



Módulo Transporte para el Sistema de Gestión Integral del Agente Transitorio

Trabajo de diploma para optar por el título de Ingeniero en
Ciencias Informáticas

Autores: Liuba de la Caridad Hernández Rodríguez

Daryl Yturalde López

Tutores: Ing. Irina Ivis Santiesteban

Ing. Leannys Rodríguez Moreno

Co-tutores: Ing. Carlos Montenegro Amador

Ing. Amet Martínez Verdecia

La Habana, 7 de Junio de 2016

“Año 58 de la Revolución”

Declaración de Autoría

Declaración de Autoría

Declaramos ser los autores del presente trabajo de diploma y otorgamos a la Universidad de las Ciencias Informáticas los derechos patrimoniales de la misma, con carácter exclusivo. Para que así conste firmamos la presente a los ____ días del mes de _____ del año _____.

Firma del Autor

Liuba de la Caridad Hernández Rodríguez

Firma del Autor

Daryl Yturralde López

Firma del Tutor

Ing. Irina Ivis Santiesteban

Firma del Tutor

Ing. Leannys Rodríguez Moreno

Pensamiento



“ En la tierra hacen falta personas que trabajen más y critiquen menos, que construyan más y destruyan menos, que prometan menos y resuelvan más, que esperen recibir menos y dar más, que digan mejor ahora que mañana. ”

Dedicatoria

Líuba

Dedico este trabajo a las personas más importantes de mi vida, a mi mamá, mi hermana, mi abuela, mi padrastro, a mi novio y a mis tíos por todo su apoyo incondicional, principalmente por ser mis amigos, confidentes y brindarme su apoyo y confianza en todas las decisiones de mi vida.

Daryl

Me dedico esta tesis, por los esfuerzos realizados durante estos 5 años, por las noches que pase despierto estudiando y los sacrificios hechos para poder cumplir este sueño.

A una de las personas más importantes en mi vida a mi mamá por su dedicación, amor, comprensión, apoyo, amistad incondicional, por su incansable exigencia, por su ayuda incondicional, por haberme traído al mundo y haber pasado junto a mi momentos importantes en mi vida.

A mi hermano Erik por compartir y formar parte de mi vida he inculcarme valores, ayudarme a tomar decisiones y aconsejarme en momentos críticos.

A mi cacharro quien estuvo a mi lado en los momentos buenos y malos, muchas veces fajados pero siempre juntos ya que somos hermanos inseparables y en mi corazón siempre está presente.

A dos personas importantes en mi vida a mi abuela Deyanira por los conocimientos y las preocupaciones transmitidas, por el cariño dado y por siempre preocuparse por su nieto, te lo dedico a ti también abuela Hilda que aunque hoy no te encuentras entre nosotros siempre fuiste una excelente abuela y madre, te quiero y te voy a querer siempre y sé que estés donde estés velas por tu nieto.

A otra de las personas especiales, a mi compañera de tesis y de la vida, mi saquito de pecas, por su apoyo, dedicación, amistad, comprensión, por soportarme durante estos 5 años y por formar parte de este sueño.

Dedico esta tesis y este título obtenido a una persona súper especial en mi vida que desde que nací ha corrido conmigo, que al igual que mi madre ha sufrido, llorado y sentido por mí, que ha sido mi inspiración, modelo a seguir e imagen, a mi padre que tanto quiero y amo que ha dado todo lo que tiene y lo que no tiene por verme realizar este sueño que es de los tres, mío de mi mamá y en especial tuyo, te lo prometí y te lo cumplí aquí tienes a tu ingeniero este título es tuyo mucho más que mío, porque eres guía, apoyo, profesor, tutor, compañero y amigo.

Agradecimientos de Liuba

Agradecimientos de Liuba

Primero que todo les agradezco a mi mamá y mi hermana, porque sin ellas no hubiese podido cumplir este sueño, gracias por el amor incondicional que me brindaron y por siempre confiar en mí.

Mamá gracias por quererme tanto, comprenderme siempre, por enseñarme que todo en la vida lleva sacrificio, gracias a ti aprendí a valerme por mí misma y a saber levantarme todas las veces que me he caído, gracias por ser madre y padre a la vez, por guiarme siempre, por enseñarme a dar lo mejor de mí, porque sin tí no sería lo que soy ahora, gracias por ser la mejor madre del mundo; te quiero mucho mamá.

A mi hermana por ser mi amiga incondicional, compañera, confidente y por quererme como soy.

A mi padrastro Yosvany por quererme y cuidarme como si fuera su hija, por el amor y comprensión que me ha brindado.

A tí Daryl gracias por tu amor, cariño, apoyo, comprensión, por aguantarme en estos 5 años de relación todas mis malcriadeces, que sé que no ha sido nada fácil; porque siempre que te he necesitado has estado para mí, en las malas y las buenas, por quererme como tú sabes hacerlo de una manera especial. TE AMO.

A mi abuela por cuidarme y malcriarme como solo ella lo sabe hacer. Le pido a dios que le de mucha salud para viva muchos años más.

A mi familia en especial a mis tíos Leandro, Lizandra y Lisbet que siempre me han estado apoyado durante toda mi vida y por estar para mí siempre que los necesité.

A mi segunda familia Reynaldo, Damaris y Dayrol por ser como mis segundos padres y hermano incondicional, por todo el amor y cariño sincero que me han brindado.

A mis compañeros de grupo los que se han quedado en el camino y los que aún siguen luchando por su sueño, por todos los momentos de alegrías y tristezas que pasamos juntos.

A mis grandes amigos Maydalis (Mamucha), Taituza, Gus (Gustavo) y Arielito por ser amigos incondicionales, por compartir muchos momentos juntos agradables, de fiesta, convivencia, amistad, fue muy bonito haberlos conocido los quiero mucho.

A todos aquellos profesores que contribuyeron a mi formación como profesional.

A los tutores por todo su apoyo, ayuda y dedicación en la realización de este sueño. En fin a todos los que de una forma u otra me han ayudado en la realización de la tesis.

Agradecimientos de Daryl

Agradecimientos de Daryl

Agradezco a la Revolución por darme la oportunidad de superarme y convertirme en ingeniero.

A mis tutores y cotutores por toda la ayuda prestada en especial a Carlos, Irina y Amet.

A mi familia por ayudarme y apoyarme, a mis primas y primos a mis tíos Vicentico y Rafael, a mi tía Maribel.

A mis amistades cesar, Adrián, Frank, Ángel, Yusniel, y todos los del grupo por sus críticas instructivas y momentos compartidos durante estos 5 años juntos que hemos pasado.

A los amigos que se quedaron en el camino que también fueron de ayuda en estos 5 años.

A todos los profesores por el excelente trabajo realizado en mi formación universitaria.

Y a aquellos que ya forman parte de mi familia y los quiero como tal a Yosvany, Liset, Leandro, Esperanza, Lizandra, Yuliosquí, Yaneisi, Lisbet, Laritza y mi sobrinita Verónica.

Resumen

La tendencia en el mundo actual es la informatización de diferentes empresas y organismos apoyados en soluciones informáticas para disminuir en gran medida los problemas empresariales. La empresa TRANSCARGO se encuentra inmersa en estas transformaciones; la misma cuenta con un sistema desarrollado en Microsoft Access que se encarga de gestionar la información relacionada con el transporte en la empresa. Dicho sistema presenta algunas deficiencias pues parte de la información se gestiona de forma manual provocando en ocasiones pérdida de la misma. Por tal motivo el objetivo de este trabajo es desarrollar un módulo para la gestión de la información que se genera del transporte dentro de la empresa. La solución propuesta fue realizada con el framework Odoon utilizado para desarrollar sistemas que se encargan de la planificación de recursos empresariales, y siguiendo la metodología de desarrollo AUP en su versión UCI. Se realizaron pruebas funcionales aplicando el método de caja negra y pruebas de aceptación con el objetivo de examinar exhaustivamente el sistema. Como aporte de esta investigación se encuentra el desarrollo de un módulo para la gestión de la información relacionada con el transporte contando con funcionalidades como: actualización de documentos oficiales, gestión del vehículo, chofer y control de las reparaciones, los gastos, las multas, las hojas de ruta, los anexos 1 además lleva un control de trazas por entidad.

Palabras clave:

Transporte, gestión de información, Odoon

Índice

Contenido

INTRODUCCIÓN	1
CAPÍTULO 1: FUNDAMENTACIÓN TEÓRICA DE LA INVESTIGACIÓN	5
1.1 INTRODUCCIÓN.....	5
1.2 CONCEPTOS ASOCIADOS A LA INVESTIGACIÓN.....	5
1.3 ERP	6
1.4 ANÁLISIS DE OTROS SISTEMAS SIMILARES	7
1.4.1 <i>Sistemas similares a nivel internacional</i>	7
1.4.2 <i>Sistemas similares a nivel nacional</i>	10
1.4.3 <i>Sistemas similares en la UCI</i>	11
1.5 METODOLOGÍA DE DESARROLLO DE SOFTWARE	12
1.6 LENGUAJE DE MODELADO	16
1.7 HERRAMIENTA CASE.....	17
1.8 LENGUAJES Y TECNOLOGÍAS DE PROGRAMACIÓN.....	18
1.8.1 <i>Lenguajes del lado del servidor</i>	18
1.8.2 <i>Framework de desarrollo</i>	19
1.8.3 <i>Entorno de desarrollo</i>	21
1.8.4 <i>Sistema Gestor de Base de Datos</i>	22
1.9 CONCLUSIONES PARCIALES	23
CAPÍTULO 2: DESCRIPCIÓN DE LA SOLUCIÓN PROPUESTA	24
2.1 INTRODUCCIÓN.....	24
2.2 DESCRIPCIÓN DEL SISTEMA.....	24
2.3 REQUISITOS	24
2.3.1 <i>Requisitos funcionales</i>	25
2.3.1.1 <i>Descripción de requisitos funcionales</i>	27
2.3.2 <i>Requisitos no funcionales</i>	34
2.4 DISEÑO Y ARQUITECTURA DE LA SOLUCIÓN PROPUESTA.....	35
2.4.1 <i>Estilo y patrones</i>	35
2.5 MODELO CONCEPTUAL.....	42
2.6 DIAGRAMA DE CLASES Y SECUENCIA DEL DISEÑO	43
2.7 MODELO DE DATOS	44
2.8 CONCLUSIONES PARCIALES.....	46
CAPÍTULO 3: IMPLEMENTACIÓN Y PRUEBA	47
3.1 INTRODUCCIÓN.....	47
3.2 ESTÁNDAR DE CODIFICACIÓN	47
3.3 DIAGRAMA DE COMPONENTES.....	51
3.4 PRUEBAS DE SOFTWARE.....	53
3.5 CONCLUSIONES PARCIALES	62
CONCLUSIONES GENERALES	63

RECOMENDACIONES:	64
BIBLIOGRAFÍA REFERENCIADA	65

Introducción

El avance alcanzado por las Tecnologías de la Información y las Comunicaciones (TIC), soporta numerosos cambios que se evidencian a gran escala en el desarrollo del mundo empresarial. Actualmente existen varias soluciones informáticas para la gestión de los procesos que se desarrollan en las empresas, estas permiten la disminución en gran medida de los problemas que puedan estar presente en las instituciones. A estos sistemas se les denominan Sistemas Integrales de Gestión, su función principal está centrada en integrar y manejar la mayoría de los negocios que se asocian a las operaciones de producción y distribución de bienes o servicios.

En Cuba, algunas empresas han emprendido la labor de informatizar paulatinamente todos sus procesos con el fin de lograr minimizar las pérdidas económicas y maximizar las ganancias, poniendo en práctica el Perfeccionamiento Empresarial. La empresa Agencia Transitaria de Cargas (en lo adelante TRANSCARGO) se encuentra inmersa en estas transformaciones. TRANSCARGO es una empresa pionera en Cuba del transporte de mercancías puerta a puerta (*door to door*), la cual, hace que el cliente se despreocupe, delegando en terceros todos los trámites necesarios para que las mercancías lleguen a su destino final. Esta permite brindar servicios relacionados con el transporte nacional e internacional de cargas comerciales y no comerciales, tanto, en la esfera marítima como aérea.

Dentro de TRANSCARGO existe una estructura organizativa compuesta por varias Unidades Empresarial de Base (en lo adelante UEB) en dependencia de los servicios que brinden. Una de ellas es la de UEB Servicios Logísticos a clientes que se encarga de:

- ✓ establecer un control del transporte, como complemento de la actividad transitaria que desarrolla la organización.
- ✓ el aseguramiento de los suministros de todo tipo.
- ✓ la gestión de portadores energéticos encargados de controlar la información del combustible, electricidad y agua.
- ✓ control de los procesos de abastecimiento de los servicios básicos.
- ✓ control del mantenimiento y las inversiones.

Los procesos relacionados con el transporte se llevan a cabo de forma manual y en ocasiones apoyados por el uso de herramientas ofimáticas como documentos Word¹ y tablas Excel². También se cuenta con un sistema realizado en Microsoft Access³ que maneja algunos elementos relacionados con el transporte, pero este presenta limitaciones como:

- ✓ no controla la explotación de los vehículos, impidiendo sacar estadísticas de mantenimiento sobre cada uno de ellos.
- ✓ no maneja las actualizaciones de los documentos oficiales tanto del vehículo como del chofer.
- ✓ no realiza una correcta gestión de los choferes y de los vehículos.
- ✓ no controla información sobre las reparaciones del vehículo, las multas, los gastos, las hojas de ruta.
- ✓ no permite que se conecte más de un usuario a la aplicación al mismo tiempo provocando sobrecarga en el sistema.
- ✓ el equipo de trabajo que lo desarrolló ya no se encuentra en la empresa para darle solución a estos problemas.

Debido a estas cuestiones la gestión de la información que se genera del transporte se hace engorroso y lento, pues la misma se encuentra en formato duro, almacenada en archivos físicos no centralizados, en ellos la información está duplicada y en ocasiones existe pérdida de la misma. La empresa le solicitó a la Universidad de las Ciencias Informáticas (en lo adelante UCI), su ayuda para solucionar los problemas que presenta referente a la gestión, control y planificación de toda la información relacionada con el transporte en dicha entidad de una forma rápida y segura.

A partir de la situación problemática descrita anteriormente se propone como **problema a resolver**: ¿Cómo contribuir a la gestión de la información que se genera del transporte dentro de la empresa TRANSCARGO? Con el fin de solucionar el problema, se define como **objeto de estudio** de la presente investigación: la gestión de la información del transporte.

Enfocando el **campo de acción en**: la gestión de la información automatizada del transporte.

¹ es una [aplicación informática](#) orientada al [procesamiento de textos](#). Fue creado por la empresa [Microsoft](#), y viene integrado en el [paquete ofimático](#) denominado [Microsoft Office](#). <http://concepto.de/word/>

² es una aplicación distribuida por la suite de oficina [Microsoft Office](#), que se caracteriza por ser un software de [hojas de cálculo](#), utilizado en tareas financieras y contables. Excel permite a los usuarios elaborar tablas y formatos que incluyan cálculos matemáticos mediante fórmulas. <http://definicion.de/excel/>

³ un sistema de gestión de bases de datos incluido en el paquete ofimático denominado Microsoft Office. http://www.lalila.org/tutoriales/tutorial_Access97.pdf

Teniendo en cuenta el problema a resolver se define como **objetivo general**: desarrollar un módulo para la gestión de la información que se genera del transporte dentro de la empresa TRANSCARGO.

Complementando el objetivo general se definen los siguientes **objetivos específicos**:

- ✓ Construir los referentes teóricos relacionando los aspectos fundamentales que sustentan la investigación, mediante los cuales se consulta, extrae y recopila la información relevante sobre el problema a investigar.
- ✓ Caracterizar el proceso de gestión de información del transporte de la institución.
- ✓ Diseñar la propuesta de solución que permita contribuir a la gestión de la información que se genera del transporte.
- ✓ Implementar la propuesta de solución utilizando las herramientas, lenguajes y tecnologías definidas.
- ✓ Validar la solución desarrollada mediante pruebas funcionales y pruebas de aceptación.

Para guiar la investigación se define como **Hipótesis**: el desarrollo de un módulo para la gestión del transporte dentro de la empresa TRANSCARGO, permitirá contribuir a la gestión de la información del transporte en la institución.

Métodos investigativos

Para realizar la investigación se utilizan los métodos teóricos y empíricos que a continuación se relacionan:

Métodos teóricos

- ✓ Analítico-Sintético: se empleó para identificar, analizar y seleccionar los conceptos y definiciones más importantes relacionadas con el tema, permitiendo generar una propuesta adecuada a la situación planteada.
- ✓ Análisis Histórico-Lógico: fue utilizado para el estudio de otras soluciones o aplicaciones similares con la investigación, así como las metodologías de desarrollo, lenguajes de programación y framework (marcos de trabajo) a utilizar en el desarrollo del Módulo Transporte.
- ✓ Hipotético-Deductivo: se empleó para la elaboración de la problemática a resolver en la investigación y la propuesta de nuevas soluciones.
- ✓ Análisis documental: se utilizó para realizar consultas a la literatura especializada en el tema abordado en la presente investigación.

Métodos empíricos

- ✓ Observación: se recurrió para el análisis de los procedimientos que se realizan actualmente en la gestión del transporte, como es el caso de la herramienta desarrollada en Microsoft Access.
- ✓ Entrevista en profundidad: se utilizó para recopilar, analizar y conformar la información a través del personal que labora en los procesos relacionados con el transporte, para establecer criterios en vista al desarrollo del sistema, garantizando así, el perfeccionamiento de la solución propuesta relacionada con el funcionamiento del transporte interno en la empresa.

