminar los diferentes	Media	Baja

		usuarios en el sistema.		
RF44	Visualizar usuario.	Permitirá visualizar los diferentes usuarios en el sistema.	Media	Media
RF45	Cambiar contraseña.	Permitirá cambiar la contraseña de un usuario en el sistema.	Media	Media
RF46	Desactivar usuario.	Permitirá desactivar un usuario en el sistema.	Media	Media
RF47	Insertar rol.	Permitirá insertar los diferentes roles en el sistema.	Media	Baja
RF48	Modificar rol.	Permitirá modificar los diferentes roles en el sistema.	Media	Media
RF49	Eliminar rol.	Permitirá eliminar los diferentes roles en el sistema.	Media	Media
RF50	Visualizar rol.	Permitirá visualizar los diferentes roles en el sistema.	Media	Baja
RF51	Insertar nomenclador.	Se permitirá insertar cada nomenclador.	Media	Baja
RF52	Modificar nomenclador.	Se permitirá modificar cada nomenclador.	Media	Media
RF53	Eliminar nomenclador.	Se permitirá eliminar cada nomenclador.	Media	Media
RF54	Visualizar nomenclador.	Se permitirá visualizar cada nomenclador.	Media	Baja

Requisitos no funcionales

Confiabilidad

-  **RnF1.** Alertar al usuario ante cualquier error que ocurra en el sistema.

Usabilidad

- ✚ **RnF2.** El sistema va dirigido a los usuarios de la empresa ASIC, tenga sexo tanto masculino como femenino, cualquier nivel de escolaridad, cualquier experiencia en la utilización del sistema.
- ✚ **RnF3.** La finalidad de la aplicación es permitir a cualquier usuario realizar las tareas necesarias en el sistema, teniendo como meta fundamental satisfacer las necesidades del entorno empresarial.
- ✚ **RnF4.** La interfaz del sistema debe de estar acorde a la identidad de la empresa ASIC.
- ✚ **RnF5.** El sistema debe poseer ayuda para cada una de sus funcionalidades y manual de usuario.

Portabilidad

- ✚ **RnF6.** El sistema debe ser ejecutado correctamente en cualquier plataforma y navegador Web.

Para una mejor comprensión de los requisitos no funcionales ver el anexo 1.

2.3. Diseño

El diseño es lo que casi todo ingeniero quiere hacer. Es el lugar en el que las reglas de la creatividad –los requerimientos de las consideraciones técnicas- se unen para formular un producto o sistema. El diseño crea una representación o modelo del software, pero, a diferencia del modelo de los requerimientos (que se centra en describir los datos que se necesitan, la función y el comportamiento), el modelo de diseño proporciona detalles sobre la arquitectura del software, estructuras de datos, interfaces y componentes que se necesitan para implementar el sistema (18).

2.3.1. Diagrama de clases

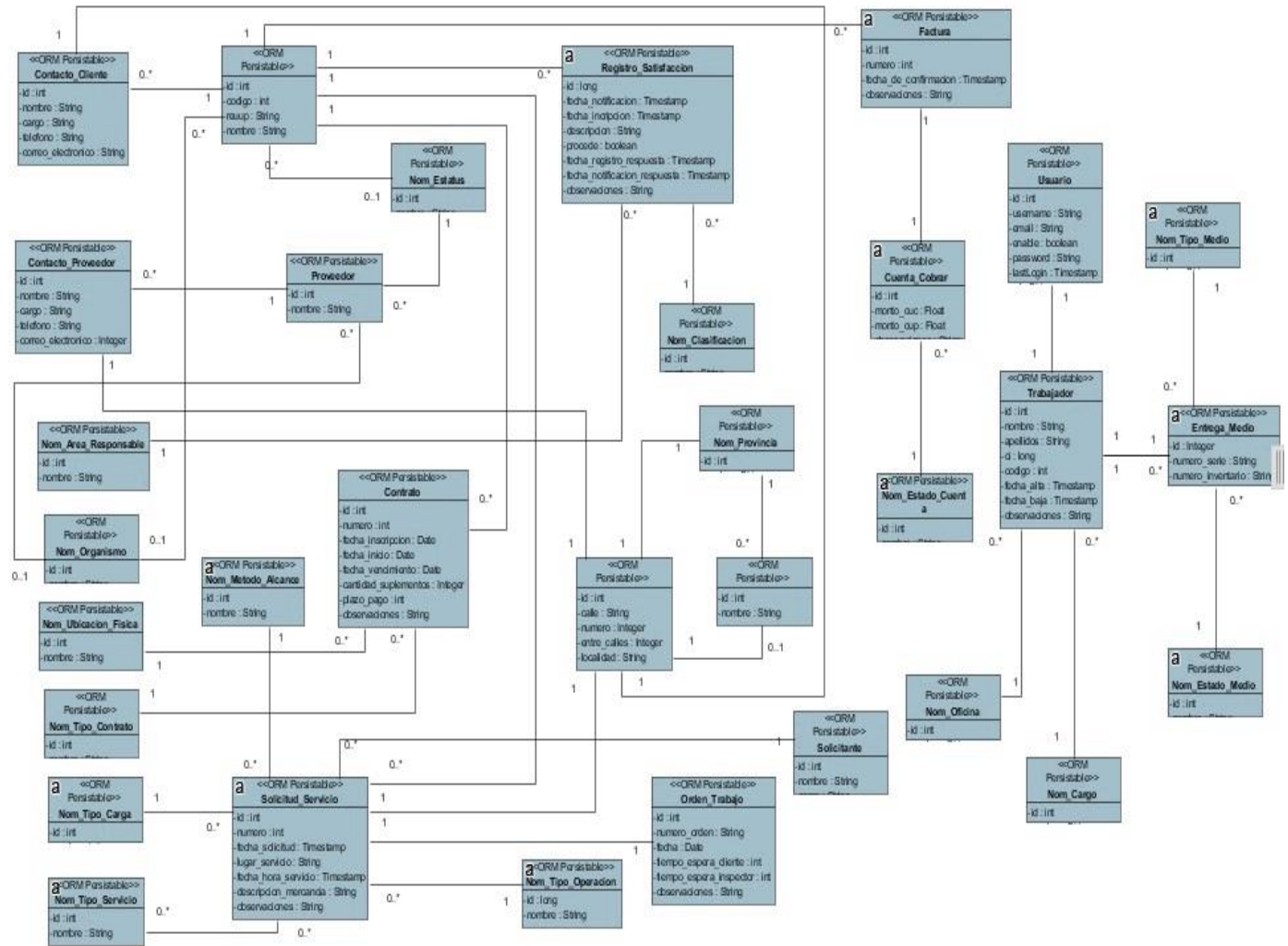


Figura 2: Diagrama de clases. (Elaboración propia)

2.3.2. Historias de Usuario

Una historia de usuario es una representación de un requisito escrito en una o dos frases utilizando el lenguaje común del usuario. Las historias de usuario son utilizadas en las metodologías de desarrollo ágiles para la especificación de requisitos (acompañadas de las discusiones con los usuarios y las pruebas de

validación). Cada historia de usuario debe ser limitada, ésta debería poderse escribir sobre una nota adhesiva pequeña. (19)

Las historias de usuario son una forma rápida de administrar los requisitos de los usuarios sin tener que elaborar gran cantidad de documentos formales y sin requerir de mucho tiempo para administrarlos. Las historias de usuario permiten responder rápidamente a los requisitos cambiantes. (19)

HU4 Gestionar solicitud de servicios

Tabla 2: HU4 Gestionar solicitud de servicios. (Elaboración propia)

Número: 04	Nombre del requisito: Insertar solicitud de servicio, Modificar solicitud de servicio, Eliminar solicitud de servicio, Visualizar solicitud de servicio.
Programador: Marbier Pérez Polo	Iteración Asignada: 1
Prioridad: Alta	
Riesgo en Desarrollo: Alta	
<p>Descripción: Permitirá insertar, modificar, eliminar y visualizar solicitudes de servicios de los diferentes clientes.</p> <p>De la solicitud se registrarán los siguientes datos:</p> <ul style="list-style-type: none"> • Número de la solicitud (Consecutivo numérico y puntos, de carácter obligatorio) • Fecha (Fecha en que se hace la solicitud, de carácter obligatorio) • Cliente (Valor alfanumérico y además guiones (-) de carácter obligatorio) • Solicitante (Valor de solo letras de carácter obligatorio) • Lugar del servicio (Valor alfanumérico de carácter obligatorio) • Dirección (Valor alfanumérico de carácter obligatorio) • Fecha y hora del servicio (Fecha y hora de carácter obligatorio) • Descripción de la mercancía (Valor alfanumérico de carácter obligatorio) • Operación (Es seleccionable y de carácter obligatorio) 	

- Tipo de carga (Es seleccionable y de carácter obligatorio)
- Servicio(s) solicitado(s) (Es seleccionable y de carácter obligatorio)
- Método y alcance de la inspección (Es seleccionable y de carácter obligatorio)
- Observaciones (Valor alfanumérico de carácter no obligatorio)

Observaciones:

El campo número está compuesto por el código de la provincia seguido un punto cliente seguido un punto consecutivo seguido un punto año: Ejemplo (01.03.01.2015)

Cuando seleccione el cliente deben salir las personas que están como personas de contacto de ese cliente para seleccionar el solicitante, y dar además la posibilidad de agregar a uno que no esté y en ese caso se debe agregar al sistema como persona de contacto.

El tipo de Operación puede ser:

- Importación
- Exportación
- Tráfico nacional
- Economía interna

Los tipos de Cargas pueden ser:

- Contenerizada
- Convencional
- Granel
- No convencional

Los Servicios Solicitados pueden ser:

- Servicio de iguala
- Vaciado de contenedores completos (FCL/FCL)
- Llenado de contenedores completos (FCL/FCL)
- Agrupe de carga Contenerizada (LCL/LCL, LCL/FCL, FCL/LCL)
- Desagrupe de carga Contenerizada (LCL/LCL, LCL/FCL, FCL/LCL)

- Expedición de carga convencional
- Recepción de carga convencional
- Agrupe de carga convencional
- Desagrupe de carga convencional
- Expedición de carga a granel
- Recepción de carga a granel.
- Inspección a equipos y vehículos
- Inspección a condiciones generales del contenedor (INTERCHANGE)
- Inspección a Inventario
- Inspección a mercancías en depósito en almacén
- Inspección a mercancías en tránsito en medio de transporte
- Certificación de pesaje de cargas

FCL es igual a uno, LCL es igual a muchos

Se pueden solicitar más de un servicio a la vez

Método y Alcance de la Inspección puede ser:

- **Por cantidad**
- **Por peso**
- **Por muestreo**
- **Detallado**
- **Exhaustivo**

Prototipo de interfaz:

Gestionar Solicitud de Servicio

Número de Solicitud Fecha de Solicitud Código Nombre Cliente

Lugar del Servicio Fecha y Hora del Servicio Dirección

Solicitante

Persona de contacto

Nombre	Cargo	Teléfono	Dirección	Correo Electrónico
<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>

Operación Tipo de Carga Descripción de la Mercancía

Servicio Solicitado Método y Alcance de la Inspección

Observaciones

Aceptar Cancelar

HU8 Gestionar RRHH

Tabla 3: HU8 Gestionar RRHH. (Elaboración propia)

Número: 08	Nombre del requisito: Insertar recursos humanos, Modificar recursos humanos, Eliminar recursos humanos, Visualizar recursos humanos.
-------------------	---

Programador: Marbier Pérez Polo	Iteración Asignada: 1
Prioridad: Alta	
Riesgo en Desarrollo: Baja	
Descripción: Permitirá insertar, modificar, eliminar y visualizar los recursos humanos de la ASIC. <ul style="list-style-type: none">• Oficina (Se refiere a la oficina a la que pertenece el trabajador, es seleccionable y de carácter obligatorio)• Nombre y Apellidos (Valor alfanumérico, de carácter obligatorio)• Carnet de Identidad (Valor numérico, de carácter obligatorio)• Cargo (Es seleccionable y de carácter obligatorio)• Código (Valor numérico, de carácter obligatorio)• Fecha alta (Fecha, de carácter obligatorio)• Fecha baja (Fecha, no obligatorio)• Observaciones (Valor alfanumérico, no obligatorio)	
Observaciones: Las oficinas son: <ul style="list-style-type: none">• Dirección ASIC• La Habana• Mariel• Pinar del Rio• Artemisa• Matanzas• Cienfuegos• Santi Spiritus• Villa Clara• Ciego de Ávila• Camagüey• Las Tunas	

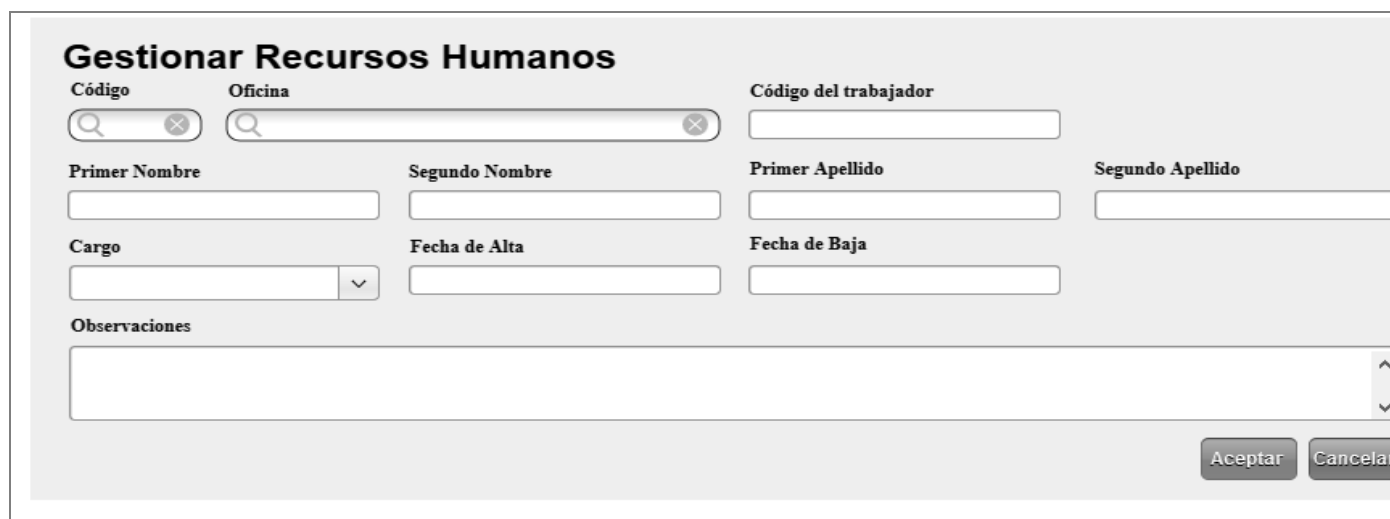
- Granma
- Holguín
- Moa
- Santiago de Cuba
- *INTER. INSPECTION AND SURVEYORS AGENCY INC (IISA INC) (CANADA)*
- *IISA INSPECTIONS PANAMA CORP (IISA INC) (PANAMA)*
- *SHENZHEN IISA INSPECTIONS SERVERS CO. (IISA INC) (CHINA)*

La codificación de los trabajadores es por oficina contará con 4 cifras que comienza con el código de la oficina seguido de punto y luego de un número consecutivo de no menos de 2 cifras, el consecutivo es por cada provincia.

Los Tipos de Cargos pueden ser:

- Director (UEB) Agencia de Supervisión e Inspección de Cargas
- Asesor C Jurídico
- Chofer D
- Especialista en Venta de Servicio Supervisión Comercial
- Técnico en Gestión Comercial
- Controlador Facturador
- Jefe de Grupo Técnico de Supervisión e Inspección de Cargas
- Especialista en Supervisión de Mercancías del Comercio Exterior
- Especialista en Cargas
- Técnico en Gestión Documental
- Tramitador de Documentos
- Jefe de Área de Trabajo de Supervisión e Inspección de Cargas de La Habana
- Jefe de Área de Trabajo de Supervisión e Inspección de Cargas del Mariel
- Jefe de Área de Trabajo de Supervisión e Inspección de Cargas de otras provincias
- Inspector Cargas (EP)
- Inspector Cargas

Prototipo de interfaz:



Gestionar Recursos Humanos

Código Oficina Código del trabajador

Primer Nombre Segundo Nombre Primer Apellido Segundo Apellido

Cargo Fecha de Alta Fecha de Baja

Observaciones

Aceptar Cancelar

Para ver las restantes historias de usuarios ver el anexo 2.

2.3.3. Arquitectura de software(AS)

El AS es la organización fundamental de un sistema encarnada en sus componentes, las relaciones entre ellos, el ambiente y los principios que orientan su diseño y evolución (20).

Patrón Modelo-Vista-Plantilla

Django se suele llamar un *framework* MVC⁶, y con razón. Es influenciado por MVC y es incluso posible argumentar que la terminología MVC es el único lugar patrón de cambios en Django. En Django, las tres capas básicas son el modelo, la vista, y la plantilla (21).

Para entender MTV se debe fijar en la analogía con MVC:

- ✚ El modelo en Django sigue siendo modelo(*Model*).
- ✚ La vista en Django se llama Plantilla (*Template*).
- ✚ El controlador en Django se llama Vista(*View*).

La siguiente imagen les ayudará a entender mejor esta relación: (22)

⁶ MVC: Modelo Vista Controlador

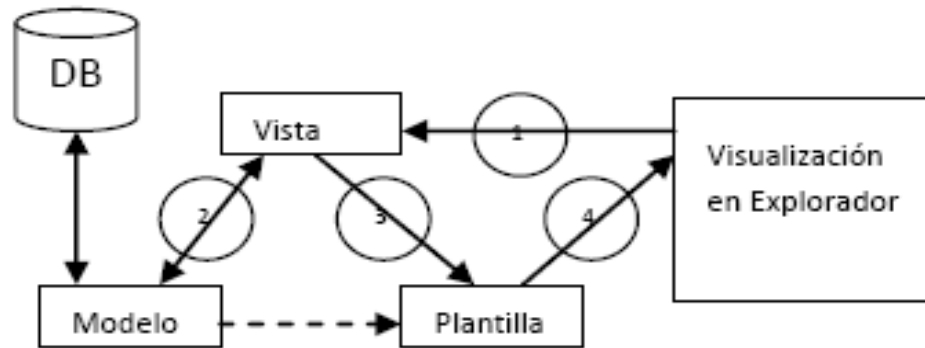


Figura 3: Arquitectura Modelo-Vista-Plantilla. (22)

- 1-El Navegador manda una solicitud
- 2-La vista interactúa con el modelo para obtener datos.
- 3-La vista llama a la plantilla.
- 4-La plantilla renderiza la respuesta a la solicitud del navegador

A continuación, realizaremos una breve explicación de lo antes mencionado con más detalle y algunos conceptos adicionales: (22)

El modelo: Se define los datos almacenados, se encuentra en forma de clases de Python, cada tipo de dato que debe ser almacenado se encuentra en una variable con ciertos parámetros, posee métodos también. Todo esto permite indicar y controlar el comportamiento de los datos.

La vista: Se presenta en forma de funciones en Python, su propósito es determinar qué datos serán visualizados, entre otras cosas más. El ORM de Django permite escribir código Python en lugar de SQL para hacer las consultas que necesita la vista. Lo más importante a entender con respecto a la vista es que no tiene nada que ver con el estilo de presentación de los datos, sólo se encarga de los datos, la presentación es tarea de la plantilla.

La plantilla: Se define básicamente como una página HTML con algunas etiquetas extras propias de Django, en sí no solamente crea contenido en HTML (también XML, CSS, JavaScript, CSV, etc.).

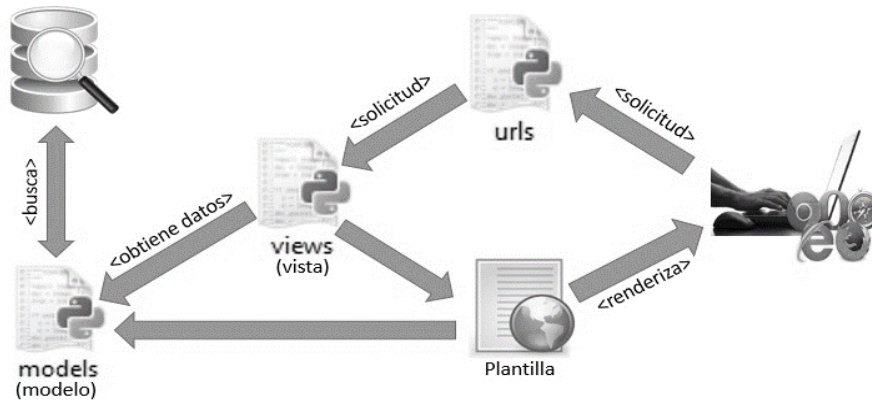


Figura 4: Arquitectura Modelo-Vista-Plantilla del software. (Elaboración propia)

El uso de dicha arquitectura se evidencia en su forma de estructuración de los componentes, como se distribuye el trabajo, es fácil manejo de las partes y ficheros que componen el proyecto a realizar. El mismo nos facilita una mejor comprensión del trabajo que se realiza y a su vez nos ayuda con una organización más fácil y adaptable a la hora de realizar un software o sistema web. Esto se ve evidenciado en la solución al observar la distribución de la estructura de trabajo de la propuesta al implementar el módulo Sistema_Gestión. En la siguiente imagen se puede distinguir la estructura compuesta por dicha arquitectura:

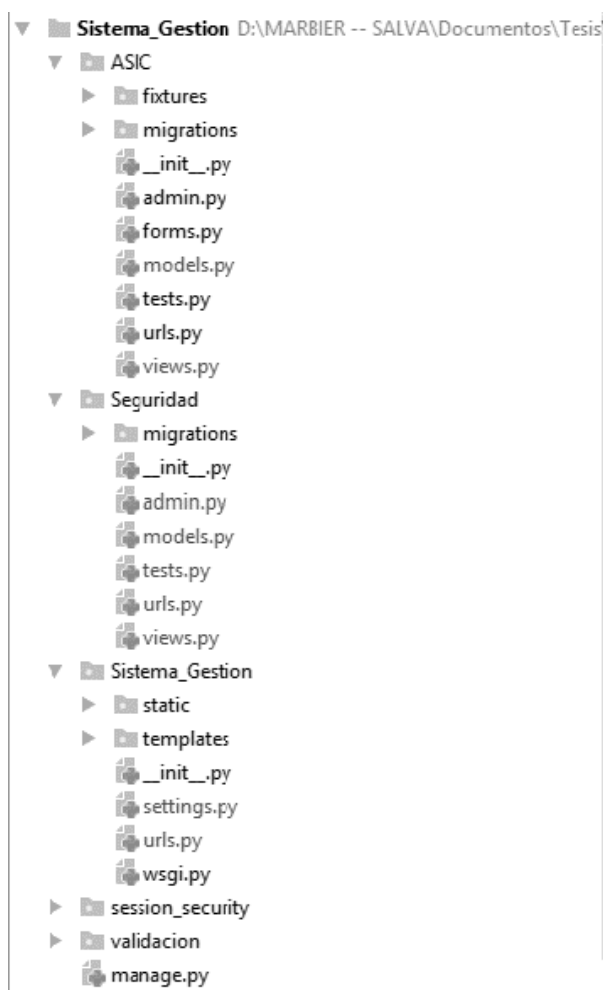


Figura 5: Estructura de carpetas del módulo. (Elaboración propia)

En la estructura de directorios anterior se pudo observar cómo se encuentran distribuidos los elementos que conforman la arquitectura MTV, se ve el fichero *urls.py*, este archivo es el encargado de hacer la petición a la vista correspondiente para cada URL que interviene en el sistema, también se observa el fichero *view.py* que es el encargado de dar repuestas a la petición entrada por la URL, además apreciamos el archivo *modelo.py* que es donde se definen todas las clases de la base de datos solución y por último y no menos importante la carpeta *templete* que es donde se encuentran todas las plantillas que le permiten la renderización de las respuestas enviadas por la vista.

Un ejemplo del funcionamiento de esta arquitectura al momento de la ejecución de la misma sería la autenticación de un usuario:

1-El usuario hace una solicitud a través de la url en el navegador a la vista.

```
url(r'^$', 'Seguridad.views.autenticar'),
```



0

Figura 6: Vista de la página Autenticar. (Elaboración propia)

2-La vista(controlador) solicitada la obtención de los datos al modelo, cual hace la petición a la base de datos.

```
def autenticar(request):  
    hay_error = False  
    datos = {}  
    if request.user.is_authenticated():  
        return HttpResponseRedirect('/inicio')  
    if request.method == 'POST':  
        usuario = request.POST['username']  
        contraseña = request.POST['password']  
        resultado = authenticate(username=usuario, password=contrasenna)
```

Figura 7: Obtención de los datos para verificación. (Elaboración propia)

3-Despues de haber obtenido los datos del modelo la vista hace una solicitud a la plantilla(vista).

```
def autenticar(request):  
    hay_error = False  
    datos = {}  
    if request.user.is_authenticated():  
        return HttpResponseRedirect('/inicio')
```

Figura 8: Verificación de autenticación. (Elaboración propia)

4-Y por último la plantilla renderiza la respuesta a la solicitud.