El contenido de este documento está estructurado en tres capítulos, conclusiones, recomendaciones, bibliografía y anexos.

Capítulo 1: Fundamentación Teórica de la Investigación.

Se realiza un análisis de los sistemas actuales destinados a la gestión del transporte. Se detallan las tendencias y tecnologías actuales, metodologías y herramientas para el desarrollo web, justificando la selección de cada una de ellas, así como los conceptos teóricos necesarios para comprender los temas tratados en el resto del documento.

Capítulo 2: Descripción de la solución propuesta.

Se exponen las características del sistema a implementar a partir de un levantamiento de las necesidades que debe cumplir la solución, definiéndose los requisitos funcionales y no funcionales de la misma. Se generan los artefactos correspondientes a la metodología de desarrollo a utilizar, que incluyen los diagramas pertinentes, posibilitando el modelado de la solución de manera visual.

Capítulo 3: Implementación y pruebas.

Este capítulo abarca todo lo relacionado con la implementación del Módulo Transporte, reflejando aspectos relacionados con la construcción de la solución propuesta, se evidencian los patrones de programación y los estándares de codificación. Además se definen las pruebas y se elaboran los casos de pruebas a realizarle al software, para así verificar la integridad del mismo y el cumplimiento de los requisitos definidos por el cliente. Finalmente se realiza la validación de la solución propuesta.

Capítulo 1: Fundamentación Teórica de la Investigación

Capítulo 1: Fundamentación Teórica de la Investigación

1.1 Introducción

La base teórica del presente trabajo está sustentada sobre los sistemas de gestión que sirven de punto de partida para la investigación. Es fundamental para tener una visión de la solución propuesta, el dominio de algunos conceptos esenciales y soluciones existentes relacionados con la gestión del transporte. Además el estudio de las herramientas y tecnologías a utilizar durante el desarrollo de la propuesta de solución, sirven de apoyo para que toda la información recopilada oriente y guie conceptualmente el desarrollo de la investigación.

1.2 Conceptos asociados a la investigación

El término sistema de gestión es un vocablo que se puede encontrar en diferentes fuentes de información con disímiles enfoques en dependencia del área de conocimiento a la que se refiera. Por ello se hace necesario indagar en las palabras que conforman la expresión Sistema de Gestión del Transporte.

Sistema

Proviene del latín systema, es un módulo ordenado de elementos que se encuentran interrelacionados y que interactúan entre sí.(1) Otras fuentes de información lo describen como un conjunto de elementos relacionados entre sí y que funcionan como un todo. (2).

Analizando las definiciones anteriores se puede afirmar que un sistema es un conjunto de elementos relacionadas entre sí ordenadamente para lograr un fin determinado.

Gestión

Procede del latín gestio, el concepto de gestión hace referencia a la acción y consecuencia de administrar o gestionar algo. Por otro lado gestionar es llevar a cabo diligencias que hacen posible la realización de una operación comercial o de un anhelo cualquiera (3), es además un conjunto de acciones u operaciones relacionadas con la administración y dirección de una organización.(4)

De acuerdo a estos conceptos se puede plantear que la gestión es la capacidad de una institución para definir, alcanzar y evaluar sus propósitos, administrando y dirigiendo de manera adecuada sus recursos disponibles.

Capítulo 1: Fundamentación Teórica de la Investigación

Un **Sistema de Gestión** es aquel que sirve de ayuda para lograr las metas y objetivos de una organización, a través de una serie de estrategias, entre las cuales se encuentran la optimización de los procesos y el enfoque basado en la gestión y la disciplina.(5) Es decir, un Sistema de Gestión involucra a un conjunto de etapas integradas en un proceso continuo que funcionan hasta lograr su mejora definitiva.

Un **Sistema de Gestión de Transporte** o también conocido como (*TM-System*) es una herramienta para los agentes de transporte. Con el sistema de gestión de transporte, el agente puede tramitar y trabajar las órdenes de transporte que recibe de la empresa cargadora o productora.(6) Con ello, los viajes y rutas pueden planearse y organizarse mejor. De esta forma, podrá ver en cualquier momento la disponibilidad de los vehículos. Este mejora el funcionamiento del área de transporte de las instituciones ya sea de transporte de mercancías o el capital humano de la empresa.

A partir de la necesidad que presentan las empresas para gestionar su información surgen sistemas para la planificación de recursos empresariales los mismos son conocidos como Sistemas ERP.

1.3 ERP

ERP (Planeación de Recursos Empresariales), por sus siglas en inglés *Enterprise Resource Planning*. Los ERP son un tipo de software que permite a las empresas controlar la información que se genera en cada departamento y cada nivel de la misma. En el ámbito empresarial, tener a la mano la información necesaria puede significar una ganancia o una pérdida monetaria. En las últimas décadas, han aparecido y evolucionado los sistemas de planeación de los recursos empresariales para ayudar en este sector, estos son mejores conocidos como ERP.(7)

El fin de los ERP es integrar los departamentos, donde antes había un sistema de información especializado para cada órgano de la empresa. Estos son capaces de generar una base de datos limpia, donde se gestione la información en tiempo real y se pueda obtener los datos requeridos en el momento que se desee.(7)

¿Cómo funciona un ERP?

Como cualquier software, funciona en base a una plataforma de programación, seguida por la gestión de disímiles bases de datos correspondientes a los distintos departamentos que se deseen integrar. Los

Capítulo 1: Fundamentación Teórica de la Investigación

sistemas ERP se organizan por medio de módulos, los cuales se conectan a distintas bases de datos, según lo que se requiera para cada departamento. Existen 2 tipos de ERP: los propietarios y los de código abierto. Los propietarios son hechos por empresas con fines de lucro que venden su software y los implementan a las empresas que lo soliciten a un elevado costo, para poder utilizarlos se necesita obtener una licencia más el costo de la implementación del software en la empresa. (7)

Ventajas de los ERP

- ✓ Como se ha mencionado anteriormente, la principal ventaja de los ERP es la gestión en tiempo real de la información, una ventaja que las empresas agradecen mucho por su fuerte interacción con la logística de información y productos, la cadena de abastecimiento, estadísticas financieras, y otras áreas que utilizan información que cambia constantemente. (7)
- ✓ La correcta implementación de los ERP repercute en el aumento de productividad de todos los departamentos, así como en el mejor aprovechamiento del tiempo, donde antes se necesitaba tiempo para llevar un informe de un departamento a otro, ahora ese tiempo es utilizado en otras funciones.

1.4 Análisis de otros sistemas similares

En el mundo, el desarrollo de las tecnologías va aparejado al incremento de sistemas para la informatización de los procesos en las distintas esferas. La necesidad de comprender y analizar la trascendencia de los sistemas informáticos⁴ relacionados con la gestión del transporte, en el ámbito internacional y nacional, es un factor clave para la realización de un nuevo sistema.

1.4.1 Sistemas similares a nivel internacional

En diferentes países se han desarrollado varias aplicaciones que gestionan de manera automatizada el transporte. A continuación se exponen algunas de las soluciones más importantes encontradas a través de la revisión bibliográfica realizada.

ERP Transporte TM (transport management), gestión del transporte (<http://www.imagine-cs.com/>)

⁴ sistema informático es un conjunto de partes que funcionan relacionándose entre sí con un objetivo preciso. Sus partes son: hardware, software y las personas que lo usan. Tomado de <http://www.alegsa.com.ar/Dic/sistema%20informatico.php>

Capítulo 1: Fundamentación Teórica de la Investigación

Esta solución tecnológica permite que las empresas de transporte tengan un enfoque adaptado a la situación actual del mercado, desechando antiguos sistemas de trabajo obsoletos, que no le permiten desarrollar todo el potencial que su empresa puede alcanzar. Es un software para el control y planificación del transporte, que gestiona la recepción y envío de cargas, los almacenes, los proveedores, el control de flotas entre otros.
(8)

Esta solución constituye una poderosa herramienta que brinda las siguientes funcionalidades:(8)

- ✓ Gestión grupaje, carga completa, contenedores y agencias.
- ✓ Expedientes / hojas de ruta múltiples.
- ✓ Liquidación a proveedores y conductores.
- ✓ Gestión de vehículos propios y ajenos.
- ✓ Control de flotas.
- ✓ Control recogidas / entregas.
- ✓ Emisión automática de ofertas.
- ✓ Tarifas personalizadas.
- ✓ Aplicación automática de precios.
- ✓ CRM.
- ✓ Sistemas de acceso remoto.
- ✓ PiD (presentación inteligente de datos).
- ✓ Data Warehouse (almacén de datos).

Software ERP empresas Transporte y Logística (<http://www.galdon.com>)

Solución ERP, software específico para la gestión de empresas de transporte de mercancía y logística. En Galdon Software⁵ se ha creado una completa aplicación para gestión de flotas de vehículos. Esta aplicación está desarrollada con la finalidad de integrar todas las áreas del sector de transporte y logística, planificar y gestionar de modo integrado los departamentos de Tráfico, Facturación, Administración y Finanzas, Gestión de Flotas y Almacenaje.

⁵ es una empresa de ingeniería que nace en 1988 con el objetivo de ofrecer soluciones integrales en el ámbito del software de gestión a empresas e instituciones, en entornos multiusuario. <http://www.galdon.com>

Capítulo 1: Fundamentación Teórica de la Investigación

Su interfaz intuitiva permite trabajar con mayor eficacia y rapidez, aumentando la productividad de la empresa. El ERP de gestión de flotas también puede incluir un módulo para movilidad en tablets o móviles para gestión de aviso y entrega de cargas.(9)

Esta solución constituye una herramienta que brinda las siguientes funcionalidades:(9)

- ✓ Multialmacén, multiempresa y multiejercicio.
- ✓ Módulo de tráfico.
- ✓ Control de pedidos de clientes.
- ✓ Disponibilidad de vehículos (propios y/o subcontratados).
- ✓ Gestión de incidencias de pedidos y/o transporte, posibilidad de realizar filtros por origen/destino de carga, fecha y/o vehículo.
- ✓ Estadísticas control de gasoil.
- ✓ Módulo de Gestión de Taller y Recambios.
- ✓ Control de órdenes de reparación (taller propio o externo).
- ✓ Control de stock de recambios.
- ✓ Ventas y compras de recambios.
- ✓ Facturación de órdenes de taller
- ✓ Control de tiempos de operarios.
- ✓ Módulo Gestión de Gastos de Flotas.
- ✓ Control de gastos por conductor, repostajes o kilómetros realizados.
- ✓ Informes de beneficios por vehículo, conductor, cliente.
- ✓ Creación y gestión de eventos de caducidad de tarjetas de transporte, tacógrafos.
- ✓ Optimización de rutas.
- ✓ Sistema de Gestión de Almacén (SGA).
- ✓ Asignación automática de ubicación en órdenes de entrada/salida.
- ✓ Gestión documental en la nube.
- ✓ Posibilidad de conexión con software de *Microsoft Map Point* y con *Google Maps*.
- ✓ Previsiones de tesorería, estadísticas y cuadro de mando.
- ✓ Contabilización automática de los documentos generados desde gestión.

Capítulo 1: Fundamentación Teórica de la Investigación

1.4.2 Sistemas similares a nivel nacional

gCARS: Sistema de gestión de vehículos (<http://www.datys.cu>)

Producto diseñado para el control de vehículos y equipos tecnológicos, en cualquier tipo de empresa u organización. Sistema capaz de controlar los recursos de una empresa, logrando una eficiente administración, el monitoreo constante de los procesos, reducir costos y/o gastos y lograr ahorro. Cuenta con un conjunto de funcionalidades y reportes, orientados a mejorar la gestión, tanto del parque automotriz, como del combustible y los recursos complementarios.(10)

A través, de sus funcionalidades, garantiza la gestión de:(10)

- Vehículos y equipos tecnológicos
 - ✓ Control histórico de los cambios realizados en el vehículo en cuanto a la matrícula, centro de costo, tipo de vehículo, número de circulación, de motor, de carrocería.
 - ✓ Control de las hojas de ruta.
 - ✓ Control de las licencias de circulación y licencias operativas.
 - ✓ Control de las revisiones técnicas y mantenimientos.
 - ✓ Control de los gastos (por concepto de taller, de renta, o cualquier tipo de gastos que defina el usuario).
- Tarjetas pre-pagadas y bonos de combustible
 - ✓ Control de la entrega y devolución de tarjetas y bonos.
 - ✓ Eliminación masiva de la entrega y devolución de tarjetas y bonos.
 - ✓ Control de las recargas.
 - ✓ Permite que las tarjetas tengan varios servicios.
 - ✓ Adjudicaciones de saldos.
- Combustibles y lubricantes
 - ✓ Control del consumo de combustibles y lubricantes tanto por vehículos como por equipos.
 - ✓ Permite crear planes de combustible teniendo en cuenta el consumo real de la entidad.
 - ✓ Control de las asignaciones realizadas al proveedor de combustible de la entidad.

Capítulo 1: Fundamentación Teórica de la Investigación

- ✓ Permite además la adjudicación de combustible.
- Recursos complementarios: agregados, accesorios
 - ✓ Control histórico de las asignaciones que fueron desarrolladas en el sistema.
- Sistema de Auditoría
 - ✓ Control de las acciones realizadas por los usuarios de la aplicación que tengan una incidencia directa en la Base de Datos, como son: insertar, modificar y eliminar.

1.4.3 Sistemas similares en la UCI

Sistema de Gestión de Mantenimiento Vehicular v1.0 (SGMV): aplicación realizada en año 2012 por la Universidad de las Ciencias Informáticas (UCI) para el Cuerpo de la Policía Nacional Bolivariana (CPNB). Este sistema incluye la automatización de los procesos de la Administración de Inspecciones Técnicas y Accidentes, a través de las cuales se lleva el control de los accidentes en los que incurren las unidades policiales y las inspecciones técnicas que se le realizan a las mismas. Esta solución permite gestionar accidentes de forma general de todas las unidades policiales del CPNB, además de gestionar los accidentes de una unidad específica. Posee la funcionalidad de registrar las inspecciones técnicas que se realizan a las unidades policiales del CPNB y finalmente emite un informe con el resultado de la inspección técnica.(11) Esta solución no está realizada en ERP pero sirve de análisis para el diseño de la propuesta de solución.

Conclusión parcial de los sistemas similares

Luego de haber realizado una descripción de los sistemas similares a la solución propuesta, se llega a la conclusión de que ninguno puede ser utilizado como solución a la problemática planteada. Por lo tanto teniendo en cuenta que las soluciones similares solo cumplen parcialmente algunas de las exigencias, se decide utilizarlas como punto de referencia a la hora de elaborar la propuesta de solución. Algunas de las razones por las que no cumple son:

- ✓ Son sistemas muy grandes compuesto por varios módulos, los cuales no se pueden integrar a la solución propuesta.
- ✓ Ninguno de estos sistemas permite gestionar los choferes, solo en gCARS se lleva un control de algunos de los documentos legales del vehículo, por tanto tampoco controlan la actualización de los documentos oficiales relacionados con la entidad.
- ✓ Estas soluciones se centran en la recepción y envío de cargas relacionadas con su negocio y no en el transporte interno de la empresa.

Capítulo 1: Fundamentación Teórica de la Investigación

- ✓ Algunos de estos sistemas realizan reparaciones del vehículo, pero no solucionan de forma completa las necesidades del transporte.
- ✓ Ninguno controla la asignación de choferes a los vehículos imposibilitando un uso correcto de los recursos humanos de la institución.

Sin embargo el análisis de las soluciones similares permitió un mayor grado de comprensión del funcionamiento del transporte en una institución. Además como punto de partida para la confección de la propuesta de solución se tomaron elementos utilizados por estos sistemas como fueron: el control de las hojas de ruta, las reparaciones del vehículo, la gestión de documentos oficiales como licencia operativa y licencia de circulación y por último la gestión del vehículo.

1.5 Metodología de desarrollo de software

Las metodologías de desarrollo surgen por la necesidad de la industria del software de agilizarse y robustecerse al mismo tiempo. Existen metodologías tradicionales y metodologías ágiles, las primeras están pensadas para el uso exhaustivo de documentación durante todo el ciclo del proyecto. Las segundas se centran en la capacidad de respuesta frente a los cambios, la confianza en las habilidades del equipo y en mantener una buena relación con el cliente.

Las metodologías tradicionales (formales) se focalizan en documentación, planificación y procesos. Además centran su atención en llevar una documentación exhaustiva de todo el proyecto y en cumplir con un plan de proyecto, definido todo esto, en la fase inicial del desarrollo del proyecto. (12), (13)

RUP

RUP (Proceso unificado de desarrollo), por sus siglas en inglés *Rational Unified Process*. Está pensado para adaptarse a cualquier proyecto. Esta metodología divide el proceso de desarrollo en ciclos, obteniendo una versión del software al final de cada ciclo. Estas se dividen en fases, las cuales concluyen con un hito donde se debe tomar una decisión importante.

Las fases son las siguientes: (12)

- ✓ Inicio.
- ✓ Elaboración.
- ✓ Construcción.
- ✓ Transición.

Capítulo 1: Fundamentación Teórica de la Investigación

A continuación se presentan los flujos de trabajo que comprende RUP:

- ✓ Modelado del negocio.
- ✓ Análisis de requisitos.
- ✓ Análisis y diseño.
- ✓ Implementación.
- ✓ Prueba.
- ✓ Despliegue.
- ✓ Gestión de configuración y cambios.
- ✓ Gestión del proyecto.
- ✓ Gestión del entorno.