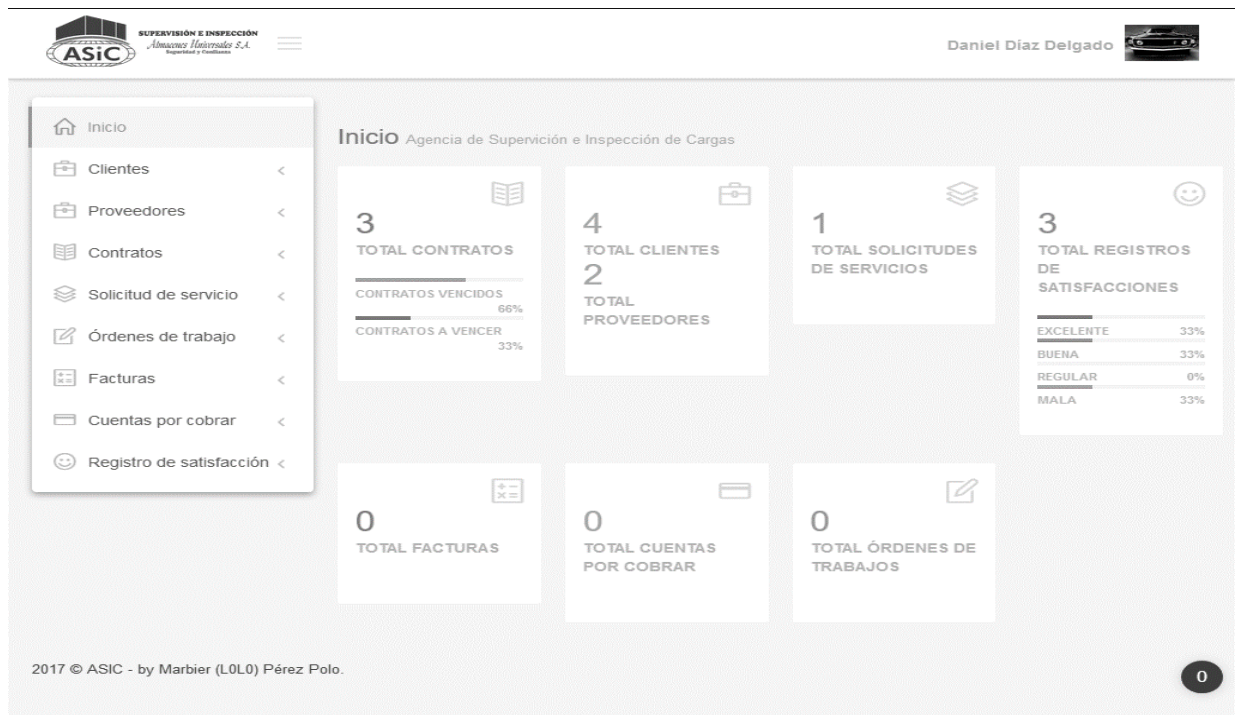


Figura 9: Vista de la página de inicio. (Elaboración propia)

2.3.4. Patrones de diseño

Brad Appleton define un patrón de diseño de la manera siguiente: “Es una mezcla con nombre propio de un punto de vista que contiene la esencia de una solución demostrada para un problema recurrente dentro de cierto contexto de necesidades en competencia” (23).

Patrones GRASP (General Responsibility Assignment Software Patterns⁷)

Se describen los principios fundamentales del diseño de objetos y la asignación de responsabilidades, expresados como patrones. (17)

- ✚ Experto en Información (o Experto): Permite la asignación de responsabilidades tales como la creación o inserción de datos, la eliminación y modificación de los mismos; y métodos tales que ayuden al funcionamiento de los mismos. (17)

```
def autenticar(request):
    hay_error = False
    datos = {}
    if request.user.is_authenticated():
        return HttpResponseRedirect('/inicio')
    if request.method == 'POST':
        usuario = request.POST['username']
        contraseña = request.POST['password']
        resultado = authenticate(username=usuario, password=contraseña)
        if resultado is None:
            messages.error(request, 'Usuario y/o contraseña incorrectos o su cuenta no existe.')
            hay_error = True
        elif not resultado.is_active:
            messages.error(request,
                'Su cuenta ha sido bloqueada, póngase en contacto con el administrador del sistema.')
            hay_error = True
        else:
            login(request, resultado)
            return HttpResponseRedirect('/inicio')
        if hay_error:
            datos['usuario'] = usuario
    return render_to_response('Seguridad/autenticar.html', {'datos': datos}, RequestContext(request))
```

Figura 10: Ejemplo del uso del Patrón Experto. (Elaboración propia)

- ✚ Creador: Se utiliza a la hora de la inserción o modificación de los datos al añadir los elementos pasados por los formularios correspondientes. Además de permitir la creación de objetos y clases en la vistas y modelos. (17)

⁷ patrones generales de software para asignar responsabilidades

- ✚ Bajo Acoplamiento: Este patrón ya viene incluido con Django que permite un bajo acoplamiento entre las piezas, lo que evita las dependencias, por lo que por ejemplo a la hora de realizar cambios en las configuraciones de las URL⁸, en las BD, plantillas HTML, etc., basta solo con realizarlo una sola vez. (17)
- ✚ Alta cohesión: Asignar una responsabilidad de manera que la cohesión permanezca alta. (17)

```
@login_required(login_url='/')
def bloqueo(request):...

@login_required(login_url='/')
def inicio(request):...
```

Figura 11: Código del bloqueo e inicio en la vista. (Elaboración propia)

- ✚ Controlador: Se utiliza para el control de las funciones, es decir, todos los métodos y funciones que se encuentran en el fichero view.py son controladores ya que son funciones que establecen la conexión entre las plantillas, modelos y BD. Además, se encargan del funcionamiento de las demás partes de la aplicación. (17)

```
@login_required(login_url='/')
def salir(request):
    logout(request)
    request.session.clear()
    return HttpResponseRedirect('/')
```

Figura 12: Código del salir en la vista. (Elaboración propia)

Patrones GoF(Gang of Four⁹)

- ✚ Decorador: Se utilizan para la hora de la seguridad de las funciones que se encuentran en el fichero view.py, no permitiendo el uso incorrecto y flujo de datos en la aplicación.

⁸ Del inglés *Uniform Resource Locator*.

⁹ "pandilla de los cuatro"

```
@login_required(login_url='/')
@permission_required(perm=('auth.add_user', 'auth.change_user'), login_url='/error403')
def perfil_cuenta(request):...

@login_required(login_url='/')
@permission_required(perm='ASIC.change_nomencladores', login_url='/error403')
def lista_nomenclaturas(request):...
```

Figura 13: Vista del uso de decoradores. (Elaboración propia)

2.3.5. Diagrama de clases del diseño

Un Diagrama de Clases de Diseño muestra la especificación para las clases software de una aplicación. Incluye la siguiente información:

- ✚ Clases, asociaciones y atributos.
- ✚ Interfaces, con sus operaciones y constantes.
- ✚ Métodos.
- ✚ Navegabilidad.
- ✚ Dependencias.

A diferencia del Modelo Conceptual, un Diagrama de Clases de Diseño muestra definiciones de entidades software más que conceptos del mundo real. (11)

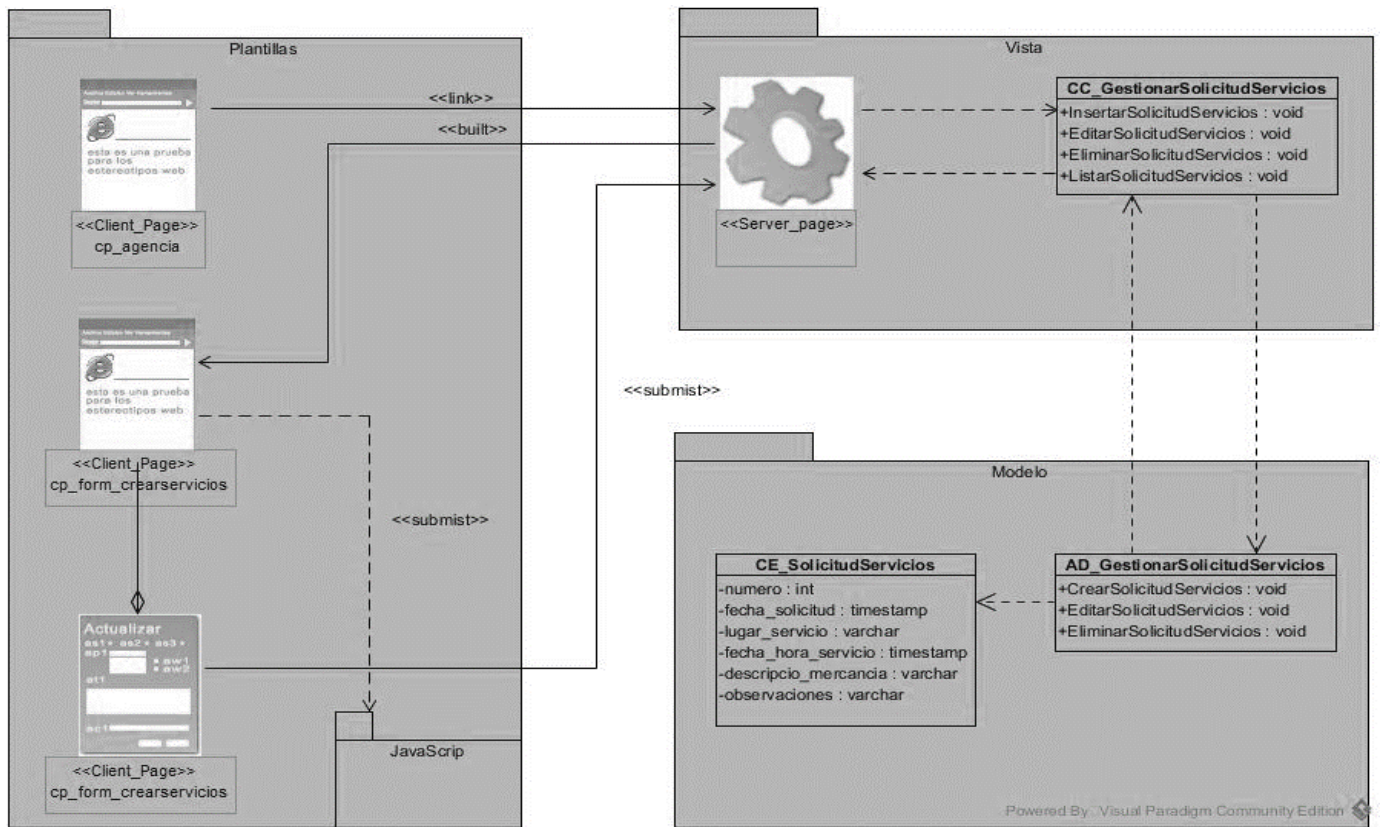


Figura 14: Diagrama de clases del diseño: Gestionar solicitud de Servicios. (Elaboración propia)

GestionarSolicitudServicio: va a ser la clase controlado de esa funcionalidad, en la misma se encontrarán los métodos:

- ❖ InsertarSolicitudServicio: Este permite la inserción de solicitudes de servicios.
- ❖ EditarSolicitudServicio: Este permite la edición de las solicitudes de servicios
- ❖ EliminarSolicitudServicio: Este permite la eliminación de las solicitudes de servicios
- ❖ ListarSolicitudServicio: Este permite mostrar una lista con todas las solicitudes de servicios realizadas.

2.3.6. Modelo de datos

Incluye los esquemas de bases de datos, y las estrategias de transformación entre representaciones de objetos y no objetuales.