MSF

Marco de soluciones en Microsoft, en inglés *Microsoft Solution Framework*, es un compendio de las mejores prácticas en cuanto a administración de proyectos se refiere. Más que una metodología rígida de administración de proyectos, MSF es una serie de modelos que pueden adaptarse a cualquier proyecto de tecnología de información.

Todo proyecto es separado en cinco fases principales: (12)

- ✓ Visión y Alcance.
- ✓ Planificación.
- ✓ Desarrollo.
- ✓ Estabilización.
- ✓ Implantación

Las metodologías ágiles permiten incorporar cambios con rapidez en el desarrollo de software. En muchas ocasiones, los modelos de gestión tradicionales no sirven para afrontar un reto que hoy en día resulta fundamental: incorporar cambios con rapidez y en cualquier fase del proyecto. Estas se envuelven en un enfoque para la toma de decisiones en los proyectos de software, que se refiere a métodos de ingeniería del software basados en el desarrollo iterativo e incremental, donde los requisitos y soluciones evolucionan con el tiempo según la necesidad del proyecto. Así el trabajo es realizado mediante la colaboración de equipos auto-organizados y multidisciplinarios, inmersos en un proceso compartido de toma de decisiones a corto plazo.(12),(14)

Capítulo 1: Fundamentación Teórica de la Investigación

XP

Programación extrema, en inglés *Extreme Programming*, es la metodología más destacada de los procesos ágiles de desarrollo de software, se diferencia de las metodologías tradicionales principalmente en que pone más énfasis en la adaptabilidad que en la previsibilidad. Los defensores de XP consideran que los cambios de requisitos sobre la marcha son un aspecto natural, inevitable e incluso deseable del desarrollo de proyectos. Creen que ser capaz de adaptarse a los cambios de requisitos en cualquier punto de la vida del proyecto es una aproximación mejor y más realista que intentar definir todos los requisitos al comienzo del proyecto e invertir esfuerzos después en controlar los cambios en los requisitos.(12)

AUP

Proceso ágil unificado, por sus siglas en inglés *Agil Unified Process*, es una versión simplificada del desarrollo de software basado en el *Rational Unified Process*, basado en disciplinas y entregables incrementales con el tiempo.

Las disciplinas de AUP son:(12),(13)

- ✓ Modelado
- ✓ Implementación
- ✓ Prueba
- ✓ Despliegue
- ✓ Administración de la configuración
- ✓ Administración o gerencia del proyecto
- ✓ Entorno

AUP-UCI

Proceso ágil unificado, por sus siglas en inglés *Agil Unified Process*, en su versión UCI. Esta metodología propone cuatro fases (inicio, elaboración, construcción, transición) para el ciclo de vida de los proyectos de la UCI mantiene la fase de inicio, pero modificando el objetivo de la misma, se unifican las restantes tres fases de AUP en una sola, a la que se llamó ejecución y se agrega la fase de cierre.(15) Para una mayor comprensión ver el [Anexo 1](#).

AUP propone siete disciplinas (modelado, implementación, prueba, despliegue, gestión de configuración, gestión de proyecto y entorno), se decide para el ciclo de vida de los proyectos de la UCI tener ocho disciplinas, pero a un nivel más atómico que el definido en AUP. Los flujos de trabajos: modelado de negocio,

Capítulo 1: Fundamentación Teórica de la Investigación

requisitos y análisis y diseño en AUP están unidos en la disciplina modelo, en la variación para la UCI se consideran a cada uno de ellos disciplinas. Se mantiene la disciplina Implementación, en el caso de prueba se desagrega en tres disciplinas: pruebas internas, de liberación y aceptación y la disciplina despliegue se considera opcional. Las restantes tres disciplinas de AUP asociadas a la parte de gestión para la variación UCI se cubren con las áreas de procesos que define CMMI-DEV (Capacidad de Integración de Madurez del Modelo, por su significado en español) v1.3 para el nivel 2, serían CM (Gestión de la configuración), PP (Planeación de proyecto) y PMC (Monitoreo y control de proyecto).(15). Para una mayor comprensión ver el [Anexo 2](#).

AUP propone nueve roles (administrador de proyecto, ingeniero de procesos, desarrollador, administrador de BD(Base Datos), modelador ágil, administrador de la configuración, *Stakeholder*(experto en dominio), administrador de pruebas, probador). Se decide para el ciclo de vida de los proyectos de la UCI tener once roles, manteniendo algunos de los propuestos por AUP y unificando o agregando otros.(15) Para una mayor comprensión ver el [Anexo 3](#).

En la variación de la metodología AUP-UCI, la disciplina Modelado de negocio propone tres variantes a utilizar en los proyectos (Casos de Uso del Negocio (CUN), Descripción de Proceso de Negocio (DPN) y Modelo Conceptual (MC). Además existen tres formas de encapsular los requisitos Casos de Uso del Sistema (CUS), Historias de usuario (HU) y Descripción de requisitos por proceso (DRP), agrupados en cuatro escenarios condicionados por el Modelado de negocio. A partir de las variantes anteriores se condicionan cuatro escenarios para modelar el sistema en la disciplina Requisitos, manteniendo en dos de ellos el MC, quedando de la siguiente forma:(15)

Escenario No 1:

Proyectos que modelen el negocio con CUN solo pueden modelar el sistema con CUS.

$CUN + MC = CUS$

Escenario No 2:

Proyectos que modelen el negocio con MC solo pueden modelar el sistema con CUS.

$MC = CUS$

Escenario No 3:

Proyectos que modelen el negocio con DPN solo pueden modelar el sistema con DRP.

Capítulo 1: Fundamentación Teórica de la Investigación

DPN + MC = DRP

Escenario No 4:

Proyectos que no modelen negocio solo pueden modelar el sistema con HU.

HU

El desarrollo de la propuesta de solución se enmarca en el escenario 3 aunque no exista proceso de negocio en la presente investigación, pero como es el seleccionado por el proyecto la investigación tiene que seguir las líneas de desarrollo para que exista homogeneidad en el trabajo en equipo.

Después de analizar las metodologías planteadas anteriormente se decidió utilizar AUP-UCI como metodología de desarrollo debido a que:

- ✓ Es factible por su habilidad de responder a los cambios que puedan surgir a lo largo del proyecto (cambios en los requisitos, en la tecnología, etc.).
- ✓ La metodología AUP es la adopción de muchas de las técnicas ágiles de XP y otros procesos ágiles que mantiene RUP.
- ✓ En relación a RUP, AUP resulta ser un proceso muy pesado y en relación a XP resulta ser un proceso muy simplificado.
- ✓ Además, es la metodología utilizada por el proyecto.

1.6 Lenguaje de modelado

Un lenguaje de modelado está compuesto por vistas, diagramas, elementos de modelo, los símbolos utilizados en los modelos y un conjunto de mecanismos generales o reglas que indican cómo utilizar los elementos. Las reglas son sintácticas, semánticas y pragmáticas.(16) Se trata de un lenguaje gráfico para construir, documentar, visualizar y especificar un sistema de software. Posee la riqueza suficiente como para crear un modelo del sistema, pudiendo modelar los procesos de negocios, funciones, esquemas de bases de datos, expresiones de lenguajes de programación.

UML

UML (Lenguaje de Modelado Unificado), por sus siglas en inglés *Unified Modeling Language*, este es la sucesión de una serie de métodos de análisis y diseño orientados a objetos que aparecen a fines de los años 80 y principios de los 90. UML es llamado un lenguaje de modelado, no un método.(17),(16)

Capítulo 1: Fundamentación Teórica de la Investigación

UML es un lenguaje de construcción de modelos para visualizar, especificar, construir y documentar los artefactos de un sistema que involucra una gran cantidad de software, con los que se construyen mayormente sistemas orientados a objetos. Con él se pueden modelar conceptos y esquemas de base de datos. Se puede aplicar en el desarrollo de software generando variedad de modelos que son utilizadas por diferentes metodologías de desarrollo de software, pero no especifica en sí mismo qué metodología utilizar.(17)

UML incrementa la capacidad de lo que se puede hacer con otros métodos de análisis y diseño orientados a objetos. Los autores de UML apuntaron también al modelado de sistemas distribuidos y concurrentes para asegurar que el lenguaje maneje adecuadamente estos dominios. (17)

Para el desarrollo del módulo se utilizará el UML, como lenguaje con que se modelarán los artefactos que se creen en el proceso de desarrollo del software. Se elige este lenguaje de modelado porque este se utiliza para modelar los procesos de cualquier sistema además de que es el lenguaje que utiliza el proyecto.

1.7 Herramienta CASE

Para seleccionar la herramienta CASE (Ingeniería de Software Asistida por Computación, por sus siglas en inglés *Computer Aided Software Engineering Visual Paradigm*) que se empleará en el modelado de los artefactos del módulo se tendrá en cuenta que para el modelado utilice UML, que es el lenguaje seleccionado y además, se encuentra bajo licencias libres.

Visual Paradigm

Es una herramienta CASE que 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. Ha sido concebida para soportar el ciclo de vida completo del proceso de desarrollo del software a través de la representación de todo tipo de diagramas. Constituye una herramienta privada disponible en varias ediciones, cada una destinada a satisfacer diferentes necesidades. Fue diseñado para una amplia gama de usuarios interesados en la construcción de sistemas de software de forma fiable a través de la utilización de un enfoque Orientado a Objetos.(18)

La herramienta Visual Paradigm 8.0 se empleará para especificar y construir los diversos artefactos ya que

Capítulo 1: Fundamentación Teórica de la Investigación

hasta esa versión es libre de ahí en adelante es privada. Además Visual Paradigm permite aumentar la calidad del software, a través de la mejora de la productividad en el desarrollo y mantenimiento del software. Aumenta el conocimiento informático de una empresa ayudando así a la búsqueda de soluciones para los requisitos. También permite la reutilización del software, portabilidad y estandarización de la documentación, además del uso de las distintas metodologías propias de la Ingeniería de Software. También es la herramienta modelado que utiliza el proyecto.

1.8 Lenguajes y tecnologías de programación

Fue necesario la selección de lenguajes y tecnologías de desarrollo de software, de acuerdo a las características del sistema que se quiere desarrollar. Para ello se hizo un análisis parcial de los lenguajes del lado del cliente y del lado del servidor. Seguidamente del framework de la capa de presentación. Fue de vital importancia el análisis y selección de los lenguajes y tecnologías de programación a utilizar para poder seguir las líneas de trabajo del proyecto.

1.8.1 Lenguajes del lado del servidor

XML

XML (Lenguaje de Etiquetado Extensible), por sus siglas en *inglés Extensible Mark-up Language*, es un meta-lenguaje que permite definir lenguajes de marcado adecuados a usos determinados. Permite representar información estructurada en la web (todos documentos), de modo que esta información pueda ser almacenada, transmitida, procesada, visualizada e impresa, por muy diversos tipos de aplicaciones y dispositivos.(19)

Después de realizar un estudio sobre el lenguaje XML se decide utilizarlo ya que este le proporciona al sistema una mejor velocidad de navegación, debido a que las páginas creadas y maquetadas correctamente se arman más rápido en los exploradores que si no lo están, ya sea porque hay errores de sintaxis o porque se usan elementos con un fin para el cual no fueron creados. También proporciona un mejor entendimiento del código principalmente para los diseñadores; pudiéndose actualizar fácilmente la web. Además, es el lenguaje que utiliza Odo para mediante su framework interpretar y elaborar las vistas.

Capítulo 1: *Fundamentación Teórica de la Investigación*

Python

Es un lenguaje de programación interpretado cuya filosofía hace hincapié en una sintaxis que favorezca un código legible. Se trata de un lenguaje de programación multiparadigma, ya que soporta orientación a objetos. Es un lenguaje interpretado, usa tipado dinámico y conteo de referencias para la administración de memoria, además, es multiplataforma.(20),(21)

Una característica importante de Python es la resolución dinámica de nombres; es decir, lo que enlaza un método y un nombre de variable durante la ejecución del programa (también llamado enlace dinámico de métodos).

Otro objetivo del diseño del lenguaje es la facilidad de extensión. Se pueden escribir nuevos módulos fácilmente en C o C++. Python puede incluirse en aplicaciones que necesitan una interfaz programable.(20)

Luego de analizar las características del lenguaje de programación del lado del servidor se decide utilizar Python ya que es el lenguaje en que se basa OdoO para la implementación de sus módulos. Además el pseudo-código natural de Python es una de sus grandes fortalezas ya que permite concentrarse en la solución del problema en lugar de la sintaxis, es decir el propio lenguaje.

1.8.2 Framework de desarrollo

OdoO

En su versión 8.0 es el marco de desarrollo utilizado para construir sistemas de gestión empresarial (ERP), de código abierto y sin coste de licencias que cubre las necesidades de las áreas de: Contabilidad y Finanzas, Ventas, Recursos Humanos, Compras, Proyectos, Almacenes, CRM (Customer Relationship Management por sus siglas en inglés y en español gestión de las relaciones con los clientes) y Fabricación.(22)

Es un sistema modular y escalable que dispone de módulos oficiales y módulos desarrollados por terceros, también permite realizar desarrollos propios para adaptarlos a las necesidades específicas de una empresa o sector.

Principales módulos:(23)

- ✓ Ventas
- ✓ Compras

Capítulo 1: Fundamentación Teórica de la Investigación

- ✓ Gestión financiera
- ✓ CRM
- ✓ Recursos humanos
- ✓ Gestión de almacenes
- ✓ Gestión de proyectos
- ✓ Fabricación
- ✓ Marketing

Principales características:(22)

- ✓ Es un sistema de código abierto: esto significa que cualquier módulo puede ser modificado o diseñado desde cero, adaptándolo a las necesidades de la empresa.
- ✓ multiplataforma: a través de un navegador web se puede acceder a la interfaz, independientemente del sistema operativo que se utilice.
- ✓ Fácil manejo: no son necesarios grandes conocimientos de informática para poder utilizar su sencilla interfaz.
- ✓ Posee una importante comunidad de desarrolladores, con más de 2000 módulos liberados disponibles que pueden combinarse entre sí.
- ✓ Integración con otras aplicaciones: visualización de documentación con formato .pdf, importación/exportación de documentos de *Microsoft Office* u *Open Office* y compatibilidad con *Google Maps*.
- ✓ Las vistas se definen en lenguaje basado en XML y Odoosobre la marcha convierte estas en una aplicación cliente-servidor o el cliente basado en web.
- ✓ Permite al desarrollador construir rápidamente aplicaciones con diferentes tipos de vista como son: árbol lista, calendario, Kanban y Gantt.
- ✓ Odoos está construido alrededor de Python:
 - Moderno, orientado a objetos.
 - Portátil (Linux, Windows, Mac, etc.).

Odoos ofrece, en un solo paquete el componente básico para la construcción de una aplicación de negocios(24):

- ✓ Mapeo objeto-relacional (ORM): Simplifica el acceso a la base de datos.

Capítulo 1: Fundamentación Teórica de la Investigación

- ✓ Sistema de Permiso: Completo sistema de permisos, menús, campos, acceso a datos, acciones comerciales.
- ✓ Multi-idioma / internacionalización: Unicode UTF-8, varios idiomas por base de datos (cambio de idioma), los datos temporales almacenados en UTC.
- ✓ Campos traducibles: Base de datos de los campos de texto pueden ser marcados como traducibles. Por ejemplo, el nombre del producto puede tener los siguientes nombres en inglés y francés "coche, voiture".
- ✓ Motor de informes: Odoos incorpora un motor de informes con la generación de PDF.
- ✓ Los servicios Web: Odoos ofrece sus servicios a través de XML-RPC y JSON (Javascript).
- ✓ Proporciona un marco integrado de aplicación para las pruebas.
- ✓ Trabaja con el sistema de gestión de base de datos de código abierto PostgreSQL.
 - Tamaño máximo de base de datos ilimitado
 - Máximo Tabla Size 32 TB
 - Máximo Fila Size 1.6 TB

Luego de realizar un análisis sobre el framework de desarrollo de software Odoos se decide utilizarlo ya que sobre este se realizan los sistemas ERP, los cuales permiten la planificación de los recursos empresariales de cualquier institución. Además brinda la posibilidad de gestionar recursos empresariales pertenecientes a los distintos departamentos de la empresa. También es la herramienta utilizada por el proyecto de desarrollo.

1.8.3 Entorno de desarrollo

PyCharm

PyCharm es un IDE (Entorno de Desarrollo Integrado) desarrollado por la compañía JetBrains, está basado en IntelliJ IDEA, el IDE de la misma compañía pero enfocado hacia Java y la base de Android Studio.(25),(26)

Dentro de todas las ventajas de PyCharm se pueden encontrar(25):

- ✓ Autocompletado, resaltador de sintaxis, herramienta de análisis y refactorización.
- ✓ Integración con frameworks web como: Django, Flask, Pyramid y Web2Py.
- ✓ Frameworks javascripts: jQuery y AngularJS.
- ✓ Debugger avanzado de Python y Javascript.

Capítulo 1: Fundamentación Teórica de la Investigación

- ✓ Integración con lenguajes de plantillas: Mako, Jinja2 y Django Template.
- ✓ Soporta entornos virtuales e intérpretes de Python 2.x, 3.x, PyPy, Iron Python y Jython.
- ✓ Compatibilidad con SQLAlchemy (ORM), Google App Engine y Cython.
- ✓ Soporte para modo VIM (Con plugin).
- ✓ Sistemas de control de versiones: Git, CVS y Mercurial.

Compatibilidad y licencia

PyCharm es multiplataforma, hay binarios para: Windows, Linux y Mac OS X. Existen dos versiones de PyCharm, una comunitaria y otra profesional con diferentes características.(25).