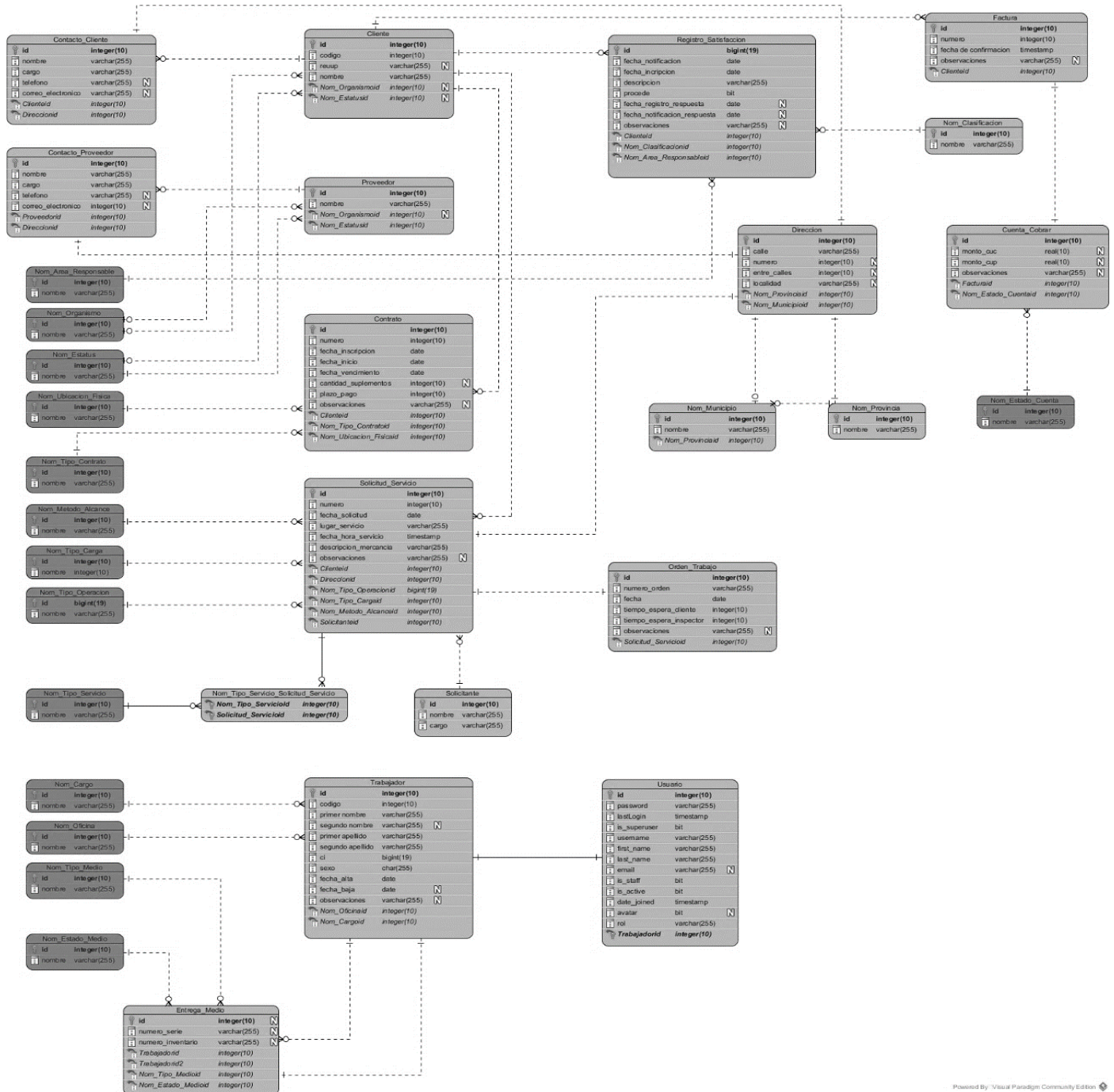


Figura 15: Modelo de datos. (Elaboración propia)

2.3.7. Diagrama de componente

Un diagrama de componentes representa cómo un sistema de software es dividido en componentes y muestra las dependencias entre estos componentes. Los componentes físicos incluyen archivos, cabeceras, bibliotecas compartidas, módulos, ejecutables, o paquetes. Los diagramas de Componentes prevalecen en el campo de la arquitectura de software, pero pueden ser usados para modelar y documentar cualquier arquitectura de sistema. (24)

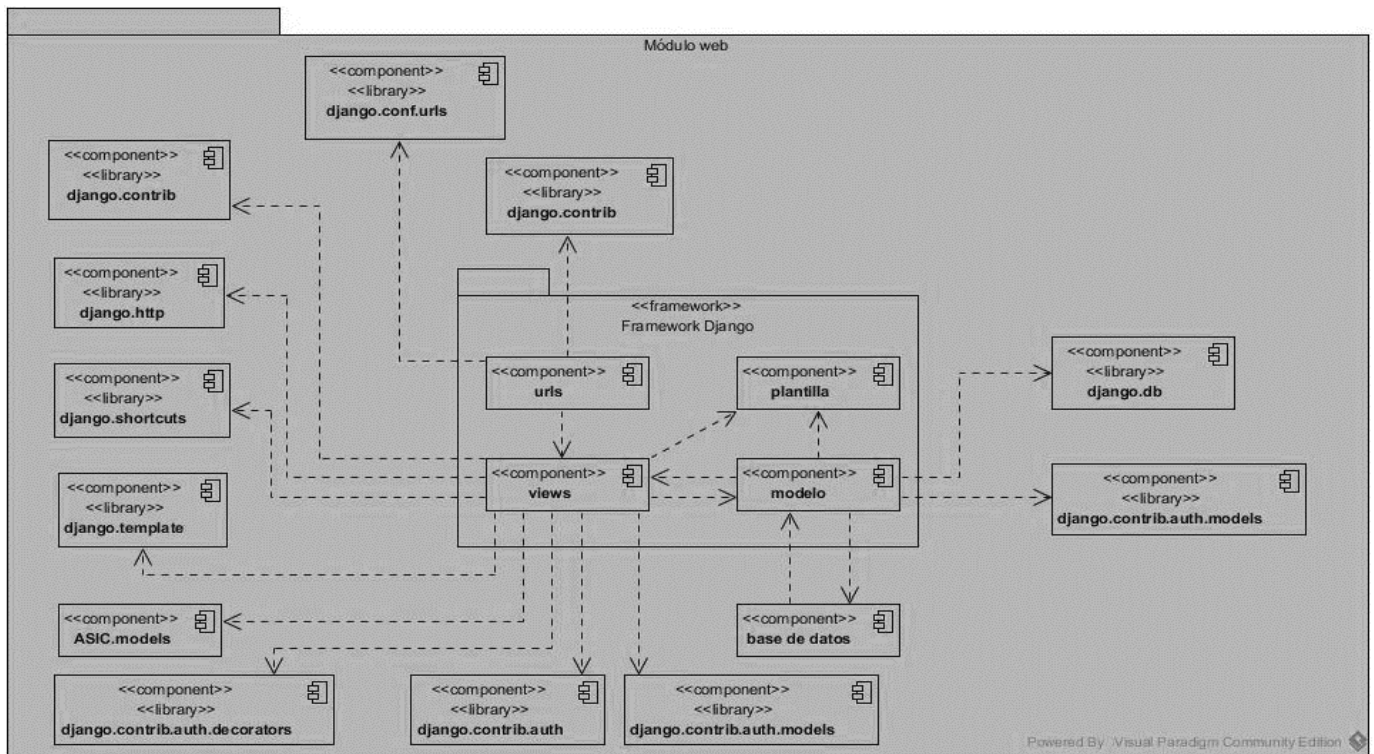


Figura 16: Diagrama de componente. (Elaboración propia)

2.3.8. Modelo o Diagrama de Despliegue

Un diagrama de despliegue muestra cómo se configuran las instancias de los componentes y los procesos para la ejecución run-time en las instancias de los nodos de proceso (algo con memoria y servicios de proceso). (25)



Figura 17: Modelo o Diagrama de Despliegue. (Elaboración propia)

2.4. Conclusiones parciales

Durante el transcurso del capítulo se realizó una descripción de la solución propuesta, se definieron y redactaron las historias de usuarios, así como las tareas de ingenierías correspondientes a cada una de ellas. Quedaron aprobados los requisitos funcionales necesarios para obtener un sistema eficiente. Resultaron claras las tareas que el sistema debe realizar. Se desarrolló satisfactoriamente su implementación a partir del diseño planteado, obteniéndose el módulo para la gestión de información en la Agencia de Supervisión e Inspección de Cargas.

CAPÍTULO 3. VALIDACIÓN DEL MÓDULO DE GESTIÓN COMERCIAL PARA EL SISTEMA DE ASESORÍA DE ASIC

El presente capítulo tiene como objetivo detallar la etapa de implementación y pruebas, fundando los artefactos pertenecientes a las mismas, como es el caso de los estándares de codificación y los casos de prueba. Posteriormente se describen los valores empleados en la ejecución de las pruebas y los resultados obtenidos.

3.1. Estándar de codificación (26)

Se utilizará el estándar de codificación ya establecido por el lenguaje de programación PEP 8.

3.1.1. Identación

- ✚ No usar los 4 espacios sino las tabulaciones, aunque algunos autores presumen de lo contrario en versiones más recientes
- ✚ Las líneas de continuación deben alinearse verticalmente con el carácter que se ha utilizado (paréntesis, llaves, corchetes) o haciendo uso de la “*hanging indent*”¹⁰. Al utilizar este último método, no debe haber argumentos en la primera línea.
- ✚ Limita todas las líneas a un máximo de 79 caracteres.
- ✚ Usar la barra invertida (“\”) para “cortar” líneas largas.

3.1.2. Líneas en blanco

- ✚ Separa funciones de alto nivel y definiciones de clase con dos líneas en blanco.
- ✚ Usa líneas en blanco en funciones, escasamente, para indicar secciones lógicas.
- ✚ Definiciones de métodos dentro de una clase son separadas por una línea en blanco.

3.1.3. Codificaciones

- ✚ El código en el núcleo de la distribución de Python siempre debería utilizar la codificación UTF-8, por lo que el utilizado en la solución es el mismo propuesto anteriormente.

¹⁰ Aplicar tabulaciones en todas las líneas con excepción de la primera.

- ✚ Se pueden incluir cadenas que no correspondan a esta codificación utilizando “\x”, “\u” o “\U”.

3.1.4. Importaciones

Las importaciones deben estar en líneas separadas.

- ✚ Las importaciones siempre se colocan al comienzo del archivo, simplemente luego de cualquier comentario o documentación del módulo, y antes de globales y constantes.
- ✚ Las importaciones deben estar agrupadas en el siguiente orden:
 1. importaciones de la librería estándar
 2. importaciones terceras relacionadas
 3. importaciones locales de la aplicación / librería
- ✚ Deberían poner una línea en blanco entre cada grupo.
- ✚ Coloca cualquier especificación `__all__` luego de las importaciones.

3.1.5. Espacios en blanco en Expresiones y Sentencias

- ✚ Evita usar espacios en blanco en las siguientes situaciones:
 - ❖ antes de una coma, un punto y coma o dos puntos
 - ❖ antes del paréntesis que comienza la lista de argumentos en la llamada a una función
 - ❖ antes de un corchete que empieza una indexación o “*slicing*”¹¹
 - ❖ Más de un espacio alrededor de un operador de asignación (u otro) para alinearlos con otro
- ✚ Deberían rodear estos operadores binarios con un espacio en cada lado:
 - ❖ asignación (=),
 - ❖ asignación de aumentación (+=, -=, etc.),
 - ❖ comparaciones (==, <, >, !=, <>, <=, >=, in, not in, is, is not),
 - ❖ “*Booleans*”¹² (and, or, not)

¹¹ término utilizado tanto en el ámbito de habla inglesa como española

¹² Buleanos

- ✚ No uses espacios alrededor del = (igual) cuando es utilizado para indicar un argumento en una función o un parámetro con un valor por defecto.

3.1.6. Comentarios

- ✚ Los comentarios deben ser oraciones completas. Si un comentario es una frase u oración, su primera palabra debe comenzar con mayúscula, a menos que sea un identificador que comienza con minúscula.
- ✚ Si un comentario es corto, el punto al final puede omitirse.
- ✚ Comentarios en bloque generalmente consisten en uno o más párrafos compuestos por oraciones completas, por lo que cada una de ellas debe finalizar en un punto.
- ✚ Deberías usar dos espacios luego de una oración que termine con un punto.

3.1.7. Comentarios en bloque

- ✚ Los comentarios en bloque generalmente se aplican a algunos (o todos) códigos que los siguen, y están indentados al mismo nivel que ese código. Cada línea de un comentario en bloque comienza con un # (numeral) y un espacio (a menos que esté indentado dentro del mismo comentario).
- ✚ Los párrafos dentro de un comentario en bloque están separados por una línea que contiene únicamente un # (numeral).

3.1.8. Comentarios en línea¹³

- ✚ Usa comentarios en línea escasamente.
- ✚ Los comentarios en línea deberían estar separados por al menos dos espacios de la sentencia. Deberían empezar con un # (numeral) seguido de un espacio.
- ✚ Utiliza los comentarios en línea solo si son necesarios.

3.1.9. Cadenas de documentación

- ✚ Deben quedar documentados todos los módulos, funciones, clases y métodos públicos.
- ✚ Para definir una cadena de documentación debe quedar encerrada dentro de ("").

¹³ comentarios en la misma línea

- ✚ Los (""") que finalizan una cadena de documentación deben quedar en una línea a no ser que la cadena sea de una sola línea.

3.1.10. Prescriptivo: Convenciones de nombramiento

Nombres para evitar:

- ✚ Nunca uses los caracteres 'l' (letra ele en minúscula), 'O' (letra o mayúscula), o 'I' (letra i mayúscula) como simples caracteres para nombres de variables.

Nombre de paquetes y módulos:

- ✚ Los módulos deben tener un nombre corto y en minúscula. Guiones bajos pueden utilizarse si mejora la legibilidad.
- ✚ Los paquetes en Python también deberían tener un nombre corto y en minúscula, aunque el uso de guiones bajos es desalentado (poco recomendado).

Nombre de clases:

- ✚ Los nombres de clases deben utilizar la convención "CapWords¹⁴". Clases para uso interno tienen un guion bajo como prefijo.

Nombre de excepciones:

- ✚ Deberías usar un sufijo "Error" en los nombres de excepciones (en caso que corresponda a un error).

Nombre de funciones:

- ✚ Las funciones deben ser en minúscula, con las palabras separadas por un guion bajo, aplicándose éstos tanto como sea necesario para mejorar la legibilidad.
- ✚ "mixedCase¹⁵" es aceptado únicamente en contextos en donde éste es el estilo predominante (por ejemplo, threading.py) con el objetivo de mantener la compatibilidad con versiones anteriores.

¹⁴ palabras que comienzan con mayúsculas.

¹⁵ primera palabra en minúscula.

Nombre de métodos y variables de instancia:

- ✚ Usa las mismas reglas que para el nombramiento de funciones: en minúscula con palabras separadas con guiones bajos, tantos como sea necesario para mejorar la legibilidad.
- ✚ Usa un solo guion bajo como prefijo para métodos no públicos y variables de instancia.

Constantes:

- ✚ Las constantes son generalmente definidas a nivel módulo, escritas con todas las letras en mayúscula y con guiones bajos separando palabras.

3.2. Prueba

La prueba es un conjunto de actividades que se planean con anticipación y se realizan de manera sistemática. Por tanto, se debe definir una plantilla para las pruebas de software (un conjunto de pasos en que se puedan incluir técnicas y métodos específicos del diseño de casos de pruebas). (14)

3.2.1. Pruebas del sistema

La fase de pruebas del sistema tiene como objetivo verificar el sistema software para comprobar si este cumple sus requisitos. Dentro de esta fase pueden desarrollarse varios tipos distintos de pruebas en función de los objetivos de las mismas. Algunos tipos son pruebas funcionales, pruebas de usabilidad, pruebas de rendimiento, pruebas de seguridad, etc. Este trabajo se centra en pruebas funcionales de aplicaciones con interfaces gráficas. Estas pruebas verifican que el sistema software ofrece a los actores humanos la funcionalidad recogida en su especificación. (18)

Ambiente de pruebas

Se realizó las pruebas en un ambiente de trabajo utilizando como host y server un cliente pc con las siguientes características:

- ✚ Core i5 a 3.20Gz con 4gb de RAM y 1tb de disco duro.
- ✚ Servidor Web: Apache v2.2.

3.2.2. Prueba de rendimiento

Las pruebas de rendimiento se usan para descubrir problemas de rendimiento que pueden ser resultado de: falta de recursos en el lado servidor, red con ancho de banda inadecuada, capacidades de base de

datos inadecuadas, capacidades de sistema operativo deficientes o débiles, funcionalidad de “webapp¹⁶” pobremente diseñada y otros conflictos de hardware o software que pueden conducir a rendimiento cliente-servidor degradado. La intención es doble: 1) comprender cómo responde el sistema conforme aumenta la carga (es decir, número de usuarios, número de transacciones o volumen de datos global) y 2) recopilar mediciones que conducirán a modificaciones de diseño para mejorar el rendimiento.

Prueba de carga:

La intención de la prueba de carga es determinar cómo responderán las “webapps” y su entorno del lado servidor a varias condiciones de carga. Conforme avanzan las pruebas, las permutas de las siguientes variables definen un conjunto de condiciones de prueba:

N, número de usuarios concurrentes

T, número de transacciones en línea por unidad de tiempo

D, carga de datos procesados por el servidor en cada transacción

En todo caso, dichas variables se definen dentro de fronteras operativas normales del sistema. Conforme se aplica cada condición de prueba, se recopila una o más de las siguientes medidas: respuesta de usuario promedio, tiempo promedio para descargar una unidad estandarizada de datos y tiempo promedio para procesar una transacción. Estas medidas deben examinarse para determinar si una disminución abrupta en el rendimiento puede rastrearse en una combinación específica de N, T y D. La prueba de carga también puede usarse para valorar las velocidades de conexión recomendadas para los usuarios de la “webapp”. El rendimiento global, P, se calcula de la forma siguiente:

$$P = N + T + D$$

Prueba de esfuerzo:

La prueba de esfuerzo es una continuación de la prueba de carga, pero en esta instancia las variables N, T y D se fuerzan a satisfacerse y luego se superan los límites operativos. La intención de estas pruebas es responder a cada una de las siguientes preguntas: (12)

¹⁶ Aplicación web

- ✚ ¿El sistema se degrada “suavemente” o el servidor se apaga conforme la capacidad se supera?
- ✚ ¿El software servidor genera mensajes “servidor no disponible”? De manera más general, ¿los usuarios están conscientes de que no pueden llegar al servidor?
- ✚ ¿El servidor pone en cola los recursos solicitados y vacía la cola una vez que disminuye la demanda de capacidad?
- ✚ ¿Las transacciones se pierden conforme la capacidad se excede?
- ✚ ¿La integridad de los datos resulta afectada conforme la capacidad se excede?
- ✚ ¿Qué valores de N, T y D fuerzan el fallo del entorno servidor? ¿Cómo se manifiesta la falla? ¿Se envían notificaciones automáticas al personal de apoyo técnico en el sitio servidor?
- ✚ Si el sistema falla, ¿cuánto tiempo tardará en regresar en línea?
- ✚ ¿Ciertas funciones de la webapp (por ejemplo, funcionalidad de cálculo intenso, capacidades de transmisión de datos) quedan descontinuadas conforme la capacidad alcanza el nivel de 80 o 90 por ciento?

Aplicación de la prueba con la herramienta apache jmeter en su versión 3.1:

La prueba se realizó utilizando muestra de hasta 50, 100, 200 y 1000; y el resultado fue promediado para su resultado final.

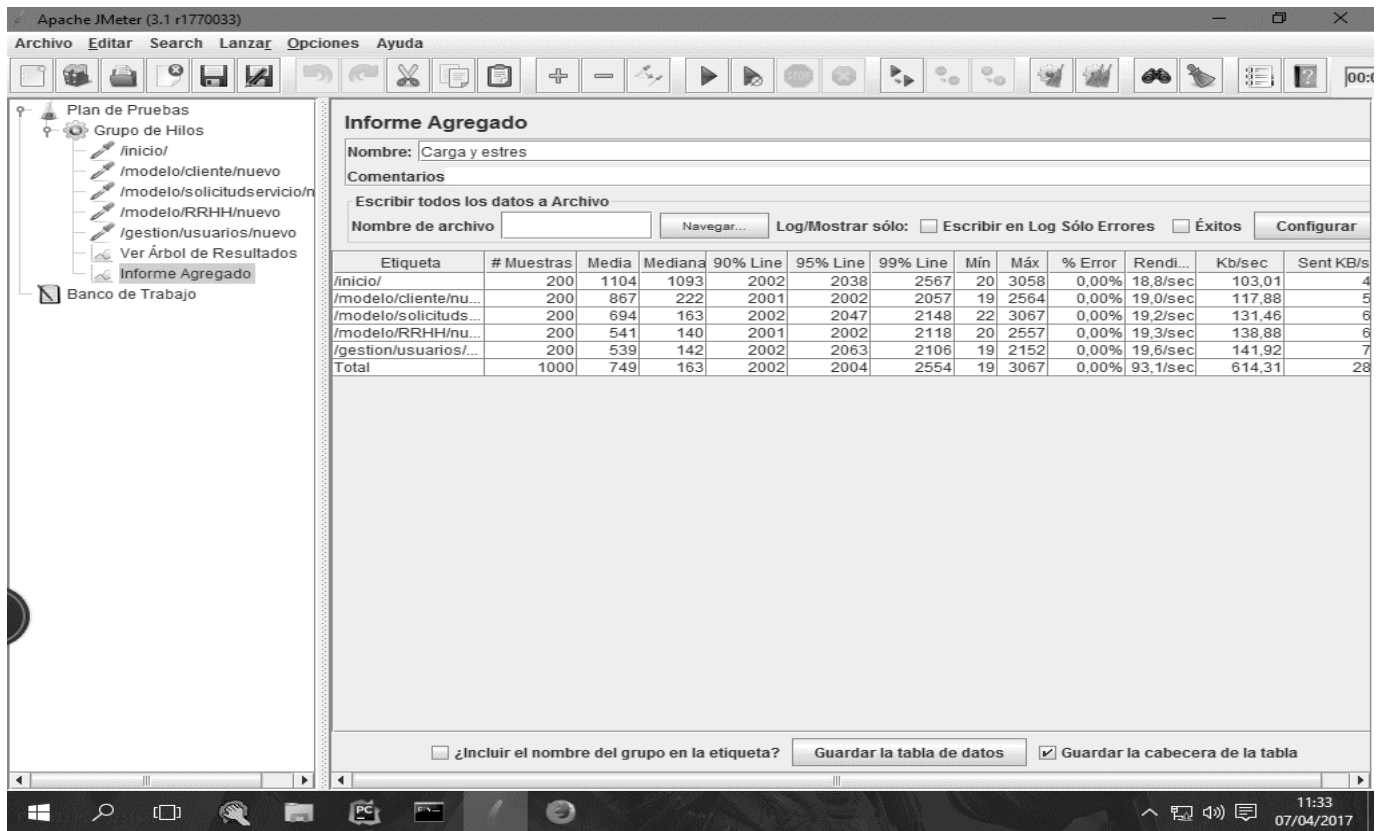


Figura 18: Interfaz del JMeter. (Elaboración propia)

Etiqueta	# Muestras	Media	Mediana	90% Line	95% Line	99% Line	Mín	Máx	% Error	Rendi...	Kb/sec	Sent KB/s
/inicio/	200	1104	1093	2002	2038	2567	20	3058	0,00%	18,8/sec	103,01	4
/modelo/cliente/nu...	200	867	222	2001	2002	2057	19	2564	0,00%	19,0/sec	117,88	5
/modelo/solicitud...	200	694	163	2002	2047	2148	22	3067	0,00%	19,2/sec	131,46	6
/modelo/RRHH/nu...	200	541	140	2001	2002	2118	20	2557	0,00%	19,3/sec	138,88	6
/gestion/usuarios/...	200	539	142	2002	2063	2106	19	2152	0,00%	19,6/sec	141,92	7
Total	1000	749	163	2002	2004	2554	19	3067	0,00%	93,1/sec	614,31	28

Figura 19: Resultado de la prueba de rendimiento. (Elaboración propia)

Resultados de las pruebas de carga y estrés

Con el objetivo de saber con un alto nivel de proximidad el soporte del módulo de gestión Comercial, se desarrollaron las pruebas para un total de 10 y 200 usuarios concurrentes.

Las pruebas se desarrollaron con el apoyo de la herramienta Apache Meter 3.1, en la que se simuló el entorno donde debe interactuar el modulo para obtener la información más correcta acerca del

comportamiento y resultados en general. Por lo que fue elegido un ambiente de gama con las siguientes características:

Software:

- ✚ Sistema Operativo: Windows 10 Arquitectura de 64bits
- ✚ Sistema Gestor de Bases de Datos: PostgreSQL 9.4
- ✚ Servidor Web: Apache v2.2

Hardware:

- ✚ Microprocesor: Intel Core(TM) i5 - 4460 CPU @ 3.20 GHz (4CPUs)
- ✚ Memoria RAM: 4GB
- ✚ Tarjeta de Red: Ethernet 10/100 Mbps

Estas pruebas arrojaron como resultado que se lograría un tiempo máx. de demora de la conexión de 2 a 3 segundos aproximadamente.