Luego de analizar algunos entornos de desarrollo se decide utilizar PyCharm ya que a partir de su versión 4.5.1 tiene licencia UCI, además de ser un IDE solamente para python. También porque es el entorno de desarrollo utilizado por el proyecto.

1.8.4 Sistema Gestor de Base de Datos

SGBD (Sistema de gestión de base de datos) o en inglés *Database Management System (DBMS)*, es una agrupación de programas que sirven para definir, construir y manipular una base de datos.(27),(28),(29)

PostgreSQL

PostgreSQL es un sistema de bases de datos objeto-relacional de código abierto. Cuenta con más de 15 años de desarrollo activo y una arquitectura probada que se ha ganado una sólida reputación por su fiabilidad, integridad de datos y la corrección. Se ejecuta en los sistemas operativos, Linux y Windows. Posee soporte completo para claves foráneas, uniones, vistas, *triggers* y procedimientos almacenados (en varios idiomas). También soporta almacenamiento de objetos binarios grandes, incluyendo imágenes, sonidos o video.(30)

Luego de realizar un estudio de SGBD se decide utilizar PostgreSQL puesto que consume muy pocos recursos tanto de CPU (control process unit, es decir unidad de control de procesos), como memoria. Además PostgreSQL está enfocado tradicionalmente en la fiabilidad e integridad de los datos. Finalmente porque es el SGBD utilizado por el proyecto.

Capítulo 1: Fundamentación Teórica de la Investigación

1.9 Conclusiones parciales

Al culminar el presente capítulo se puede concluir a partir de la bibliografía consultada que los sistemas analizados para la gestión del transporte interno poseen una serie de deficiencias, las cuales impiden que sean utilizados para realizar la informatización de los procesos de transporte en TRANSCARGO. Estos sistemas se van a tener en cuenta para el desarrollo de un sistema informático capaz de satisfacer las necesidades que actualmente presenta esta área en cada uno de los procesos que realiza. En la investigación se realizará el diseño e implementación de estos procesos utilizando como metodología de desarrollo AUP-UCI, lenguaje de programación Python, lenguaje de modelado UML, como herramienta de modelado Visual Paradigm, framework de desarrollo Odoo, entorno de desarrollo PyCharm y como SGBD PostgreSQL.

Capítulo 2: Descripción de la solución propuesta

Capítulo 2: Descripción de la solución propuesta

2.1 Introducción

En el proceso de construcción de software una de las partes más difíciles del trabajo conceptual es decidir precisamente qué construir, de modo que se logre entender las necesidades del cliente y los demás involucrados. En el presente capítulo se hace una descripción de la solución propuesta, de acuerdo a la situación problemática planteada anteriormente. Se especifican y se describen los requisitos funcionales y no funcionales incluyendo una muestra del prototipo de interfaz de usuario. Además se generan los diagramas de clases y secuencia pertenecientes al diseño. Finalmente se realiza el modelo de datos de la aplicación para manejar los datos de cada una de las entidades y sus relaciones entre ellas.

2.2 Descripción del sistema

Se propone como solución a la problemática planteada un módulo para la gestión del transporte en la empresa TRANSCARGO. Dicho módulo brinda la funcionalidad de gestionar los vehículos y los choferes. Una vez incluidos los choferes ofrece la posibilidad de notificar la actualización de documentos oficiales como el chequeo médico y la recalificación de su licencia de conducción. Además de llevar los registros de las hojas de ruta y los anexos 1. También permite la asignación de los choferes a los vehículos, el control de las reparaciones, los gastos, las multas, la asignación de herramientas a los choferes, autorizos de parqueo, controles sorpresivos, todo lo que concierne al transporte en la institución. La aplicación propuesta aspira ser una herramienta útil para agilizar la gestión de la información.

2.3 Requisitos

Los requisitos son declaraciones que identifican atributos, capacidades, características y/o cualidades que necesita cumplir un sistema (o un sistema de software) para que tenga valor y utilidad para el usuario. En otras palabras, los requisitos muestran qué elementos y funciones son necesarias para un proyecto.(31)

La entrevista en profundidad con las personas involucradas en todo lo que concierne al transporte en la empresa, permitió el levantamiento de información. Generando así 76 requisitos funcionales (RF) y 6 requisitos no funcionales en la especificación de requisitos. Además de la prioridad de cada uno de ellos definida por el cliente y la complejidad vista a través del documento oficial del centro de desarrollo FORTES (Centro de Tecnologías para la Formación). Este documento plantea cada una de las pautas que debe seguir un requisito para que su complejidad sea alta, baja o media, para tener más información sobre las especificaciones de requisitos consultar [Anexo 4](#).

Capítulo 2: Descripción de la solución propuesta

2.3.1 Requisitos funcionales

Tabla: 1 Requisitos funcionales

No	Descripción	Prioridad para el cliente	Complejidad
RF1	Incluir chofer	Alta	Alta
RF2	Registrar trazas del chofer	Alta	Media
RF3	Modificar datos del chofer	Alta	Alta
RF4	Eliminar chofer	Alta	Alta
RF5	Ver datos de un chofer	Alta	Media
RF6	Listar chofer	Media	Baja
RF7	Filtrar listado de choferes	Baja	Media
RF8	Incluir gasto	Alta	Alta
RF9	Registrar trazas del gasto	Media	Media
RF10	Modificar datos del gasto	Alta	Alta
RF11	Eliminar gasto	Media	Alta
RF12	Ver datos del gasto	Media	Media
RF13	Listar gastos	Media	Baja
RF14	Filtrar listado de gastos	Baja	Media
RF15	Incluir hoja de ruta	Alta	Alta
RF16	Registrar trazas de hoja de ruta	Media	Media
RF17	Modificar datos de la hoja de ruta	Alta	Alta
RF18	Eliminar hoja de ruta	Media	Alta
RF19	Ver datos de la hoja de ruta	Media	Media
RF20	Listar hoja de ruta	Media	Baja
RF21	Filtrar listado de hoja de ruta	Baja	Media
RF22	Incluir anexo 1	Alta	Alta
RF23	Registrar trazas del anexo 1	Media	Media
RF24	Modificar datos del anexo 1	Alta	Alta
RF25	Eliminar anexo 1	Media	Alta
RF26	Ver datos del anexo 1	Media	Media

Capítulo 2: Descripción de la solución propuesta

RF27	Listar anexo 1	Media	Baja
RF28	Filtrar listado de anexo 1	Baja	Media
RF29	Incluir vehículo	Alta	Alta
RF30	Registrar trazas del vehículo	Media	Media
RF31	Modificar datos del vehículo	Alta	Alta
RF32	Eliminar vehículo	Media	Alta
RF33	Ver datos de un vehículo	Media	Media
RF34	Listar vehículos	Media	Baja
RF35	Filtrar listado de vehículos	Baja	Media
RF36	Generar reporte de explotación	Media	Media
RF37	Incluir una reparación	Alta	Alta
RF38	Registrar trazas de una reparación	Media	Media
RF39	Modificar una reparación	Alta	Alta
RF40	Eliminar una reparación	Media	Alta
RF41	Ver datos de la reparación	Media	Media
RF42	Listar una reparación	Media	Baja
RF43	Filtrar listado de reparaciones	Baja	Media
RF44	Incluir herramienta	Alta	Alta
RF45	Registrar trazas de una herramienta	Media	Media
RF46	Modificar herramienta	Alta	Alta
RF47	Eliminar herramienta	Media	Alta
RF48	Ver herramienta	Media	Media
RF49	Listar herramientas	Media	Baja
RF50	Filtrar herramientas	Baja	Media
RF51	Incluir multa	Alta	Alta
RF52	Registrar trazas de una multa	Media	Media
RF53	Modificar multa	Alta	Alta
RF54	Eliminar multa	Media	Alta
RF55	Ver multa	Media	Media

Capítulo 2: Descripción de la solución propuesta

RF56	Listar multa	Media	Baja
RF57	Filtrar multa	Baja	Media
RF58	Ver notificaciones	Media	Media
RF69	Incluir autorizo de parqueo	Alta	Alta
RF60	Registrar trazas de autorizo de parqueo	Media	Media
RF61	Modificar autorizo de parqueo	Alta	Alta
RF62	Eliminar autorizo de parqueo	Media	Alta
RF63	Ver autorizo de parqueo	Media	Media
RF64	Listar autorizo de parqueo	Media	Baja
RF65	Filtrar autorizo de parqueo	Baja	Media
RF66	Incluir control sorpresivo	Alta	Alta
RF67	Registrar trazas de control sorpresivo	Media	Media
RF68	Modificar control sorpresivo	Alta	Alta
RF79	Eliminar control sorpresivo	Media	Alta
RF70	Ver control sorpresivo	Media	Media
RF71	Listar control sorpresivo	Media	Baja
RF72	Filtrar control sorpresivo	Baja	Media
RF73	Incluir marca	Alta	Alta
RF74	Eliminar marca	Media	Alta
RF75	Ver marca	Media	Media
RF76	Listar marca	Media	Baja

2.3.1.1 Descripción de requisitos funcionales

La descripción de requisitos muestra el flujo de eventos que debe tener cada requisito cuando se ejecuta, la secuencia de pasos y condiciones que se cumplen tanto por parte del usuario como del sistema.

A continuación, se presenta la descripción del requisito funcional correspondiente al RF: Incluir chofer. Para tener información sobre las demás descripciones de requisitos consultar [Anexo 5](#).

Capítulo 2: Descripción de la solución propuesta

Tabla 2: Descripción del RF: Incluir chofer

Precondiciones	El usuario debe estar autenticado en el sistema. El usuario debe poseer los permisos para incluir un chofer.
Flujo de eventos	
Flujo básico Incluir chofer	
1.	El usuario selecciona la opción Incluir chofer.
2.	<p>El sistema debe permitir introducir y/o seleccionar los siguientes datos para incluir un chofer:</p> <ul style="list-style-type: none"> - (*) Nombre(s) y apellidos: representa el nombre y los apellidos de un chofer. Campo de texto que admite caracteres alfabéticos con un máximo de 80 caracteres. Ejemplo: Paco Pérez León. - (*) Carnet de identidad: número identificativo de los ciudadanos cubanos. Campo de texto que admite caracteres numéricos y tiene un máximo de 11 caracteres. - (*) Licencia de conducción: contiene el número de la licencia de un chofer. Campo de texto que admite caracteres alfanuméricos y tiene un máximo de 11 caracteres. Ejemplo: 6P643533692. - (*) Tipo de licencia: representa el tipo de licencia de cada chofer. Campo de selección múltiple, que incluye. <ul style="list-style-type: none"> • A-1 ciclomotor • B automóvil • C1- camión • C camión • D-1 microbús • D ómnibus • E articulado • F agroindustrial • FE tractor con remolque - Dirección particular: representa la dirección particular en la que reside un chofer. Campo de texto que tiene un máximo de hasta 255 caracteres. - Estado civil: representa el estado civil de cada chofer. Campo de selección simple que incluye: <ul style="list-style-type: none"> • Soltero • Casado • Viudo - Cant. de accidentes: número de accidentes que ha tenido el chofer. Campo numérico que admite hasta 2 caracteres. - (*) Fecha de recalificación: representa la fecha de la última recalificación de la licencia del chofer, se realiza cada 6 meses. Campo de texto que solo admite formato de fecha: DD/MM/AAAA. Ejemplo: 13/05/2014. - (*) Fecha del chequeo médico: representa la fecha del último chequeo médico del chofer, se realiza cada 2 años. Campo de texto que solo admite formato de fecha: DD/MM/AAAA. Ejemplo: 13/05/2014. - (*) Fecha de autorizo de circulación: representa la fecha del último autorizo de circulación que se hizo el chofer. Campo de texto que solo admite formato de fecha: DD/MM/AAAA. Ejemplo: 13/05/2014.

Capítulo 2: Descripción de la solución propuesta

	<ul style="list-style-type: none"> - Herramientas asignadas: representa un listado con las herramientas asignadas al vehículo hasta el momento. Campo de texto que admite caracteres alfabéticos. Ejemplo: gato. - Cartilla operativa: representa un listado con un conjunto de pautas que tiene que cumplir el chofer para poder operar con el vehículo. Campo de texto que admite caracteres alfabéticos. - (*) Resultado del chequeo médico: representa el resultado del último chequeo médico del chofer. Campo de selección simple, que incluye: <ul style="list-style-type: none"> • Aprobado • Desaprobado - (*) Estado: representa el estado del chofer con respecto al vehículo, campo de selección simple, que incluye: <ul style="list-style-type: none"> • Libre: por defecto aparece libre • Ocupado
3.	Se introducen y/o seleccionan los datos para incluir un chofer.
4.	El sistema brinda las siguientes opciones: <ul style="list-style-type: none"> - <i>Guardar</i> - <i>Descartar</i>
5.	El usuario selecciona la opción Guardar.
6.	<i>El sistema valida los datos.</i>
7.	<i>El sistema guarda los datos del nuevo chofer</i>
8.	Concluye así el requisito.
Pos-condiciones	
1.	Se incluyó un nuevo chofer satisfactoriamente.
Flujos alternativos	
Flujo alternativo 6.a Información incompleta	
1.	El sistema señala el o los campos obligatorios que no hayan sido introducidos y/o seleccionados y muestra el mensaje de información: <i>Los siguientes campos son obligatorios;</i> indicando los campos en cuestión.
2.	El usuario introduce y/o selecciona los datos.
3.	Volver al paso 3 del flujo básico
Pos-condiciones	
1.	NA
Flujos alternativo 6.b Información incorrecta	
1.	El sistema señala el o los campos incorrectos y muestra el mensaje de información: <i>Los siguientes campos son incorrectos;</i> indicando los campos en cuestión.
2.	El usuario introduce y/o selecciona correctamente los datos.
3.	Volver al paso 3 del flujo básico.
Pos-condiciones	
1.	N/A
Flujo alternativo 4.a Descartar cuando se hayan introducido datos	
1.	El usuario selecciona la opción <i>Descartar</i>
2.	El sistema muestra un mensaje de información: <i>¿Está seguro que quiere abandonar la página?</i>
3.	El sistema brinda las siguientes opciones:

Capítulo 2: Descripción de la solución propuesta

	- <i>Aceptar</i> - <i>Cancelar</i>
4.	El usuario selecciona la opción <i>Aceptar</i>
5.	El sistema elimina los datos introducidos y regresa al listado de choferes.
6.	Concluye así el requisito.
Pos-condiciones	
1.	No se crea la entidad chofer.
Flujo alternativo 4.b Descartar cuando se hayan introducido datos opción Cancelar	
1.	El usuario selecciona la opción <i>Cancelar</i> .
2.	Volver al paso 4.a del flujo básico.
Pos-condiciones	
1	N/A
Flujo alternativo * Descartar	
1.	El sistema regresa al listado de choferes.
2.	Concluye así el requisito.
Pos-condiciones	
1.	No se crea la entidad chofer
Validaciones	
	N/A
Conceptos	NA
Requisitos especiales	
Asuntos pendientes	NA

Capítulo 2: Descripción de la solución propuesta

SIGAX Sistema de Gestión para Actividades Transitarias y de Carga

http://www.transcargo.transnet.cu

Comercial Corresponsalia **Modulo Transporte** Modulo .. N Configuración Usuario

Chofer/Nuevo

Guardar o Descartar

LOGO Odo

- CHOFER
 - Chofer**
 - Multa
- VEHICULO
 - Hoja de ruta
 - Anexo 1
 - Gasto
 - Herramienta
 - Explotación
 - Reparación
 - Reparación capital
 - Cambiar pieza
 - Autorizo de parqueo
 - Control sorpresivo
- CONFIGURACIÓN
 - Marca
 - Estado de chofer
 - Estado de vehículo

Libre Ocupado

Nombre(s) y apellidos * ? 0 Multa

Carnet de identidad * ?

Licencia de conducción * ?

Tipo de licencia * ?

Dirección particular ?

Estado civil ?

Cant. de accidentes ?

Fecha de recalificación * ?

Fecha del chequeo médico * ?

Fecha de autorizo de circulación * ?

Herramientas asignadas ?	Chofer	Chapa	Herramienta
	Añadir un elemento		

Cartilla operativa ?

Resultado del chequeo médico * ?

Figura 1: Prototipo interfaz de usuario del RF Incluir chofer

Capítulo 2: Descripción de la solución propuesta

SIGAX Sistema de Gestión para Actividades Transitorias y de Carga

http://www.transcargotransnet.cu

Comercial Corresponsalia **Modulo Transporte** Modulo . N Configuración

Usuario

Los siguientes campos son obligatorios:

Nombre(s) y apellidos
Licencia de conducción

LOGO
Odoo

Chofer/Nuevo

Guardar o Descartar

Libre Ocupado

Nombre(s) y apellidos * ?

Carnet de identidad * ? 930308080851

Licencia de conducción * ?

Tipo de licencia * ? Automovil

Dirección particular ? Calle 14, % 21 y 22, Playa, Ciudad Habana

Estado civil ? Casado/a

Cant. de accidentes ? 3

Fecha de recalificación * ? 10/12 /2016

Fecha del chequeo médico * ? 4/5 /2016

Fecha de autorizo de circulación * ? 12 /6 /2016

Herramientas asignadas ?

Chofer	Chapa	Herramienta
Añadir un elemento		

Cartilla operativa ?

Debe de tener con el siempre los documentos oficiales del vehículo.
Debe tener los documentos actualizados.

Resultado del chequeo médico * ? Aprobado

Figura 2: Prototipo interfaz de usuario del RF Incluir chofer (Flujo alternativo información incompleta)

Capítulo 2: Descripción de la solución propuesta

SIGAX Sistema de Gestión para Actividades Transitorias y de Carga

http://www.transcargotransnet.cu

Comercial Corresponsalia **Modulo Transporte** Modulo .. N Configuración

Usuario

Los siguientes campos son incorrectos:
Licencia de conducción

Chofer/Nuevo

Guardar o Descartar

LOGO Odo

- CHOFER
 - Chofer
 - Multa
- VEHICULO
 - Hoja de ruta
 - Anexo 1
 - Gasto
 - Herramienta
 - Explotación
 - Reparación
 - Reparación capital
 - Cambiar pieza
 - Autorizo de parqueo
 - Control sorpresivo
- CONFIGURACIÓN
 - Marca
 - Estado de chofer
 - Estado de vehiculo

Libre Ocupado

Nombre(s) y apellidos * ? Paco Pérez León. 0 Multa

Carnet de identidad * ? 930308080851

Licencia de conducción * ? 63@#%\$3698

Tipo de licencia * ? Automovil

Dirección particular ? Calle 14, % 21 y 22, Playa, Ciudad Habana

Estado civil ? Casado/a

Cant. de accidentes ? 3

Fecha de recalificación * ? 10/12 /2016

Fecha del chequeo médico * ? 4/5 /2016

Fecha de autorizo de circulación * ? 12 /6 /2016

Chofer	Chapa	Herramienta
Añadir un elemento		

Cartilla operativa ? Debe de tener con el siempre los documentos oficiales del vehiculo.
Debe tener los documentos actualizados.

Resultado del chequeo médico * ? Aprobado

Figura 3: Prototipo interfaz de usuario del RF Incluir chofer (Flujo alternativo información incorrecta)

Capítulo 2: Descripción de la solución propuesta

2.3.2 Requisitos no funcionales

- ✓ Confiabilidad
 - Seguridad de acceso:
 - Permitir el acceso a la información solo a los usuarios autorizados y denegar el acceso a aquellos que no lo están.
 - Establecer un sistema de roles, donde cada rol solo tiene acceso a un conjunto de funcionalidades específicas.
 - Recuperabilidad:
 - Verificar las consecuencias asociadas al fallo del sistema.
- ✓ Eficiencia
 - Comportamiento en el tiempo:
 - Proporcionar tiempos temporales de respuesta al realizar acciones.
- ✓ Portabilidad
 - Instalabilidad:
 - Ser instalado en cualquier entorno que cuente con un servidor de base de datos relacional con cualquier gestor de BD, se recomienda PostgreSQL v 9.2 o superior.
 - Los servidores web deben tener como mínimo 2Gb de RAM y 100GB de HDD.
 - Las estaciones clientes desde las que se accederá al sistema pueden tener cualquier sistema operativo y solo deberán tener un navegador web moderno instalado y como mínimo 512 MB de RAM.
- ✓ Usabilidad
 - Conformidad:
 - Cumplir con las pautas de diseño establecidas en la Estrategia Marcaria de la Universidad y las establecidas por el diseñador del proyecto.
 - Operabilidad:
 - Permitir al administrador del sistema el control total del mismo, además de realizar todas las configuraciones necesarias para que pueda ser utilizado por los restantes usuarios que tendrán acceso a la herramienta.

Capítulo 2. Descripción de la solución propuesta

2.4 Diseño y arquitectura de la solución propuesta

Durante el ciclo de vida del software, y a fin de garantizar diversos factores de calidad, se tienen en cuenta un conjunto de patrones que proporcionan un marco de referencia necesario para guiar la construcción de un software. A fin de garantizar una organización entre los componentes fundamentales del software y las relaciones entre ellos. Es por ello que el diseño y la arquitectura del software constituyen la columna vertebral para construir un sistema informático.

2.4.1 Estilo y patrones

Los diseñadores de software van creando un amplio repertorio de principios y expresiones que los guían a desarrollar el software, a unas y a otras se les puede asignar el nombre de patrones siempre y cuando se codifiquen en un formato estructurado que describa el problema y su solución. Por lo que se puede afirmar que un patrón es una descripción de un problema y su solución que recibe un nombre y que puede emplearse en otros contextos; en teoría, indica la manera de utilizarlo en circunstancias adversas (32).

Los patrones no se proponen descubrir ni expresar nuevos principios de la ingeniería de software, solo intentan codificar el conocimiento, las expresiones y los principios ya existentes. Los patrones son soluciones simples y elegantes a problemas específicos y comunes del diseño orientado a objetos, son soluciones basadas en la experiencia y que se han demostrado que funcionan. Pero existe una diferencia notable entre estilo arquitectónico, patrón arquitectónico y patrón de diseño aunque se relacionan entre sí formando parte de un todo, estos son los que en gran medida aportan calidad al sistema resultante. El estilo arquitectónico, patrón arquitectónico y patrón de diseño, representan, de lo general a lo particular, los niveles de abstracción que componen la arquitectura de software. (32)

El **estilo arquitectónico** es el encargado de:

- ✓ Describir la estructura general de un sistema, independientemente de otros estilos.
- ✓ Definir los componentes del sistema, su relación e interactividad.

El **patrón arquitectónico** es el nivel en el cual la arquitectura de software:

- ✓ Define la estructura básica de un sistema, pudiendo estar relacionada con otros patrones.
- ✓ Representa una plantilla de construcción que provee un conjunto de subsistemas aportando las normas para su organización.

Capítulo 2: Descripción de la solución propuesta

El **patrón de diseño** es el tercer nivel de abstracción de la arquitectura de software, cuya finalidad es la de precisar en detalles los subsistemas y componentes de la aplicación (32).

Habiendo marcado la diferencia entre estos tres conceptos que conforman los niveles de abstracción de la arquitectura de software, están creadas las condiciones de abordar los detalles de los niveles de la arquitectura definidos a utilizar en el desarrollo del Módulo Transporte, los cuales serán aplicados a la propuesta de solución.

2.4.1.1 Estilo o modalidad arquitectónica

La arquitectura del sistema Odoo es cliente – servidor, lo que permite que todos los usuarios trabajen sobre el mismo repositorio de datos. Esto tiene la ventaja de que toda la información está disponible y sincronizada en todo momento además de que descarga la mayor parte del trabajo de procesamiento de datos de las máquinas cliente (donde trabajan efectivamente los usuarios).(33)

La arquitectura cliente-servidor es un modelo de aplicación distribuida en el que las tareas se reparten entre los proveedores de recursos o servicios, llamados servidores, y los demandantes, llamados clientes. Un cliente realiza peticiones a otro programa, el servidor, quien le da respuesta. Esta idea también se puede aplicar a programas que se ejecutan sobre una sola computadora, aunque es más ventajosa en un sistema operativo multiusuario distribuido a través de una red de computadoras.(33)

La separación entre cliente y servidor es una separación de tipo lógico, donde el servidor no se ejecuta necesariamente sobre una sola máquina ni es necesariamente un sólo programa. Los tipos específicos de servidores incluyen los servidores web, los servidores de archivo, los servidores del correo, etc. Mientras que sus propósitos varían de unos servicios a otros, la arquitectura básica seguirá siendo la misma.(32)

Un sistema de Odoo está formado por tres componentes principales:(32)

- ✓ El servidor de base de datos PostgreSQL, el cual contiene todas las bases de datos, cada una contiene todos los datos y la mayoría de los elementos de configuración relacionados con el sistema Odoo.
- ✓ El servidor de Odoo el cual contiene toda la lógica empresarial.
- ✓ El servidor web, el cual permite conectarse a Odoo desde un navegador web estándar.

Capítulo 2: Descripción de la solución propuesta

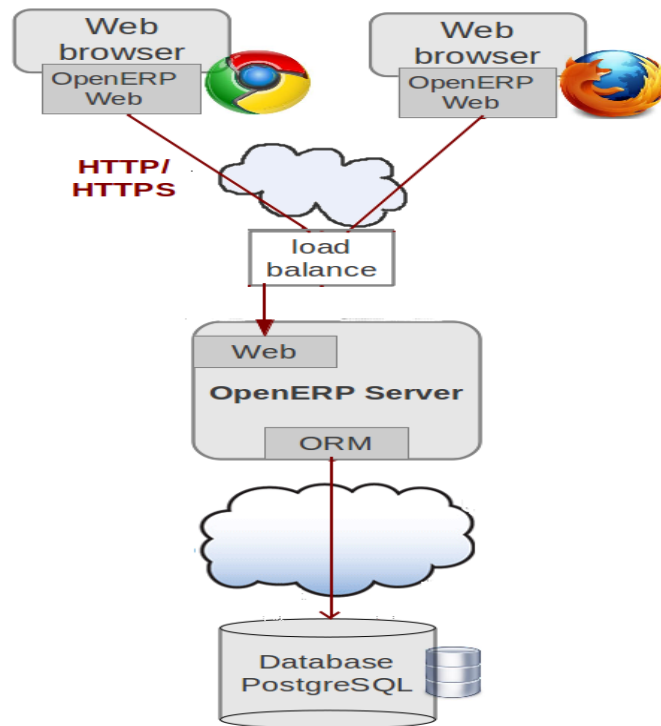


Figura: 4 Arquitectura de Odoo

El componente web puede ser pensado como un servidor o un cliente dependiendo de su forma de verlo. Este actúa como un servidor web para un usuario final, conectándose desde un navegador web, pero también puede actuar como un cliente al servidor de aplicaciones de Odoo.

En la arquitectura C/S el remitente de una solicitud es conocido como cliente. Sus características son:(32)

- ✓ Es quien inicia solicitudes o peticiones, tienen por tanto un papel activo en la comunicación (dispositivo maestro o amo).
- ✓ Espera y recibe las respuestas del servidor.
- ✓ Por lo general, puede conectarse a varios servidores a la vez.
- ✓ Normalmente interactúa directamente con los usuarios finales mediante una interfaz gráfica de usuario.

Al receptor de la solicitud enviada por el cliente se conoce como servidor. Sus características son:(32)

Capítulo 2. Descripción de la solución propuesta

- ✓ Al iniciarse esperan a que lleguen las solicitudes de los clientes, desempeñan entonces un papel pasivo en la comunicación (dispositivo esclavo).
- ✓ Tras la recepción de una solicitud, la procesan y luego envían la respuesta al cliente.
- ✓ Por lo general, acepta las conexiones de un gran número de clientes (en ciertos casos el número máximo de peticiones puede estar limitado).

En la arquitectura C/S sus características generales son:(32)

- ✓ El Cliente y el Servidor pueden actuar como una sola entidad y también pueden actuar como entidades separadas, realizando actividades o tareas independientes.
- ✓ Las funciones de Cliente y Servidor pueden estar en plataformas separadas, o en la misma plataforma.
- ✓ Cada plataforma puede ser escalable independientemente. Los cambios realizados en las plataformas de los Clientes o de los Servidores, ya sean por actualización o por reemplazo tecnológico, se realizan de una manera transparente para el usuario final.
- ✓ La interrelación entre el hardware y el software están basados en una infraestructura poderosa, de tal forma que el acceso a los recursos de la red no muestra la complejidad de los diferentes tipos de formatos de datos y de los protocolos.
- ✓ Su representación típica es un centro de trabajo (PC), en donde el usuario dispone de sus propias aplicaciones de oficina y sus propias bases de datos, sin dependencia directa del sistema central de información de la organización.

2.4.1.2 Patrón arquitectónico

El patrón conocido como Modelo-Vista-Controlador (MVC) separa el modelado del dominio, la presentación y las acciones basadas en datos ingresados por el usuario en tres clases diferentes.(32) De manera genérica, los componentes de MVC se podrían definir como sigue:(32)

- ✓ El Modelo: es la representación de la información con la cual el sistema opera, por lo tanto gestiona todos los accesos a dicha información, tanto consultas como actualizaciones, implementando también los privilegios de acceso que se hayan descrito en las especificaciones de la aplicación (lógica de negocio). Envía a la 'vista' aquella parte de la información que en cada momento se le solicita para que sea mostrada (típicamente a un usuario). Las peticiones de acceso o manipulación de información llegan al modelo a través del controlador.

Capítulo 2. Descripción de la solución propuesta

- ✓ El Controlador: responde a eventos (usualmente acciones del usuario) e invoca peticiones al 'modelo' cuando se hace alguna solicitud sobre la información (por ejemplo, editar un documento o un registro en una base de datos). También puede enviar comandos a su 'vista' asociada si se solicita un cambio en la forma en que se presenta el 'modelo' (por ejemplo, desplazamiento o *scroll* por un documento o por los diferentes registros de una base de datos), por tanto se podría decir que el 'controlador' hace de intermediario entre la vista y el modelo.
- ✓ La Vista: presenta el modelo (información y lógica de negocio) en un formato adecuado para interactuar (usualmente la interfaz de usuario) por tanto requiere de dicho 'modelo' la información que debe representar como salida.

Tanto la vista como el controlador dependen del modelo, el cual no depende de las otras clases. Esta separación permite construir y probar el modelo independientemente de la representación visual. La separación entre vista y controlador puede ser secundaria en aplicaciones de clientes ricos y, de hecho, muchos *framework* de interfaz implementan ambos roles en un solo objeto. En aplicaciones Web, por otra parte, la separación entre la vista (el *browser*) y el controlador (los componentes del lado del servidor que manejan los requerimientos de HTTP) está mucho más expresamente definida.(32)

Capítulo 2: Descripción de la solución propuesta

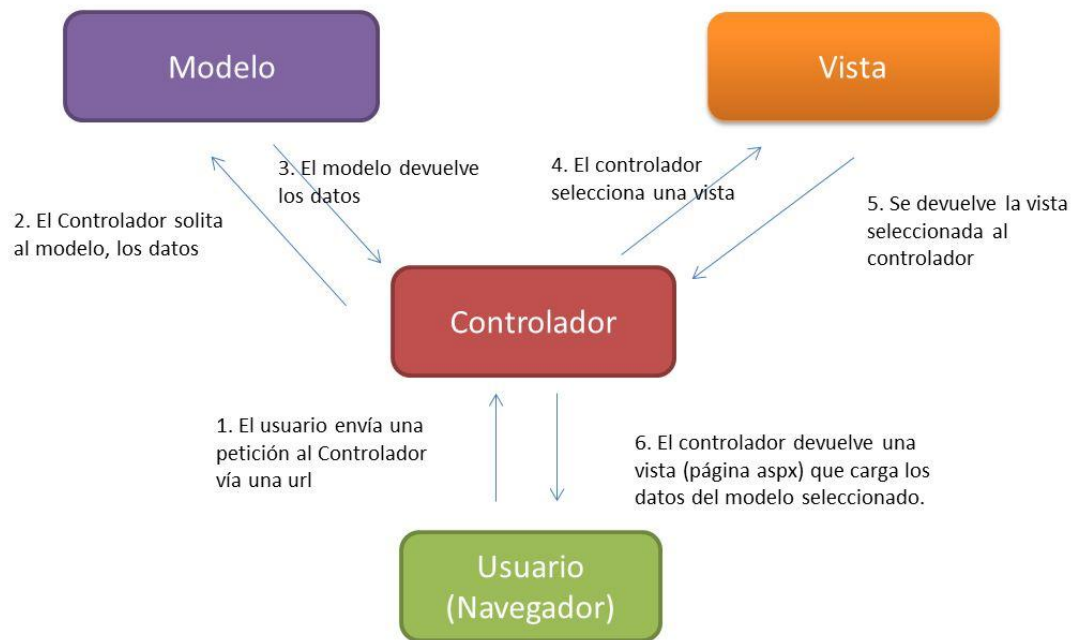


Figura: 5 Modelo Vista Controlador

2.4.1.3 Patrones de diseño

Los patrones de diseño brindan una solución ya probada y documentada a problemas de desarrollo de software que están sujetos a contextos similares. Existen diversos patrones que en dependencia de su uso, permiten resolver un problema general de diseño, dentro de un contexto particular, constituyendo una vía para que los desarrolladores realicen buenas prácticas en la implementación.(34)

Los patrones de diseño pretenden:(34)

- ✓ Proporcionar catálogos de elementos reusables en el diseño de sistemas software.
- ✓ Evitar la reiteración en la búsqueda de soluciones a problemas ya conocidos y solucionados anteriormente.
- ✓ Formalizar un vocabulario común entre diseñadores.
- ✓ Estandarizar el modo en que se realiza el diseño.

Capítulo 2: Descripción de la solución propuesta

- ✓ Facilitar el aprendizaje de las nuevas generaciones de diseñadores condensando conocimiento ya existente.

Clasificación de patrones GOF(Banda de los Cuatro), por sus siglas en inglés *Gang of Four* : (34)

- ✓ Patrones creacionales: utilizados para instanciar objetos, y así separar la implementación del cliente de la de los objetos que se utilizan. Con ellos se intenta separar la lógica de creación de objetos y encapsularla.
- ✓ Patrones de comportamiento: se utilizan a la hora de definir cómo las clases y objetos interaccionan entre ellos.
- ✓ Patrones estructurales: son utilizados para crear clases u objetos incluidos dentro de estructuras más complejas.

Patrones utilizados en el diseño de la propuesta de solución:

Patrones GRASP (Patrones Generales de Software para Asignación de Responsabilidades), por sus siglas en inglés *General Responsibility Assignment Software Patterns*: Los patrones GRASP describen los principios fundamentales de la asignación de responsabilidades a objetos, expresados en forma de patrones. Estos patrones son:

- ✓ Experto: este patrón establece la responsabilidad de la creación de un objeto o la implementación de un método, debe recaer sobre la clase que conoce toda la información necesaria para crearlo. Se evidencia en la clase vehículo.
- ✓ Bajo acoplamiento: es la idea de tener las clases lo menos ligadas entre sí que se pueda. De tal forma que en caso de producirse una modificación en alguna de ellas, se tenga la mínima repercusión posible en el resto de clases, potenciando la reutilización, y disminuyendo la dependencia entre las clases. Se evidencia en la clase gasto.