3.2.3. Pruebas de seguridad

Cualquier sistema de cómputo que maneje información confidencial o que desencadene acciones que dañen (o beneficien) inapropiadamente a los individuos es un blanco para irrupciones impropias o ilegales. La irrupción abarca un amplio rango de actividades: hackers que tratan de entrar en los sistemas por juego, empleados disgustados que tratan de irrumpir como forma de venganza, e individuos deshonestos que buscan ganancias personales ilícitas. La prueba de seguridad comprueba que los mecanismos de protección integrados en el sistema realmente lo protejan de irrupciones inapropiadas. (14)

Al módulo desarrollado se le realizaron una serie de pruebas de seguridad mediante la herramienta de detección de vulnerabilidades *Acunetix Web Vulnerability Scanner 9.5*, la cual arrojó un total de 21 vulnerabilidades, las cuales se clasifican en alta, baja, media e informativas. Se realizaron 3 iteraciones, proyectando los siguientes resultados:

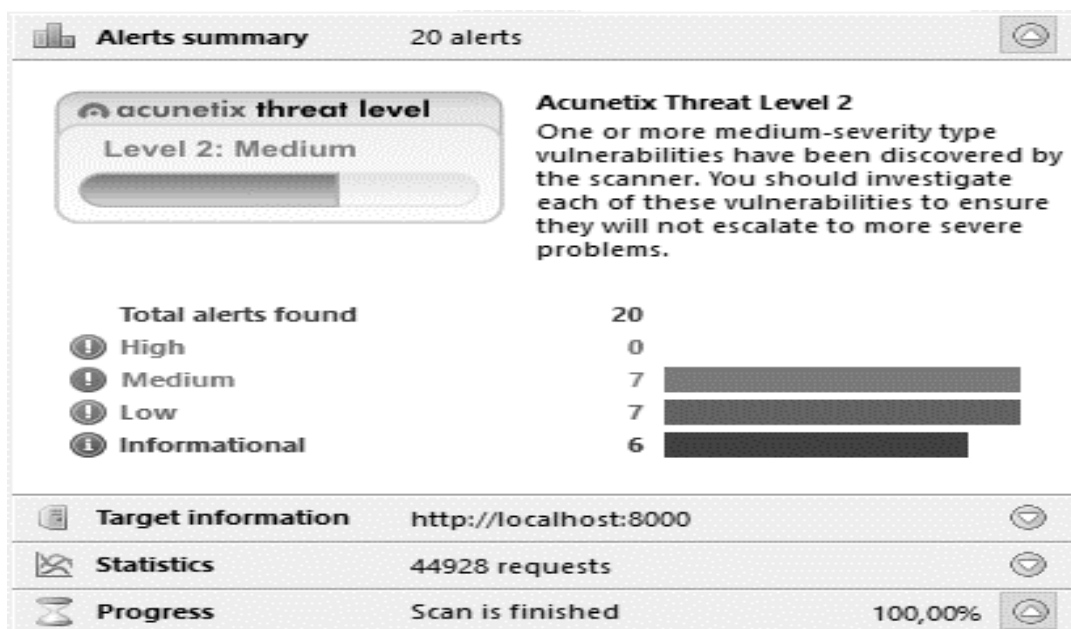


Figura 20: Prueba de seguridad: 1ra iteración. (Elaboración propia)

En la primera iteración no se detectó ninguna vulnerabilidad de alta complejidad, de complejidad media se detectó 7, las credenciales de usuario se transmiten a través de un canal sin cifrar. Esta información siempre debe transferirse a través de un canal cifrado (HTTPS) para evitar ser interceptada por usuarios malintencionados. De baja complejidad se detectaron 7 vulnerabilidades, en las cuales se encontró la existencia de directorios sensibles este control busca recursos sensibles comunes como directorios de copia de seguridad, volcados de base de datos, páginas de administración, directorios temporales. Cada uno de estos directorios podría ayudar a un atacante a aprender más sobre su objetivo. Se encontraron también vulnerabilidades de Cookie de sesión sin conjunto de indicadores seguros y posibles hosts virtuales mientras que de carácter informativo se detectaron 6.

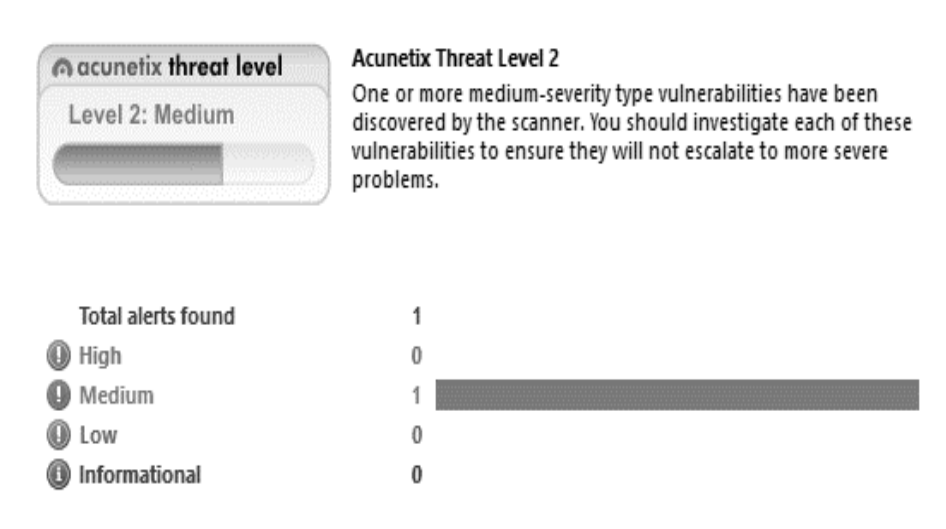


Figura 21: Prueba de seguridad: 2da iteración. (Elaboración propia)

Para la segunda iteración se corrigieron las 20 vulnerabilidades de las detectadas en la iteración 1, mientras que en esta iteración se detectaron 1 alerta tipo media, donde se mantuvo la vulnerabilidad de las credenciales en texto plano y credenciales encontradas en un fichero.

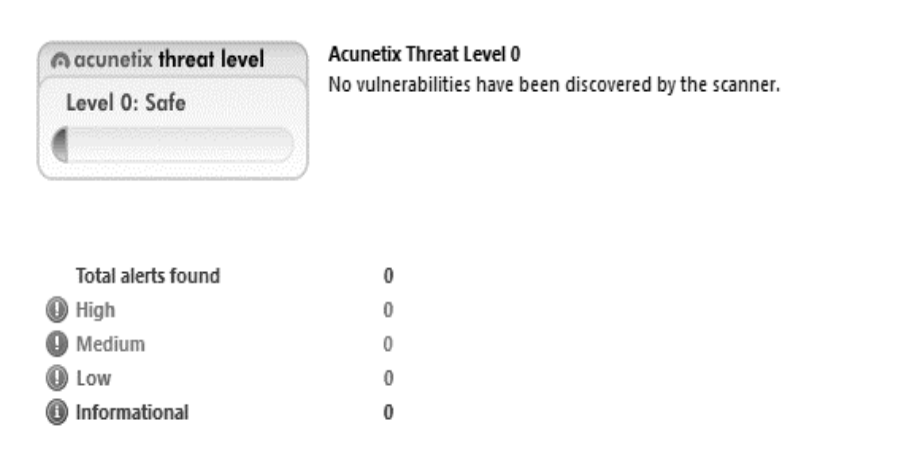


Figura 22: Prueba de seguridad: 3ra iteración. (Elaboración propia)

En la última iteración se evidencia la solución de las no conformidades de la iteración anterior, y no se encontraron nuevas vulnerabilidades.

3.2.4. Pruebas funcionales

Pruebas de caja negra

Las pruebas de caja negra, también llamadas pruebas de comportamiento, se enfocan en los requerimientos funcionales del software; es decir, las técnicas de prueba de caja negra le permiten derivar conjuntos de condiciones de entrada que revisarán por completo todos los requerimientos funcionales para un programa. Las pruebas de caja negra no son una alternativa para las técnicas de caja blanca. En vez de ello, es un enfoque complementario que es probable que descubra una clase de errores diferente que los métodos de caja blanca. Las pruebas de caja negra intentan encontrar errores en las categorías siguientes:

- 1) funciones incorrectas o faltantes,
- 2) errores de interfaz,
- 3) errores en las estructuras de datos o en el acceso a bases de datos externas,
- 4) errores de comportamiento o rendimiento y
- 5) errores de inicialización y terminación.

A diferencia de las pruebas de caja blanca, que se realizan tempranamente en el proceso de pruebas, la prueba de caja negra tiende a aplicarse durante las últimas etapas de la prueba. Puesto que, a propósito, la prueba de caja negra no considera la estructura de control, la atención se enfoca en el dominio de la información. Las pruebas se diseñan para responder a las siguientes preguntas: (12)

- ✚ ¿Cómo se prueba la validez funcional?
- ✚ ¿Cómo se prueban el comportamiento y el rendimiento del sistema?
- ✚ ¿Qué clases de entrada harán buenos casos de prueba?
- ✚ ¿El sistema es particularmente sensible a ciertos valores de entrada?
- ✚ ¿Cómo se aíslan las fronteras de una clase de datos?
- ✚ ¿Qué tasas y volumen de datos puede tolerar el sistema?
- ✚ ¿Qué efecto tendrán sobre la operación del sistema algunas combinaciones específicas de datos?

Aplicación de las pruebas de caja negra

Caso de prueba Gestionar Solicitud de servicios:

Tabla 4: Variables para los casos de prueba funcional Gestionar Solicitud de servicios. (Elaboración propia)

No.	Nombre del campo	Clasificación	Valor Nulo	Descripción
1	Número de solicitud	Campo numérico	No	Se inserta una cadena de texto, puede contener valores consecutivos numéricos y puntos.
2	Fecha de inscripción	Campo fecha seleccionable	No	Se selecciona la fecha asociada a la inscripción de la solicitud de servicio.
3	Cliente	Campo seleccionable	No	Se selecciona el cliente asociada a la solicitud de servicio.
4	Fecha y hora del servicio	Campo fecha y hora seleccionable	No	Se selecciona la fecha y hora que se realizará el servicio.
5	Dirección del lugar	Campo alfanumérico	No	Se inserta una cadena de texto, puede contener valores alfanuméricos.
6	Solicitante	Campo seleccionable	No	Se selecciona el Solicitante que realiza la solicitud de servicio.
7	Operación	Campo seleccionable	No	Se selecciona el tipo de operación asociada a la solicitud de servicio.
8	Tipo de carga	Campo seleccionable	No	Se selecciona el tipo de carga asociada a la solicitud de servicio.
9	Descripción de la mercancía	Campo alfanumérico	No	Se inserta una cadena de texto, puede contener valores alfanuméricos.
10	Servicios solicitados	Campo multi-seleccionable	No	Se selecciona o seleccionaran el tipo de tipo de servicio que se solicita o solicitan

11	Método y alcance de la inspección	Campo seleccionable	No	Se selecciona el método de alcance de la inspección asociada a la solicitud de servicio.
12	Observaciones	Campo alfanumérico	Si	Se inserta una cadena de texto, puede contener valores alfanuméricos.

Las celdas de la tabla contienen V (indica válido), I (indica inválido), N/S (No es necesario llenar).

Tabla 5: Caso de prueba funcional Nueva Solicitud de servicios. (Elaboración propia)

Escenario	Descripción	V1	V2	V11	V12	R/ del sistema	Flujo central
EC1.1 Introducir datos correctamente	El usuario con permisos correspondientes crea la nueva solicitud de servicios introduciendo los datos correctamente.	V	V	V	V	V	El sistema debe mostrar un mensaje de notificación indicando "Solicitud de servicio realizada satisfactoriamente".	Se sigue la ruta en el árbol de menú de la interfaz principal: "Solicitud de servicios" y "Nueva solicitud de servicios". Aparecerá el formulario con los datos correspondientes a la nueva solicitud de servicios, una vez insertados los datos de la solicitud de servicios presiona sobre

								<p>el botón "ACEPTAR".</p> <p>Si no desea crear la solicitud de servicios oprimir el botón "CANCELAR" para ir al formulario de listar dicha funcionalidad.</p>
EC1.2	El usuario deja uno o más campos vacíos.						El sistema muestra un mensaje de error.	Se sigue la ruta en el árbol de menú de la interfaz principal:
Campos vacíos		(vacío)	(vacío)	(vacío)	(vacío)	(vacío)	1. "(*) los campos son obligatorios."	<p>"Solicitud de servicios" y "Nueva solicitud de servicios". Aparecerá el formulario con los datos correspondientes a la nueva solicitud de servicios, una vez insertados los datos de la solicitud de servicios</p>

								<p>presiona sobre el botón "ACEPTAR".</p> <p>Si no desea crear la solicitud de servicios oprimir el botón "CANCELAR" para ir al formulario de listar dicha funcionalidad.</p>
EC1.3	No permite insertar una penalidad correctamente.						<p>El sistema muestra un mensaje de error.</p> <p>1. "El campo <nombre del campo> esta de forma incorrecta."</p> <p>2. "El campo <nombre del campo> debe estar solo compuesto por caracteres numéricos y separados por puntos."</p>	<p>Se sigue la ruta en el árbol de menú de la interfaz principal: "Solicitud de servicios" y "Nueva solicitud de servicios". Aparecerá el formulario con los datos correspondientes a la nueva solicitud de servicios, una vez insertados los datos de la solicitud de</p>