Patrones GOF:

Patrones de comportamiento

- ✓ *Observer* (Observador): Define una dependencia de uno-a-muchos entre objetos, de forma que cuando un objeto cambie de estado se notifique y actualicen automáticamente todos los objetos que dependen de él. Se evidencia en la clase chofer.

Capítulo 2: Descripción de la solución propuesta

2.5 Modelo conceptual

Un modelo conceptual es un conjunto de conceptos y de reglas destinados a representar de forma global los aspectos lógicos de los diferentes tipos de elementos existentes en la realidad que está siendo analizada. Este ayuda a comprender cuales son los principales elementos que componen la brigada de transporte y como se relacionan entre sí. A continuación se presenta en la **figura 6** el modelo conceptual de la solución propuesta.

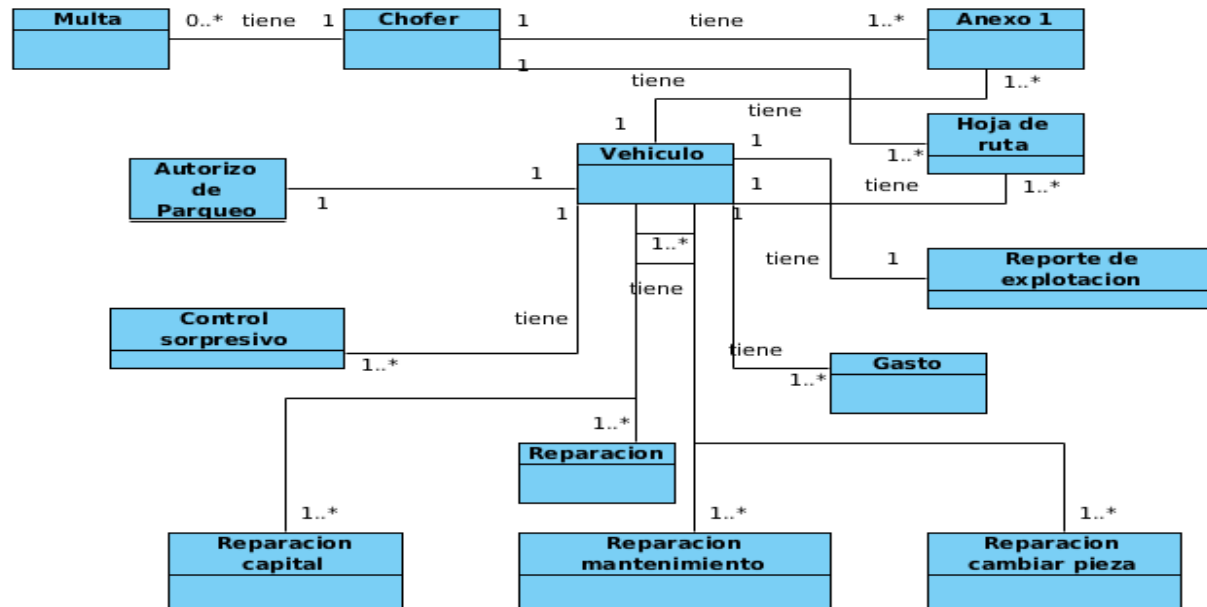


Figura: 6 Modelo Conceptual

Capítulo 2: Descripción de la solución propuesta

2.6 Diagrama de clases y secuencia del diseño

Los diagramas de clases del diseño son empleados para representar las relaciones que se establecen entre las clases, mientras que los de secuencia muestran la interacción de un conjunto de objetos en una aplicación a través del tiempo. A continuación se presenta el diagrama de clase y secuencia del diseño correspondiente al requisito funcional del sistema Incluir chofer, el resto se encuentran disponibles en el [Anexo # 6](#) y [Anexo # 7](#) del documento.

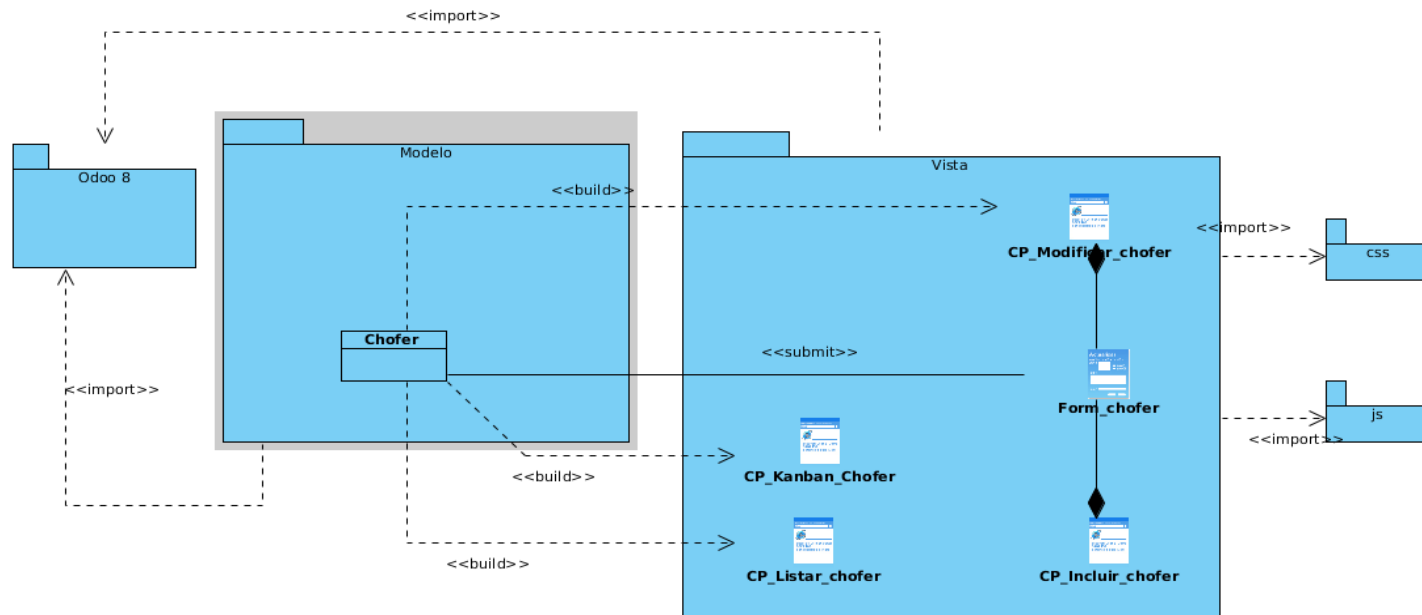


Figura: 7 Diagrama de clases del diseño del RF Incluir chofer

Capítulo 2: Descripción de la solución propuesta

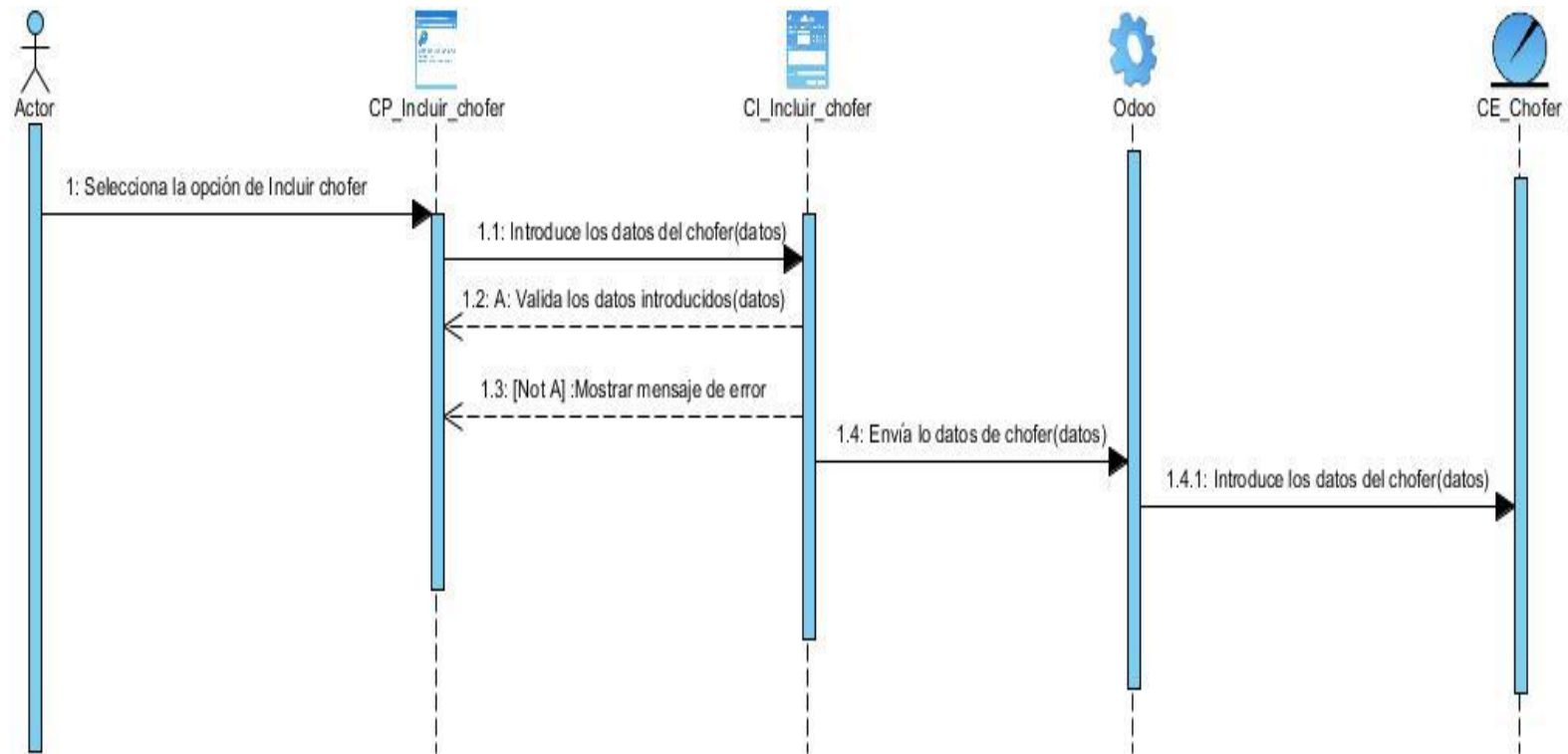


Figura: 8 Diagrama de secuencia del diseño del RF Incluir chofer

2.7 Modelo de datos

El modelo de datos permite identificar, analizar y manipular los datos de cada uno de los elementos que componen la brigada de transporte y el manejo de sus relaciones. Este cuenta con un total de 21 tablas persistentes que almacenan toda la información del transporte. A continuación se muestra en la **figura 9** el modelo de datos de la solución propuesta. Para una mejor comprensión de cada una de las tablas ver [Anexo # 8.](#)

Capítulo 2: Descripción de la solución propuesta

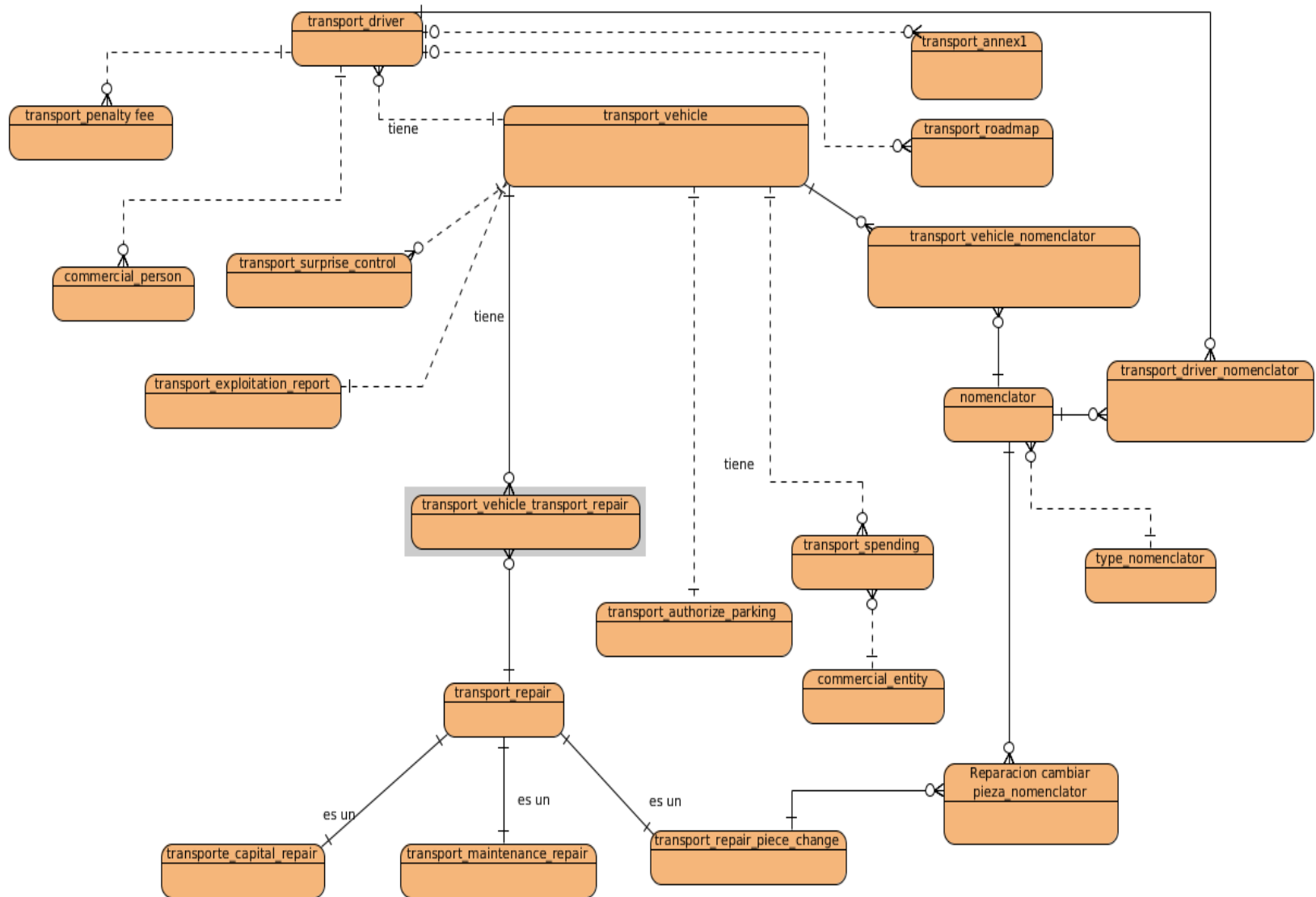


Figura: 9 Modelo de datos

Capítulo 2. Descripción de la solución propuesta

2.8 Conclusiones Parciales

La realización de la entrevista en profundidad ayudó a comprender el funcionamiento del transporte en la empresa. Por ende las técnicas de captura de requisitos utilizadas facilitaron la definición y comprensión de las funcionalidades del sistema, lo que permitió realizar el levantamiento de los requisitos funcionales y no funcionales en su totalidad. A través de la especificación y descripción de los requisitos se documentó toda la información relativa a la propuesta de solución, garantizando la revisión y aceptación por todos los involucrados. Por otro lado la descripción de los estándares de la arquitectura y el diseño permite que se cuente con los elementos necesarios para la implementación del módulo para la gestión del transporte.

Capítulo 3: Implementación y Prueba

3.1 Introducción

En el presente capítulo se describen los estándares de código fundamentales usados en la implementación de la solución propuesta. Además se modela el diagrama de componentes para mostrar la organización y las dependencias lógicas que existen entre los ficheros que contienen código fuente. Una vez culminado el proceso de implementación se procede a aplicar un conjunto de pruebas que permiten validar la eficiencia y el rendimiento de la aplicación informática.