								<p>servicios presiona sobre el botón "ACEPTAR".</p> <p>Si no desea crear la solicitud de servicios oprimir el botón "CANCELAR" para ir al formulario de listar dicha funcionalidad.</p>
EC1.5 Cancelar	Se cancela la operación para insertar una penalidad.	NS	NS	NS	NS	NS	Cierra la ventana, no guarda los cambios realizados.	<p>Se sigue la ruta en el árbol de menú de la interfaz principal: "Solicitud de servicios" y "Nueva solicitud de servicios". Aparecerá el formulario con los datos correspondientes a la nueva solicitud de servicios, una vez insertados los datos de la</p>

								<p>solicitud de servicios presiona sobre el botón "ACEPTAR".</p> <p>Si no desea crear la solicitud de servicios oprimir el botón "CANCELAR" para ir al formulario de listar dicha funcionalidad.</p>
--	--	--	--	--	--	--	--	--

Tabla 6: Caso de prueba funcional Eliminar solicitud de servicios. (Elaboración propia)

Escenario	Descripción	R/ del sistema	Flujo central
EC2.1 Eliminar una solicitud de servicios correctamente.	Permite eliminar una solicitud de servicios correctamente.	El sistema elimina el servicio y muestra un mensaje "Solicitud de servicio eliminada con éxito".	Se sigue la ruta en el árbol de menú de la interfaz principal: "Solicitud de servicios" y "Lista de solicitudes de servicios". Aparecerá un listado con las solicitudes de servicios existentes y varias opciones, presiona sobre el botón "ELIMINAR" el servicio solicitado que desea eliminar. Aparecerá un mensaje de confirmación: "Esta seguro que desea eliminar el servicio <número del servicio>", oprimir el botón "ACEPTAR" para eliminar la solicitud de servicio (si desea volver al formulario del listado de las solicitudes y no eliminarlo oprimir el botón "CANCELAR").

Tabla 7: Caso de prueba funcional Modificar Solicitud de servicios. (Elaboración propia)

Escenario	Descripción	V1	V2	V11	V12	R/ del sistema	Flujo central
EC4.1 Modificar solicitudes de servicios correctamente.	Permite modificar un solicitud de servicio correctamente	V	V	V	V	V	El sistema valida los campos, los adiciona o modifica y muestra un mensaje indicando, "Solicitud de servicio editado satisfactoriamente".	Se sigue la ruta en el árbol de menú de la interfaz principal: "Solicitud de servicios" y "Lista de solicitudes de servicios". Aparecerá un listado con las solicitudes de servicios existentes y varias opciones, presiona sobre el botón "EDITAR" de la solicitud de servicio que desea editar y aparecerá el formulario con los datos correspondientes a la solicitud de servicios seleccionada.

<p>EC 4.2</p> <p>Modificar una solicitud de servicios dejando campos vacíos.</p>	<p>El usuario deja uno o más campos vacíos.</p>	<p> </p> <p>(vacío)</p>	<p> </p> <p>(vacío)</p>	<p> </p> <p>(vacío)</p>	<p> </p> <p>(vacío)</p>	<p> </p> <p>(vacío)</p>	<p>El sistema muestra un mensaje de error.</p> <p>1. “(*) los campos son obligatorios.”</p>	<p>Se sigue la ruta en el árbol de menú de la interfaz principal:</p> <p>“Solicitud de servicios” y “Lista de solicitudes de servicios”.</p> <p>Aparecerá un listado con las solicitudes de servicios existentes y varias opciones, presiona sobre el botón “EDITAR” de la solicitud de servicio que desea editar y aparecerá el formulario con los datos correspondientes a la solicitud de servicios seleccionada.</p>
<p>EC 4.3</p> <p>Ingreso de</p>	<p>No permite insertar una</p>	<p> </p>	<p> </p>	<p> </p>	<p> </p>	<p> </p>		<p>Se sigue la ruta en el árbol de</p>

caracteres inválidos a la hora de Modificar.	solicitud de servicios correctamente.						El sistema muestra un mensaje de error. 1. "El campo <nombre del campo> esta de forma incorrecta." 2. "El campo <nombre del campo> debe estar solo compuesto por caracteres numéricos y separados por puntos."	menú de la interfaz principal: "Solicitud de servicios" y "Lista de solicitudes de servicios". Aparecerá un listado con las solicitudes de servicios existentes y varias opciones, presiona sobre el botón "EDITAR" de la solicitud de servicio que desea editar y aparecerá el formulario con los datos correspondientes a la solicitud de servicios seleccionada.
EC 4.4 Cancelar	Se cancela la operación para insertar	NS	NS	NS	NS	NS	Cierra la ventana, no guarda los cambios realizados.	Se sigue la ruta en el árbol de menú de la interfaz principal:

	una solicitud de servicios.							<p>“Solicitud de servicios” y “Lista de solicitudes de servicios”. Aparecerá un listado con las solicitudes de servicios existentes y varias opciones, presiona sobre el botón “EDITAR” de la solicitud de servicio que desea editar y aparecerá el formulario con los datos correspondientes a la solicitud de servicios seleccionada.</p>
--	-----------------------------	--	--	--	--	--	--	---

Resultados de las pruebas funcionales

Para la validación de los requisitos funcionales se realizaron tres iteraciones donde se encontraron un total de 20 no conformidades, 20 en la primera iteración de las cuales 15 fueron resueltas quedando 5 pendientes; 5 en la segunda iteración, las cuales fueron pendientes de la iteración anterior, estas fueron resueltas en su totalidad. Se realizó una tercera iteración donde no se detectó ninguna no conformidad, arrojando como

resultado final que el sistema funciona correctamente. En la siguiente figura se muestran los resultados obtenidos:

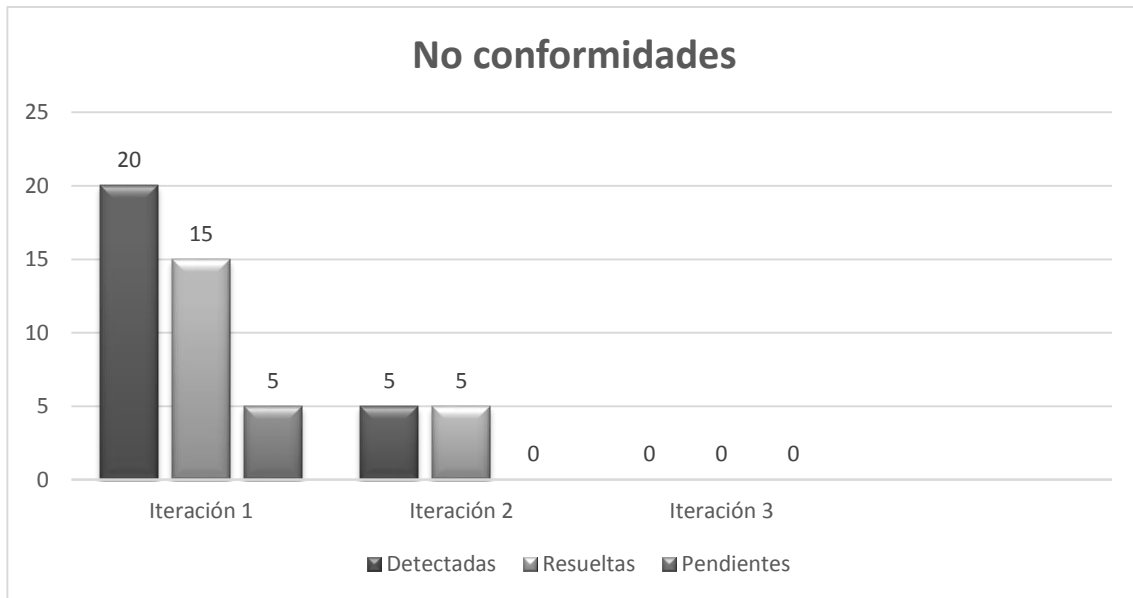


Figura 23: Comportamiento de las no conformidades por iteración. (Elaboración propia)

3.2.5. Pruebas de aceptación

Cuando se construye software a la medida para un cliente, se realiza una serie de pruebas de aceptación a fin de permitir al cliente validar todos los requerimientos. Realizada por el usuario final en lugar de por los ingenieros de software, una prueba de aceptación puede variar desde una “prueba de conducción” informal hasta una serie de pruebas planificadas y ejecutadas sistemáticamente. De hecho, la prueba de aceptación puede realizarse durante un periodo de semanas o meses, y mediante ella descubrir errores acumulados que con el tiempo puedan degradar el sistema. (27)

Aplicación de las pruebas de aceptación

- ✚ **Código:** identificador de la prueba realizada sugerente a la HU a la que hace referencia.
- ✚ **Nombre:** nombre de la prueba a realizar.
- ✚ **Nombre del probador:** nombre de la persona que realiza la prueba.
- ✚ **Descripción:** se describe la funcionalidad que se desea probar.

- ✚ **Condiciones de Ejecución:** mostrará las condiciones que deben cumplirse para poder llevar a cabo el caso de prueba, estas condiciones deben ser satisfechas antes de la ejecución del caso de prueba para que se puedan obtener los resultados esperados.
- ✚ **Entradas/Pasos de Ejecución:** descripción de cada uno de los pasos seguidos durante el desarrollo de la prueba, se tiene en cuenta cada una de las entradas que hace el usuario con el objetivo de ver si se obtiene el resultado esperado.
- ✚ **Resultado esperado:** breve descripción del resultado que se espera obtener con la prueba realizada.
- ✚ **Evaluación de la prueba:** acorde al resultado de la prueba realizada se emitirá una evaluación sobre la misma. Esta evaluación tendrá uno de los dos valores que a continuación se describen:
 - ❖ Satisfactorio
 - ❖ Insatisfactorio

Caso de prueba Gestionar RRHH

Tabla 8: Caso de prueba Gestionar RRHH. (Elaboración propia)

Caso de prueba aceptación	
Código de caso de prueba: CISED_ASIC-HU_08-8	Nombre de historia de usuario: Gestionar RRHH.
Nombre de la persona que realiza la prueba: Marbier Pérez Polo.	
Descripción de la prueba: Prueba a la funcionalidad nueva RRHH.	
Entrada/Pasos de ejecución: Se sigue la ruta en el árbol de menú de la interfaz principal: "Recursos Humanos" y "Nuevo trabajador". Aparecerá el formulario con los datos correspondientes al nuevo RRHH, una vez insertados presiona clic sobre el botón "Aceptar". Si no desea crear el RRHH oprimir el botón "Cancelar" para ir al formulario del listado.	

Oficinas: La Habana

Código: 02.12

Nombre: Marbier

Apellidos: Pérez Polo

CI: 89062701146

Sexo: Masculino

Cargo: Asesor C Jurídico

Fecha de alta: 2017-05-18

Observaciones: -

Escenarios:	Resultados Esperado	Evaluación de la prueba:
EC 1.1 Insertar RRHH correctamente.	Muestra el listado de los RRHH incluyendo el RRHH insertado.	Satisfactoria.
EC 1.2 Ingreso de caracteres inválidos al insertar un RRHH.	<p>Muestra un mensaje según el error encontrado:</p> <p>1- “El campo <nombre del campo> debe estar compuesto solo por caracteres numéricos y separado por puntos.”</p> <p>2. “El campo <nombre del campo> debe estar compuesto solo por letras, separado por espacio y empezar con mayúscula.”</p> <p>3. “El campo <nombre del campo> debe estar solo</p>	Satisfactoria.

	compuesto por caracteres numéricos.”	
EC 1.3 Insertar un RRHH dejando campos vacíos.	Muestra un mensaje según el error encontrado: 1- Si es un listado: “El campo seleccionable <nombre del campo> es obligatorio.”	Satisfactoria.
EC 1.4 Cancelar	Cierra la ventana, no guarda los cambios realizados.	Satisfactoria.

Caso de prueba Gestionar Solicitud de servicios

Tabla 9: Caso de prueba Gestionar Solicitudes de servicios. (Elaboración propia)

Caso de prueba aceptación	
Código de caso de prueba: CISED_ASIC-HU_04-4	Nombre de historia de usuario: Gestionar Solicitud de servicios.
Nombre de la persona que realiza la prueba: Marbier Pérez Polo.	
Descripción de la prueba: Prueba a la funcionalidad nueva solicitud de servicios.	
Entrada/Pasos de ejecución: Se sigue la ruta en el árbol de menú de la interfaz principal: “Solicitud de servicios” y “Nueva solicitud de servicios”. Aparecerá el formulario con los datos correspondientes a la nueva solicitud de servicios, una vez insertados presiona clic sobre el botón “Aceptar”. Si no desea crear la solicitud de servicios oprimir el botón “Cancelar” para ir al formulario del listado.	