3.2 Estándar de codificación

Un estándar de codificación son reglas que se siguen para la escritura del código fuente. Estos permiten que otros programadores puedan identificar las variables, las funciones o métodos al leer los códigos de otras personas. Se definen estándares de codificación porque un estilo de programación homogéneo en un proyecto para un lenguaje de programación permite que todos los participantes lo puedan entender en menos tiempo y que cualquier persona que se desempeñe como codificador de dicho lenguaje pueda interpretar de manera eficiente.(35)

A continuación se exponen más detalladamente, los estándares de codificación seguidos en el desarrollo del módulo:

Directorios: este estándar permite la organización de los directorios por los que está integrado el módulo los cuales son:

- ✓ *data/* : demo y datos XML
- ✓ *models/* : modelos de definición
- ✓ *controllers/*: contiene los controladores de rutas (HTTP).
- ✓ *views/* : contiene las vistas y plantillas
- ✓ *static/*: contiene los elementos web, separados en *css / js /, img /, lib /,...*

Nomenclatura de archivos: permite enunciar las clases del modelo y vistas, separando la lógica de negocio por conjuntos de modelos. Facilitando la comprensión de los elementos correspondientes en cada modelo permitiendo crear para cada conjunto denominado *<main_model>* los siguientes archivos:

- ✓ *models/<main_model>.py*
- ✓ *models/<inherited_main_model>.py*

Capítulo 3: Implementación y Pruebas

- ✓ `views/<main_model>_templates.xml`
- ✓ `views/<main_model>_views.xml`

Ejemplo:

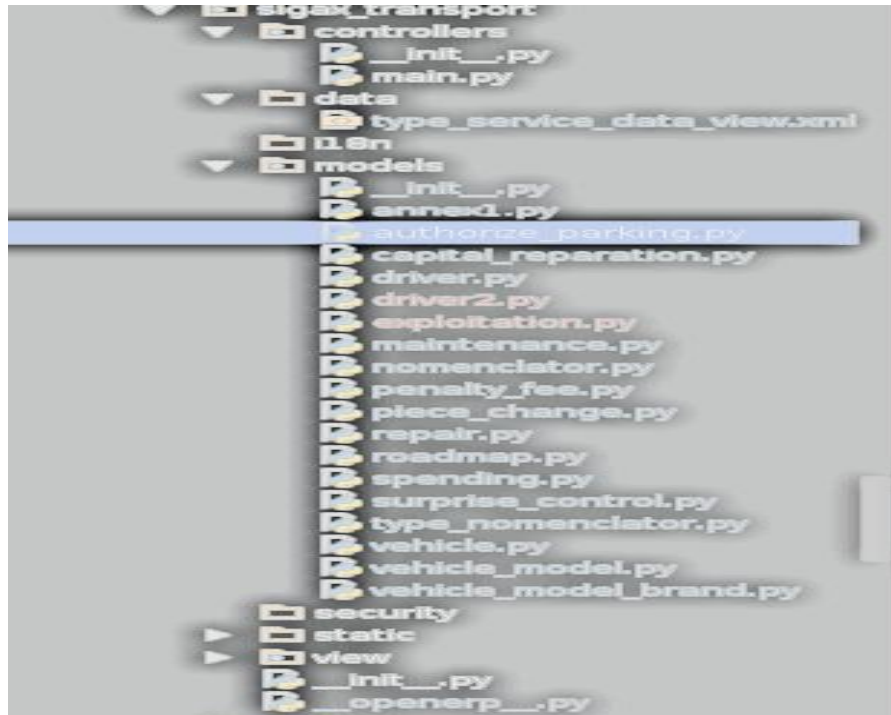


Figura: 10 Estándar de codificación

Archivos XML

Formato: Se utilizó para la declaración de registros en XML en los cuales el atributo `id` se escribe antes del modelo y en la declaración de campo el nombre de atributo es el primero.

Capítulo 3: Implementación y Pruebas

Ejemplo:

```
<record id='vehicle_view_tree' model='ir.ui.view'>
  <field name='name'>tree.vehicle</field>
  <field name='model'>vehicle</field>
  <field name='arch' type='xml'>
    <tree string='Vehicle' >
      <field name='model_id' class='oe_inline' />
      <field name='license_plate' class='oe_inline' />
      <field name='driver_id' />
      <field name='vehicle_type' />
      <field name='engine_number' />
      <field name='fuel_type' />
      <field name='repair' />
      <field name='state_id' />
      <field name='plan_fuel' class='oe_inline' />
      <field name='plan_index' />
      <field name='seats' />
      <field name='fabrication_date' />
      <field name='weight_kg' class='oe_inline' />
      <field name='fuel_type' />
      <field name='odometer_working' />
      <field name='colour' />
      <field name='dent' />
      <field name='circulation_date' class='oe_inline' />
      <field name='FICAO_date' class='oe_inline' />
      <field name='LOT_date' class='oe_inline' />
      <field name='litre_test' class='oe_inline' />
      <field name='result_litre_test' class='oe_inline' />
    </tree>
  </field>
</record>
```

Figura: 11 Estándar de codificación

Nombre xml_id

Se utilizó para definir los nombres de los elementos de seguridad, las vistas y las funcionalidades los cuales se utilizaron de la siguiente manera:

- ✓ en la vista la estructura es: `<model_name>_view_<view_type>`, donde `view_type` es Kanban, form, tree, search.
- ✓ en la funcionalidad: se representa así `<model_name>_action`.
- ✓ en la seguridad se define en grupo: `<model_name>_group_<group_name>` donde `group_name` es el nombre del grupo, por lo general 'usuario', 'administrador'.

Capítulo 3: Implementación y Pruebas

Ejemplo:

```
<record id='vehicle_view_form' model='ir.ui.view'>
  <field name="name">vehicle.view.form</field>
  <field name="model">vehicle</field>
  <field name="arch" type="xml">
    <form string="Vehicle">
  </field>
</record>
<record id='vehicle_view_tree' model='ir.ui.view'>
  <field name="name">tree.vehicle</field>
  <field name="model">vehicle</field>
  <field name="arch" type="xml">
    <tree string="Vehicle">
  </field>
</record>
<record id="vehicle_view_search" model="ir.ui.view">
  <field name="name">vehicle.view.search</field>
  <field name="model">vehicle</field>
  <field name="arch" type="xml">
</record>

<record id='vehicle_view_kanban' model='ir.ui.view'>
  <field name="name">vehicle.kanban</field>
  <field name="model">vehicle</field>
  <field name="arch" type="xml">
</record>

<record id='vehicle_action' model='ir.actions.act_window' >
  <field name="name">Vehicle</field>
  <field name="res_model">vehicle</field>
  <field name="view_type">form</field>
  <field name="view_mode">kanban,tree,form</field>
  <field name="help" type="html">
</record>

<menuitem name="Vehicle" parent="transporte_menupropio" id="fleet_vehicles" sequence="1" />
<menuitem action="vehicle_action" parent="fleet_vehicles" id="fleet_vehicle_menu" sequence="2"/>
```

Figura: 12 Estándar de codificación

Opciones PEP8

Se utilizó para resaltar las buenas prácticas de programación, donde cada línea cumple que no sea demasiado larga, se respeta la sangría, hay cierre de corchete y los bloques de comentario empiezan con '#'.

Import

Se utilizó para importar las librerías tanto internas como externas las cuales se organizan alfabéticamente de arriba hacia abajo y de izquierda a derecha.

:

Capítulo 3: Implementación y Pruebas

```
import datetime
import re
from openerp import api, exceptions, fields, models
from openerp import tools
from openerp.tools.translate import _
```

Figura: 13 Estándar de codificación

3.3 Diagrama de componentes

Un diagrama de componentes al igual que un sistema de software es dividido en componentes y muestra las dependencias entre ellos. 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.(36)

En el diagrama de componente de la presente investigación se cuenta con 3 paquetes los cuales son el paquete de modelo, de vistas y de interacción con el framework de desarrollo Odoo. El paquete de modelo está compuesto por todos los modelos de la propuesta de solución y el paquete de vista por las vistas de cada modelo. El modelo se encarga de guardar todos los datos en el sistema gestor de base de datos y las vistas son realizadas en archivos xml las cuales también son almacenadas en la base de datos. Odoo es el responsable de adquirir la información guardada por el modelo en la base de datos y a su vez de renderizar los archivos xml para convertirlos en vistas.

A continuación se presenta el diagrama de componentes del módulo para la gestión del transporte:

Capítulo 3: Implementación y Pruebas

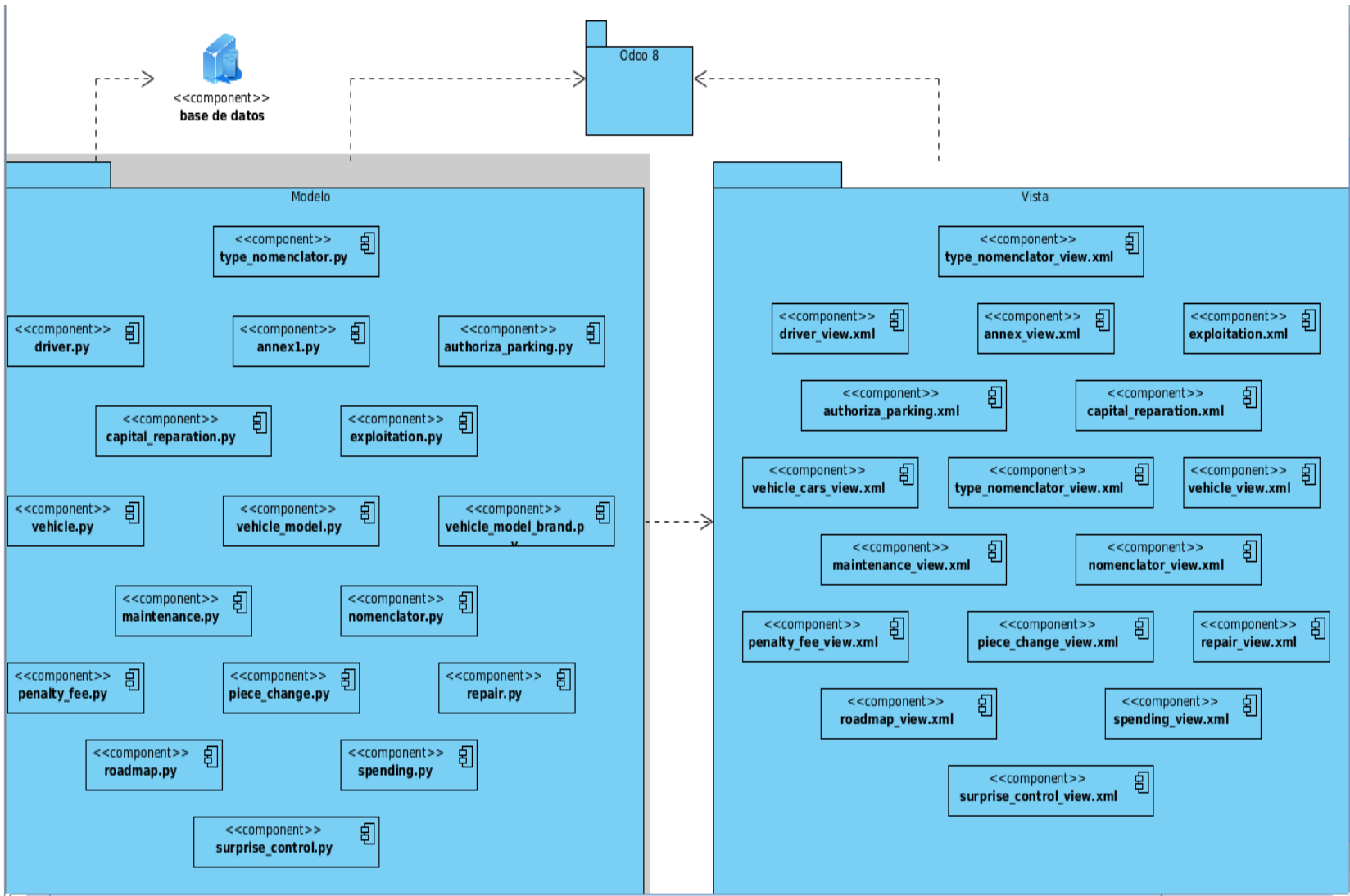


Figura: 14 Diagrama de componentes del módulo transporte

Capítulo 3: Implementación y Pruebas

3.4 Pruebas de software

Las pruebas son actividades en las cuales un sistema o componente es ejecutado bajo condiciones o requerimientos especificados. Las pruebas de software son realizadas con el objetivo de encontrar errores, verificando de esta manera la calidad del producto(14). En la realización de estas pruebas es importante comprobar la cobertura de los requisitos, dado que su incumplimiento puede comprometer la aceptación del sistema por el equipo de operación responsable de realizar las pruebas de implantación del sistema.

3.4.1 Niveles de Prueba

Las pruebas son aplicadas para diferentes tipos de objetivos, en diferentes escenarios o niveles de trabajo.(14) Entre los niveles de prueba que propone la metodología de desarrollo AUP-UCI se encuentran pruebas internas, pruebas de liberación y pruebas de aceptación. Las pruebas realizadas a la solución propuesta fueron pruebas funcionales y pruebas de aceptación.

Pruebas de liberación: la realizan una entidad certificadora de la calidad del software, estas se encargan de la Verificación de los requisitos. Además permite determinar si los requisitos están completos y correctos.

Pruebas de aceptación: al momento de aplicarlas el software está listo para implantarse, entonces debe ser el cliente quien realice estas pruebas apoyado por personas del equipo de pruebas directamente en el entorno donde se va a explotar el mismo.

Pruebas funcionales: es una prueba basada en la ejecución, revisión y retroalimentación de las funcionalidades previamente diseñadas para el software. Las pruebas funcionales se hacen mediante el diseño de modelos de prueba que buscan evaluar cada una de las opciones con las que cuenta el paquete informático. Dicho de otro modo son pruebas específicas, concretas y exhaustivas para probar y validar que el software hace lo que debe y sobre todo, lo que se ha especificado(14).

3.4.2 Métodos de Prueba

Las estrategias de pruebas del software se integran a las técnicas del diseño de Casos de Prueba (CP), en una serie de pasos planificados que dan como resultado una correcta construcción del software, con tal grado de confianza que se detectarán la mayor parte de los errores existentes en él (14).

Pruebas de caja negra o funcional: realizan pruebas sobre la interfaz del programa a probar, entendiendo por interfaz las entradas y salidas de dicho programa. No es necesario conocer la lógica del programa, solo la funcionalidad que debe realiza(14).

Capítulo 3: Implementación y Pruebas

3.4.3 Diseño de Casos de Prueba

A continuación se muestra el diseño de caso de prueba correspondiente a la funcionalidad Incluir chofer. Los restantes diseños de casos de pruebas pueden ser consultados en el **Anexo # 9**.

Tabla: 3 Diseño de caso de prueba del RF Incluir chofer

Escenario	EC1.1 Incluir chofer.	EC 1.2 Guardar los datos	EC 1.3 Información incompleta	EC 1.4 Información incorrecta	EC 1.5 Descartar cuando se hayan introducido datos	EC 1.6 Descartar cuando se hayan introducido datos opción Aceptar	EC 1.7 Descartar cuando se hayan introducido datos opción Cancelar	EC 1.8 Descartar
Descripción	Se selecciona la opción crear para incluir un nuevo chofer	Se Introducen y/o seleccionan los datos del chofer y selecciona la opción guardar.	Existen campos obligatorios que no se completaron	Existen campos incorrectos.	Se selecciona el botón descartar cuando ya se han introducido datos	Se selecciona la opción Aceptar	Se selecciona la opción Cancelar	Se selecciona la opción Descartar
Nombre (s) y apellidos	NA	NA	V	I	NA	NA	NA	NA
Carnet de identidad	NA	NA	V	I	NA	NA	NA	NA
Licencia de conducción	NA	NA	V	I	NA	NA	NA	NA

Capítulo 3. Implementación y Pruebas

Tipo de licencia	NA	NA	V	V	NA	NA	NA	NA
Dirección particular	NA	V	I	V	NA	NA	NA	NA
Estado civil	NA	V	I	V	NA	NA	NA	NA
Cant. accidentes	NA	V	I	I	NA	NA	NA	NA
Fecha de recalificación	NA	NA	V	V	NA	NA	NA	NA
Fecha del chequeo médico	NA	NA	V	V	NA	NA	NA	NA
Fecha de autorizo de circulación	NA	NA	V	V	NA	NA	NA	NA
Herramientas asignadas	NA	V	I	I	NA	NA	NA	NA
Cartilla operativa	NA	V	I	I	NA	NA	NA	NA
Resultado del chequeo médico	NA	NA	V	V	NA	NA	NA	NA
Estado	NA	NA	V	V	NA	NA	NA	NA
Multa	NA	V	I	I	NA	NA	NA	NA
Respuesta del sistema	El sistema debe permitir introducir	El sistema valida y	El sistema señala el o los	Llevar todos los documentos del	El sistema muestra un	El sistema elimina los	El sistema mantiene los	El sistema regresa al listado

Capítulo 3: Implementación y Pruebas

	<p>los siguientes datos para incluir un chofer, de manera obligatoria:</p> <ul style="list-style-type: none"> - Nombre(s) y apellidos del chofer - Carnet de identidad - Licencia de conducción - Tipo de licencia - Dirección particular es de manera opcional - Estado civil es de manera opcional - Cant. de accidentes es de manera opcional - Fecha de recalificación - Fecha del chequeo médico - Fecha de 	<p>guarda los datos de nuevo chofer</p>	<p>campos obligatorios que no hayan sido introducidos y/o seleccionados y muestra el mensaje de información:</p> <p>"Los siguientes campos son obligatorios";</p> <p>indicando los campos en cuestión.</p>	<p>vehículo</p>	<p>mensaje de información:</p> <p>Se ha modificado la entidad, sus cambios se perderán y solicita confirmación ¿Está seguro que quiere abandonar la página? El sistema brinda las siguientes opciones:</p> <ul style="list-style-type: none"> - Aceptar - Cancelar 	<p>datos introducidos y regresa al listado de choferes</p>	<p>cambios realizados.</p>	<p>de choferes.</p>
--	--	---	--	-----------------	--	--	----------------------------	---------------------

Capítulo 3: Implementación y Pruebas

	<p>autorizo de circulación</p> <ul style="list-style-type: none"> - Herramientas asignadas al vehículo es de manera opcional - Cartilla operativa es de manera opcional -Resultado del chequeo médico - Estado - Multa es de manera opcional <p>Y brinda las siguientes opciones:</p> <ul style="list-style-type: none"> - Guardar o Descartar 							
Respuesta del sistema	Módulo Transporte / Choferes/Crear/ Guardar	Módulo Transporte / Choferes/Crear/ Guardar	Módulo Transporte / Choferes/Crear/ Guardar	Módulo Transporte / Choferes/Crear/ Guardar	Módulo Transporte / Choferes/Crear/ Descartar	Módulo Transporte /Choferes/Crear/Descartar/Aceptar	Módulo Transporte / Choferes/Crear/Descartar/Cancelar	Módulo Transporte / Choferes/Crear/Descartar