Número de solicitud: 01.03.01.2017

Fecha de inscripción: 2017-05-31

Cliente: UCI

Lugar de servicio: UCI

Fecha y hora: 2012-12-21 15:25

Dirección del lugar: UCI

Operación: Economía Interna

Tipo de carga: Contenerizada

Descripción de la mercancía: PC de escritorios con mouse, teclado, monitor

Servicio solicitado: Inspección al inventario, Vaciado de contenedores completo(FCL/FCL)

Observaciones: Mercancías frágiles

Escenarios:	Resultados Esperado	Evaluación de la prueba:
EC 4.1 Insertar solicitud de servicio correctamente.	Muestra el listado de las solicitudes de servicios incluyendo la solicitud de servicio insertada.	Satisfactoria.
EC 4.2 Ingreso de caracteres inválidos al insertar una solicitud de servicio.	<p>El sistema muestra un mensaje de error.</p> <ol style="list-style-type: none"> 1. "El campo <nombre del campo> esta de forma incorrecta." 2. "El campo <nombre del campo> debe estar solo compuesto por caracteres numéricos y separados por puntos." 	Satisfactoria.

<p>EC 4.3 Insertar una solicitud de servicios dejando campos vacíos.</p>	<p>Muestra un mensaje según el error encontrado:</p> <p>1- Si es un listado: “El campo seleccionable <nombre del campo> es obligatorio.”</p>	<p>Satisfactoria.</p>
<p>EC 4.4 Cancelar</p>	<p>Cierra la ventana, no guarda los cambios realizados.</p>	<p>Satisfactoria.</p>

Resultado de las pruebas aceptación

Las pruebas de aceptación dieron como resultado que no existía ninguna no conformidad realizada por el cliente quedando así satisfecho con el producto final.

3.3. Conclusiones parciales

En este capítulo se han abordado los elementos correspondientes a la culminación y aprobación del módulo de gestión Comercial para el Sistema de Asesoría de la Agencia de Supervisión e Inspección de Carga. La enunciación de los estándares de codificación a utilizar en la implementación de la solución permitió un mejor entendimiento del código y diseño de la solución. Se comprobó el correcto funcionamiento del módulo Comercial a través de la realización de pruebas al software. Se logró una satisfacción del cliente, ya que la propuesta solución planteada fue de su complacencia y comprensión. También se logró un grado de rendimiento aceptable y la seguridad se encuentra de forma grata. Se culminó, después de varias iteraciones de pruebas, una solución sin la obtención de no conformidades.

CONCLUSIONES

Una vez culminado el trabajo de diploma permitió no solo desarrollar el Módulo de gestión Comercial para el Sistema de Asesoría de la Agencia de Supervisión e Inspección de Cargas, sino que permitió arribar a las siguientes conclusiones:

- ✓ El establecimiento de los fundamentos teórico-metodológicos de la investigación realizada permitió conceptualizar los procesos de gestión de la información de la empresa.
- ✓ El análisis de la situación actual de la gestión de la información en la colaboración económica desde el exterior permitió constatar que las soluciones existentes para el proceso de gestión de la información en esta área no responden a las exigencias presentadas por el ASIC. Esto aportó como resultado la necesidad de crear una aplicación que responda a las exigencias de la organización.
- ✓ El establecimiento de los fundamentos que debe sostener el módulo web permitió identificar los requisitos funcionales y no funcionales a desarrollar dando a conocer la propuesta de solución.
- ✓ Las pruebas desarrolladas al sistema, arrojaron resultados satisfactorios. Dándole culminación a los módulos implementados, quedando satisfecho el cliente con la propuesta solución.

RECOMENDACIONES

- ✓ Se recomienda implementarle al módulo de gestión Comercial la funcionalidad de llevar el registro de todas las acciones realizadas por un usuario dentro del sistema, para una mejor administración de la información que se manipula.
- ✓ Para las futuras versiones del cliente es necesario crear un sistema de ayuda que guíe al usuario mientras navega por las diferentes vistas y un manual de usuarios.

BIBLIOGRAFÍA

1. Heras Saizarbitoria, Iñaki. Los estándares internacionales de gestión: pasado, presente y futuro. s.l. : Martí : CASADESÚS FA, 2006. págs. 45-61. Boletín económico de ICE.
2. El concepto de tecnologías de la información. Benchmarking sobre las definiciones de las TIC en la sociedad del conocimiento. Romaní Cobo, Juan Cristóbal. 2009 : s.n., Revista de estudios de comunicación=Komunikazio ikasketen aldizkaria, págs. 295-318.
3. FACULTAD DE ADMINISTRACION DE EMPRESAS Y NEGOCIOS INTERNACIONALES TEMA: SISTEMAS DE GESTION PARA LAS EMPRESAS COMERCIALES. Luna, Kalu. 16 de Diciembre de 2014. FACULTAD DE ADMINISTRACION DE EMPRESAS Y NEGOCIOS INTERNACIONALES TEMA: SISTEMAS DE GESTION PARA LAS EMPRESAS COMERCIALES.
4. Definición MX. [En línea] 27 de Junio de 2014. [Citado el: 26 de Mayo de 2017.] Definición de Contratación. <https://definicion.mx/contratacion/>.
5. [En línea] 1 de Febrero de 2013. [Citado el: 26 de Mayo de 2017.] Su Definición, Concepto y Significado. <http://conceptodefinicion.de/servicio/>.
6. MENDOZA CORDOVA, Cinthya Geoconda. La gestión del talento humano y su incidencia en el servicio del Sindicato de Choferes Profesionales del cantón Quevedo, año 2014. Quevedo-UTEQ. 2016. Tesis de Licenciatura.
7. García, Gustavo Villar. GARCÍA, Gustavo Villar. Implementación de un sistema de gestión de la investigación para la especialidad de Ingeniería Informática de la PUCP. Facultad de Ciencias e Ingeniería, Pontificia Universidad Católica del Perú. 2015. Tesis Doctoral. Mención: Ingeniería Informática.
8. Herrero Palomo, Julián. Administración, gestión y comercialización en la pequeña empresa. s.l. : Editorial Paraninfo, 2001. pág. 151.
9. GestioPolis.com Experto. [En línea] 16 de Marzo de 2001. [Citado el: 28 de Mayo de 2017.] <https://www.gestiopolis.com/que-es-gestion-comercial/>.
10. Booch, G., Rumbaugh, J. y Jacobson, I. Unified Modeling Language User Guide. s.l. Segunda. s.l. : Addison-Wesley Object Technology Series, 2005.

11. Diagrama de Clases de Diseño | LENGUAJE DE MODELADO UNIFICADO - UML. [En línea] 2015. [Citado el: 11 de Mayo de 2017.] http://stadium.unad.edu.co/ovas/10596_9836/diagrama_de_clases_de_diseo.html.
12. Pressman, Roger S. Ingeniería de Software: Un enfoque práctico. Séptima. s.l. : McGraw-Hill, 2002. Universidad de Connecticut. ISBN 978-607-15-0314-5.
13. VAN LANCKER, Luc. HTML5 y CSS3: Domine los estándares de las aplicaciones Web [2ª edición]. s.l. : Ediciones ENI, 2013.
14. NIXON, Robin. Learning PHP, MySQL, JavaScript, CSS & HTML5: A Step-by-Step Guide to Creating Dynamic Websites. s.l. : O'Reilly Media, Inc., 2014.
15. Apache JMeter - Apache JMeter™. [En línea] 4 de Julio de 2017. [Citado el: 20 de Mayo de 2017.] <http://jmeter.apache.org/>.
16. Inge-soft. [En línea] 25 de Marzo de 2010. [Citado el: 21 de Febrero de 2017.] <http://inge-soft.blogspot.com/2010/03/conceptos-y-principios-del-analisis.html>.
17. Larman, Craig. UML y Patrones: Una introducción al análisis y diseño orientado a objetos y al proceso unificado. [ed.] Prentice Hall. Segunda. 2000.
18. Pressman, Roger S. Ingeniería de software: Un enfoque práctico. Séptima. Mexico : McGraw Hill, 2010.
19. Steinberg, Daniel H y Palmer, Daniel W. Extreme Software Engineering. s.l. : Prentice-Hall, Inc., 2003. ISBN 0-13-047381-2.
20. IEEE-Std-1471-2000 Recommended Practice for Architectural Description of Software-Intensive Systems. Group, A. W. s.l. : Institute of Electrical and Electronics (IEEE), 14 de Noviembre de 2000. IEEE 1471.
21. Condori Ayala, José Luis. Revista de Información, Tecnología y Sociedad. [En línea] [Citado el: 20 de Marzo de 2017.] www.revistasbolivianas.org.bo.
22. Python - DjangoFramework de desarrollo web para perfeccionistas Basado en el Modelo MTV. Condori Ayala, José Luis. 7, La Paz : s.n., Noviembre de 2012, Revista de Información, Tecnología y Sociedad, pág. 36. www.revistasbolivianas.org.bo. ISSN 1997-4044.

23. Appleton, B. Patterns and software: Essential Concepts and Terminology. [aut. libro] Roger S Pressman. Ingeniera del Software: Un enfoque práctico. Mexico : McGRAW-HILL, 2010.
24. PIATTINI, Mario. Análisis y diseño de aplicaciones informáticas de gestión. s.l. : Alfa y Omega, Bogotá, 2007.
25. GUEDES, Gilleanes T. UML: uma abordagem prática. s.l. : Novatec Editora, 2008.
26. MARZAL, Andrés y LUENGO, Isabel Gracia. Introducción a la Programación con Python y C. s.l. : Publicacions de la Universitat Jaume I, 2004.
27. PRESSMAN, Roger S y TROYA, Jose Maria. Ingeniería del software. 1988.
28. Dsdm Dynamic Systems Development Method: The Method in Practice. Stapleton, J. s.l. : Addison-Wesley, 1997.
29. Agile Software Development with SCRUM. Schwaber, K., Beedle, M. y Martin, R.C. s.l. : Prentice Hall, 2001.
30. LA INTEGRACIÓN DE SISTEMAS DE GESTIÓN BASADOS EN ESTÁNDARES INTERNACIONALES: RESULTADOS DE UN ESTUDIO EMPÍRICO REALIZADO EN LA CAPV1. SAIZARBITORIA, IÑAKI HERAS, BERNARDO, MERCÈ y CASADESÚS, MARTÍ. 14, s.l. : Revista de Dirección y Administración de Empresas, 2007, págs. 155-174.
31. Lean Software Development: An Agile Toolkit for Software Development Managers. Poppendieck, M. y Poppendieck, T. s.l. : Addison Wesley, 2003.
32. Medina, J. Manual de prospectiva y decisión estratégica: bases teóricas e instrumentos para América Latina y el Caribe. 2006.
33. LETELIER, Patricio. Metodologías ágiles para el desarrollo de software: eXtreme Programming (XP). 2006.
34. jfmonzon. Adigal. [En línea] [Citado el: 6 de Diciembre de 2016.] Software ERP de gestión comercial para facturar y control de almacen. Programa de compras y ventas para todo tipo de empresas. Trazabilidad de alimentos. <http://www.adigal.com/software-gestion-comercial/>.
35. Agile Software Development Ecosystems. Highsmith, J. s.l. : Addison-Wesley, 2002.

-
36. Adaptive Software Development: A Collaborative Approach to Managing Complex Systems. Highsmith, J y Orr, K. s.l. : Dorset House, 2000.
 37. GINESTÀ, Marc Gibert y MORA, Oscar Pérez. Bases de datos en PostgreSQL.
 38. Gestión comercial desde un enfoque prospectivo. DÍAZ, Yailín Infantes, VIVAR, Rodobaldo Martínez y RODRÍGUEZ, Alexander Sánchez. 18, 2015, Entelequia: revista interdisciplinar, págs. 143-166.
 39. Agile Software Development. Cockbun, A. s.l. : Addison-Wesley, 2001.
 40. Java Modeling In Color With UML: Enterprise Components and Process. Coad, P., Lefebvre, E. y De Luca, J. s.l. : Prentice Hall, 1999.
 41. CASADESÚS, M., HERAS, I. y MERINO, J. Calidad Práctica. Madrid. : Prentice Hall - Financial Times, Pearson Educación S.A., 2005.
 42. BORJAS, D. Propuesta de procedimiento para la gestión comercial en la Cadena de Tiendas TRD Caribe en Cuba. Tesis en opción al grado de Master en Dirección. . Universidad de Holguín. Cuba. : s.n., 2009.
 43. Extreme Programming Explained. Embrace Change. Beck, K. s.l. : Pearson Education, 1999. Traducido al español como: "Una explicación de la programación extrema. Aceptar el cambio", Addison Wesley, 2000..