Capítulo 3: Implementación y Pruebas

1.4.4 Resultados de las pruebas

Se realizaron pruebas de caja negra en 3 iteraciones, obteniéndose en la primera iteración 38 no conformidades, en la segunda iteración se detectaron 9 no conformidades y en la tercera iteración el sistema estaba libre de no conformidades, cumpliendo con los requisitos funcionales especificados. Estas fueron agrupadas en la siguiente tabla siendo evaluadas en un rango comprendido entre: Alta, Media, Baja y No procede. A continuación se presentan los resultados arrojado por estas pruebas en la 1ra iteración:

Tabla: 4 No Conformidades del Módulo Transporte 1ra iteración

No. CP	Caso de prueba	No conformidades				
		Alta	Media	Baja	No procede	Total
1	CP Incluir chofer	-	-	3	-	3
2	CP Modificar chofer	-	-	-	-	-
3	CP Ver chofer	-	-	-	-	-
4	CP Eliminar chofer	-	-	-	-	-
5	CP Listar chofer	-	-	1	-	1
6	CP Filtrar listado de choferes	-	-	2	-	2
7	CP Incluir vehículo	-	-	3	-	3
8	CP Modificar vehículo	-	-	-	-	-
9	CP Ver vehículo	-	-	-	-	-
10	CP Eliminar vehículo	-	-	-	-	-
11	CP Listar vehículo	-	-	1	-	1
12	CP Filtrar listado de vehículos	-	-	2	-	2
13	CP Incluir anexo 1	-	-	3	-	3
14	CP Modificar anexo 1	-	-	-	-	-
15	CP Ver anexo 1	-	-	-	-	-
16	CP Eliminar anexo 1	-	-	-	-	-
17	CP Listar anexo 1	-	-	1	-	1
18	CP Filtrar listado de anexos 1	-	-	2	-	2
19	CP Incluir hoja de ruta	-	-	3	-	3
20	CP Modificar hoja de ruta	-	-	-	-	-
21	CP Ver hoja de ruta	-	-	-	-	-

Capítulo 3: Implementación y Pruebas

22	CP Eliminar hoja de ruta	-	-	-	-	-
23	CP Listar hojas de ruta	-	-	1	-	1
24	CP Filtrar listado de hojas de ruta	-	-	2	-	2
25	CP Incluir herramienta	-	-	-	-	-
26	CP Modificar herramienta	-	-	-	-	-
27	CP Ver herramienta	-	-	-	-	-
28	CP Eliminar herramienta	-	-	-	-	-
29	CP Listar herramientas	-	-	-	-	-
30	CP Filtrar listado de herramientas	-	-	-	-	-
31	CP Incluir reparación	-	-	1	-	1
32	CP Modificar reparación	-	-	-	-	-
33	CP Ver reparación	-	-	-	-	-
34	CP Eliminar reparación	-	-	-	-	-
35	CP Listar reparaciones	-	-	1	-	1
36	CP Filtrar listado de reparaciones	-	-	2	-	2
37	CP Incluir autorizo de parqueo	-	-	-	-	-
38	CP Modificar autorizo de parqueo	-	-	-	-	-
39	CP Ver autorizo de parqueo	-	-	-	-	-
40	CP Eliminar autorizo de parqueo	-	-	-	-	-
41	CP Listar autorizos de parqueo	-	-	-	-	-
42	CP Filtrar listado de autorizos de parqueo	-	-	-	-	-
43	CP Incluir control sorpresivo	-	-	1	-	1
44	CP Modificar control sorpresivo	-	-	-	-	-
45	CP Ver control sorpresivo	-	-	-	-	-
46	CP Eliminar control sorpresivo	-	-	-	-	-
47	CP Listar controles sorpresivos	-	-	1	-	1
48	CP Filtrar listado de controles sorpresivos	-	-	1	-	1
49	CP Incluir multa	-	-	2	-	2
50	CP Modificar multa	-	-	-	-	-
51	CP Ver multa	-	-	-	-	-
52	CP Eliminar multa	-	-	-	-	-
53	CP Listar multas	-	-	1	-	1
54	CP Filtrar listado de multas	-	-	1	-	1
55	CP Incluir gasto	-	-	1	-	1

Capítulo 3: Implementación y Pruebas

56	CP Modificar gasto	-	-	-	-	-
57	CP Ver gasto	-	-	-	-	-
58	CP Eliminar gasto	-	-	-	-	-
59	CP Listar gastos	-	-	1	-	1
60	CP Filtrar listado de gastos	-	-	1	-	1
61	CP Incluir marca	-	-	-	-	-
62	CP Eliminar marca	-	-	-	-	-
63	CP Ver marca	-	-	-	-	-
64	CP Listar marcas	-	-	-	-	-
Total		-	-	38	-	38

Para verificar las No Conformidades (NC) detectadas se presenta una tabla con los siguientes datos: el número de caso de prueba (No. CP), NC detectada, tipo de NC (clasificada en validación, funcionalidad, correspondencia con otro artefacto, errores de idioma, ortografía, redacción, recomendación, error de caso de prueba) y estado con respecto a la Solución (RA: Resuelta y Aprobada por el revisor, PD: Pendiente por solución del equipo de desarrollo, NP: No Procede, AV: Aplazada para resolver en próximas versiones, NR: Nuevo Requisito). Para ver más información sobre las demás iteraciones ver **Anexo # 10**.

Tabla: 5 Descripción de No Conformidades 1ra iteración

No. NC	No. CP	Descripción	Complejidad	Estado
1	1	Correspondencia con otro artefacto	Baja	RA
2	1	Ortografía	Baja	RA
3	1	Redacción	Baja	RA
4	5	Correspondencia con otro artefacto	Baja	RA
5	6	Error de caso de prueba	Baja	RA
6	6	Correspondencia con otro artefacto	Baja	RA
7	7	Correspondencia con otro artefacto	Baja	RA
8	7	Ortografía	Baja	RA
9	7	Redacción	Baja	RA

Capítulo 3: Implementación y Pruebas

10	11	Correspondencia con otro artefacto	Baja	RA
11	12	Error de caso de prueba	Baja	RA
12	12	Correspondencia con otro artefacto	Baja	RA
13	13	Correspondencia con otro artefacto	Baja	RA
14	13	Ortografía	Baja	RA
15	13	Redacción	Baja	RA
16	17	Correspondencia con otro artefacto	Baja	RA
17	18	Error de caso de prueba	Baja	RA
18	18	Correspondencia con otro artefacto	Baja	RA
19	19	Correspondencia con otro artefacto	Baja	RA
20	19	Ortografía	Baja	RA
21	19	Redacción	Baja	RA
22	23	Correspondencia con otro artefacto	Baja	RA
23	24	Error de caso de prueba	Baja	RA
24	24	Correspondencia con otro artefacto	Baja	RA
25	31	Correspondencia con otro artefacto	Baja	RA
26	35	Correspondencia con otro artefacto	Baja	RA
27	36	Error de caso de prueba	Baja	RA
28	36	Correspondencia con otro artefacto	Baja	RA
29	43	Redacción	Baja	RA
30	47	Error de caso de prueba	Baja	RA
31	48	Correspondencia con otro artefacto	Baja	RA
32	49	Error de caso de prueba	Baja	RA
33	49	Correspondencia con otro artefacto	Baja	RA
34	53	Correspondencia con otro artefacto	Baja	RA
35	54	Correspondencia con otro artefacto	Baja	RA
36	55	Correspondencia con otro artefacto	Baja	RA

Capítulo 3: Implementación y Pruebas

37	59	Redacción	Baja	RA
38	60	Error de caso de prueba	Baja	RA

3.5 Conclusiones parciales

En este capítulo se evidencian varios aspectos fundamentales que fueron necesarios para la implementación del sistema y su validación. Se describieron los estándares de código utilizados, que facilitaron el trabajo del programador a la hora de implementar el sistema. Con la modelación del diagrama de componentes se logró tener una vista general de todas las dependencias y funcionalidades necesarias para el correcto funcionamiento del módulo. La realización de pruebas funcionales a la solución desarrollada, permitió encontrar no conformidades, dando la posibilidad de corregirlas a tiempo, obteniendo un módulo que responde completamente a los requisitos funcionales existentes.

Conclusiones Generales

Con la culminación del presente trabajo se ha dado cumplimiento a los objetivos trazados en la investigación, obteniéndose como resultado principal, un Módulo para la gestión del Transporte. A continuación se exponen las conclusiones generales a las que fueron arribadas luego del desarrollo de la propuesta de solución:

- El estudio realizado como parte de la investigación sirvió de apoyo en la toma de decisiones con vista al desarrollo del sistema, demostrando que las soluciones similares no cumplen con los requerimientos de la solución deseada, por lo que quedaron descartados para ser utilizados.
- El levantamiento de información sirvió de guía para el posterior análisis y elaboración de la propuesta de solución. Además, los artefactos generados correspondientes a la metodología de desarrollo AUP en su versión UCI facilitaron la implementación del Módulo para la gestión del Transporte.
- La implementación del Módulo Transporte permitió darle solución a la problemática planteada garantizando la informatización de la gestión del transporte en la empresa.
- Las pruebas realizadas, permitieron garantizar la calidad del software y comprobar que los requerimientos definidos en el sistema son los que realmente el cliente desea.

Recomendaciones:

Con el desarrollo de futuras investigaciones se recomienda perfeccionar e incorporar nuevas funcionalidades para lograr mejoras superiores en la gestión de la información dentro del objeto de estudio de la presente investigación, entre las cuales se pueden considerar las siguientes:

- Agregar al módulo una funcionalidad para la gestión del combustible.
- Agregar al módulo una funcionalidad que permita llevar a cabo el control sobre los planes de trabajo de los choferes.

Bibliografía referenciada

1. Definición de sistema - Qué es, Significado y Concepto. [online]. 2014 [Accessed 19 January 2016]. Available from: <http://definicion.de/sistema/>
2. Significado de Sistema - Qué es un, Concepto y Definición. 2010 [online]. [Accessed 19 January 2016]. Available from: <http://www.significados.com/sistema/>
3. Definición de CSS. [online]. 2014 [Accessed 19 January 2016]. Available from: <http://definicion.de/css/>
4. Significado de Gestión - Qué es, Concepto y Definición. [online]. 2010 [Accessed 19 January 2016]. Available from: <http://www.significados.com/gestion/>
5. FEDERICO ALONSO ATEHORTUA HURTADO, RAMON ELIAS BUSTAMANTE VELEZ, JORGE ALBERTO VALENCIA DE LOS RÍOS. Sistema de gestión integral. Una sola gestión, un solo equipo - Federico Alonso Atehortua Hurtado, Ramon Elias Bustamante Velez, Jorge Alberto Valencia de los Rios - Google Libros. [online]. 2008. [Accessed 3 February 2016]. Available from: <https://books.google.com.cu/books?Páginas254>
6. Un sistema de gestión de transporte | El Diccionario de Transporte. [online]. 2016 [Accessed 3 February 2016]. Available from: <http://www.timocom.es/?lexicon=1503241125198229%7CUn%20sistema%20de%20gesti%C3%B3n%20de%20transporte%7CEl%20Diccionario%20de%20Transporte>
7. RICARDO MONTAÑO BADILLA. Sistema ERP. Definición, funcionamiento, ventajas y desventajas • GestioPolis. [online]. 10 February 2010. [Accessed 26 January 2016]. Available from: <http://www.gestiopolis.com/sistema-erp-definicion-funcionamiento-ventajas-desventajas/>
8. ERP Transporte TM | Software Gestión De Transporte. [online]. 2015 [Accessed 15 March 2016]. Available from: <http://www.imagine-cs.com/erp-transporte-tm/>
9. Software ERP para empresas de Transporte y Logística | Galdon Software. [online]. 2013 [Accessed 15 March 2016]. Available from: <http://www.galdon.com/erp-automocion/software-transporte-mercancia-logistica/>
10. DATYS. SOLUCIONES TECNOLÓGICAS. [online]. 2015 [Accessed 23 March 2016]. Available from: <http://www.datys.cu/spa/site/product/19>
11. LISVAN PORRO PIMIENTA. *Sistema de Gestión de Mantenimiento Vehicular v1.0: Diseño e Implementación del proceso de Administración de Inspecciones Técnicas y Accidentes*. [online]. 2012. [Accessed 26 January 2016]. Available from: http://repositorio_institucional.uci.cu/jspui
12. ROBERTH G. FIGUEROA, Armando A. Cabrera. *Metodologías Tradicionales VS. Metodologías Ágiles* [online] 2010 . Universidad Técnica Particular de Loja, Escuela de Ciencias en Computación. Available from:

- http://eva.uci.cu/file.php/161/Documentos/Materiales_complementarios/UD_1_Procesos/Metodologias/METODOLOGIAS_TRADICIONALES_VS._METODOLOGIAS_AGILES.pdf Páginas 30
13. JUAN SALVADOR. METODOLOGIAS AGILES AUP | Academia.edu. [online]. 2016. [Accessed 20 January 2016]. Available from: http://www.academia.edu/7894130/METODOLOGIAS_AGILES_AUP
 14. ROGER S. PRESSMAN. *Ingeniería de Software, un enfoque práctico*. Quinta Edición. S.I. : McGraw-Hill Companies, 2010. Páginas 325
 15. TAMARA RODRÍGUEZ and SÁNCHEZ. *Metodología de desarrollo para la Actividad productiva de la UCI*. 12 November 2014.
 16. LEANDRO ALEGA. Definición de UML. [online]. 26/11/2008 [Accessed 20 January 2016]. Available from: <http://www.alegsa.com.ar/Dic/uml.php>
 17. JOSE ENRIQUE GONZALEZ CORNEJO. El Lenguaje de Modelado Unificado (UML). [online]. January 2008 [Accessed 20 January 2016]. Available from: <http://www.docirs.com/uml.htm>
 18. Visual Paradigm para UML. [online]. 2009 [Accessed 4 February 2016]. Available from: <http://www.visual-paradigm.com/news/vpsuite40/vpum170.jsp>
 19. XML.com. [online]. March 2010 .2 May 2016. [Accessed 2 March 2016]. Available from: <http://www.xml.com/>
 20. GUIDO VAN ROSSUM. TutorialPython2.pdf. [online]. September 2009. [Accessed 23 March 2016]. Available from: <http://docs.python.org.ar/tutorial/pdfs/TutorialPython2.pdf>. Páginas 67.
 21. Welcome to Python.org. [online]. October 2011. 2 March 2016. [Accessed 2 March 2016]. Available from: <https://www.python.org/>
 22. Odo – OpenERP – ERP, CRM, MRP, SGA 100% Libre – | Sin Licencias Portal OpenErp (Odo) en España. [online]. 2 March 2014 [Accessed 29 February 2016]. Available from: <http://openerpspain.com/>
 23. Odo – OpenERP – ERP, CRM, MRP, SGA 100% Libre – | Sin Licencias Funcionalidades de OpenERP - OpenERP | Odo. [online]. 2 March 2014 [Accessed 29 February 2016]. Available from: <http://openerpspain.com/funcionalidades/>
 24. Odo – OpenERP Portal OpenErp (Odo) en España. [online]. 2 March 2014 [Accessed 29 February 2016]. Available from: <http://openerpspain.com/>
 25. PyCharm. [online]. October 2000 [Accessed 23 March 2016]. Available from: <https://www.jetbrains.com/pycharm/>
 26. PyCharm: El mejor IDE para tus proyectos en Python. [online]. 16 September 2014 [Accessed 23 March 2016]. Available from: <http://www.cristalab.com/tutoriales/pycharm-el-mejor-ide-para->

tus-proyectos-en-python-c114084/

27. Definición de SGBD. [online]. 2014 [Accessed 4 February 2016]. Available from: <http://www.alegsa.com.ar/Dic/sghd.php>
28. PostgreSQL-es. [online]. 2009 [Accessed 4 February 2016]. Available from: <http://www.postgresql.org.es/>
29. PostgreSQL 9.2.* · Comunidad de PostgreSQL. [online]. 2013 [Accessed 20 May 2016]. Available from: https://postgresql.uci.cu/?page_id=179
30. POSTGRESQL. About PostgreSQL Global Development Group. [online]. 9 January 2016 [Accessed 4 February 2016]. Available from: <http://www.postgresql.org/about/>
31. Definición de Requerimientos (desarrollo de sistemas/software). [online]. [Accessed 23 March 2016]. Available from: <http://www.alegsa.com.ar/Dic/requerimientos.php>
32. CARLOS REYNOSO, – Nicolás Kicillof. Estilos y Patrones en la Estrategia de Arquitectura de Microsoft. *Estilos y Patrones en la Estrategia de Arquitectura de Microsoft* [online]. March 2004. [Accessed 23 March 2016]. Available from: <http://carlosreynoso.com.ar/archivos/arquitectura/Estilos.PDF> UNIVERSIDAD DE BUENOS AIRES
33. Odoo – OpenERP – ERP, CRM, MRP, SGA 100% Libre – | Sin Licencias Características Técnicas de OpenErp 8.0 (Odoo). [online]. [Accessed 23 March 2016]. Available from: <http://openerpspain.com/openerp/caracteristicas-tecnicas/>
34. JUAN PAVÓN MESTRAS. Patrones de diseño. [online]. 2004. [Accessed 23 March 2016]. Available from: <http://www.fdi.ucm.es/profesor/jpavon/poo/2.14pdoo.pdf>
35. DANIEL JOSÉ SALAS ÁLVAREZ. Estándares de codificación Java. [online]. 2010. [Accessed 3 March 2016]. Available from: <http://www.aves.edu.co/ovaunicor/recursos/view/265>.
36. UML 2 Component Diagramming Guidelines. [online]. [Accessed 28 April 2016]. Available from: <http://www.agilemodeling.com/style/componentDiagram.htm>