

Universidad de Cienfuegos “Carlos Rafael Rodríguez”  
Facultad de Ingeniería  
Carrera de Ingeniería Informática



## **SISTEMA INFORMÁTICO PARA EL CONTROL DEL TRANSPORTE EN LA UNIVERSIDAD DE CIENFUEGOS.**

**Trabajo de diploma para optar por el título de Ingeniería en  
Informática**

**Autor:**

Jorge Luis de Armas Díaz.

**Tutores:**

**Msc.** Hugandy Álvarez Acosta.

**Ing.** Annelys Gato Saura.

**Cienfuegos, Cuba  
Curso 2011 – 2012**

## Resumen

La presente investigación tiene como finalidad la elaboración de un sistema informático capaz de gestionar los datos relacionados con el control del transporte en la Universidad de Cienfuegos “Carlos Rafael Rodríguez”, el cual fundamenta la idea de alcanzar una mejor gestión de la información y lograr elevar la rapidez del proceso.

El sistema propuesto tiene como nombre “Sistema Informático para el control del transporte en la Universidad de Cienfuegos”. Permite la gestión de los datos de cada vehículo satisfactoriamente así como el combustible que se le asigna a cada uno y los viajes que realizan y facilita que el acceso, manipulación y compartición de los datos sean fáciles y seguros. A través del documento de la investigación quedan descritos los elementos que conforman el análisis, diseño e implementación del sistema propuesto, siguiendo lo establecido por la metodología Desarrollo Guiado por Funcionalidades y utilizando el Lenguaje Unificado de Modelado. Para la implementación del mismo se utilizó MySQL como Sistema Gestor de Bases de Datos, Apache como servidor web, PHP como lenguaje de programación y el framework CodeIgniter.

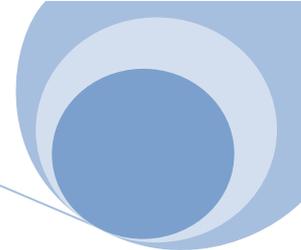
El software desarrollado cumple con las necesidades reales de los clientes y usuarios finales y contribuye a que toda la información sea accesible. Obtiene además, una reducción del tiempo en las búsquedas de información. Por otra parte, permite mejorar las condiciones de trabajo del personal, evitándoles el agotamiento y demora que produce el procesamiento manual de la información al contribuir positivamente en el almacenamiento y control de ésta.

# Índice

Introducción.....	6
Capítulo 1 .....	11
1.1    Introducción.....	11
1.2    Sistemas automatizados existentes vinculados al campo de acción ..	11
1.3    Metodologías de desarrollo .....	13
1.4    FDD.....	14
1.5    Lenguaje de modelado, UML .....	14
1.6    Herramienta de Modelado, Visual Paradigm.....	15
1.7    Situación actual de la institución .....	15
1.8    Aplicaciones web.....	16
1.9    Elementos de arquitectura .....	17
1.10   Servidor web, Apache .....	17
1.11   Herramienta para el almacenamiento de la información, MySQL ....	18
1.12   Lenguaje utilizado del lado del servidor, PHP.....	18
1.13   Framework del lado del servidor, CodeIgniter.....	19
1.14   Lenguaje utilizado del lado del cliente, JavaScript.....	19
1.15   Framework del lado del cliente, Dojo .....	20
1.16   Entorno de desarrollo integrado, NetBeans .....	21
1.17   Herramienta de diseño gráfico, GIMP.....	22
1.18   Conclusiones .....	22
Capítulo 2.....	24
2.1    Introducción.....	24
2.2    Desarrollo de un modelo global.....	24
2.2.1  Flujo actual de los procesos involucrados en el campo de acción.	24
2.2.2  Objeto de automatización. ....	27

2.2.3	Modelo de dominio. Diagrama de objetos. ....	28
2.2.4	Reglas a considerar. ....	29
2.3	Construcción de una lista de funcionalidades. ....	29
2.3.1	Requerimientos funcionales. ....	29
2.3.2	Requerimientos no funcionales. ....	32
2.4	Estudio de la factibilidad. ....	34
2.4.1	Beneficios tangibles e intangibles. ....	40
2.4.2	Análisis de costos y beneficios. ....	40
2.5	Planeación por funcionalidad. ....	40
2.6	Diseño por funcionalidad. ....	42
2.6.1	Principios de diseño. ....	42
2.6.2	Modelo conceptual de la base de datos. ....	43
2.6.3	Modelo lógico de la base de datos. ....	44
2.6.4	Modelo físico de la base de datos. ....	44
2.7	Conclusiones. ....	45
Capítulo 3	.....	46
3.1	Introducción. ....	46
3.2	Descripción de la arquitectura propuesta. ....	46
3.3	Diagrama de clases. ....	47
3.4	Estándar de implementación. ....	48
3.5	Seguridad del sistema. ....	49
3.6	Tratamiento de errores. ....	49
3.7	Diagrama de despliegue. ....	50
3.8	Validación de la solución propuesta. ....	50
3.8.1	Resultados estadísticos de las mediciones realizadas. ....	51
3.9	Conclusiones. ....	55
Conclusiones	.....	56

Recomendaciones.....	57
Referencias Bibliográficas.....	58



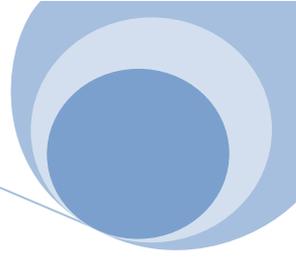
## **Introducción**

El desarrollo de las tecnologías de la información y las comunicaciones (TIC) han provocado un gran impacto en la sociedad. Las TIC han estimulado la ocurrencia de cambios trascendentales en las formas de acceder al conocimiento y de ejecutar la comunicación.[1]

El Ministerio de Educación Superior de Cuba ha puesto esencial interés y preocupación en llevar a todos los Centros Universitarios, los adelantos más actualizados de las TIC, al servicio de profesores y estudiantes, con el objetivo de lograr un egresado con mente abierta y gran creatividad capaz de enfrentar los retos que ofrece la sociedad de estos tiempos. Además el empleo de los diferentes productos informáticos constituye un eslabón esencial pues eleva el interés en los estudiantes, profesores y trabajadores por la investigación científica y posibilita el mejoramiento de las habilidades creativas, la imaginación, habilidades comunicativas y colaborativas pudiendo acceder a mayor cantidad de información y proporcionando los medios para un mejor desarrollo integral de los individuos. [1]

La Universidad de Cienfuegos “Carlos Rafael Rodríguez” (UCF), con un colectivo altamente comprometido con la Revolución, garantiza la formación integral y la superación continúa de los profesionales revolucionarios que demanda la sociedad desde el contexto cienfueguero con calidad y pertinencia. Consolida, desarrolla y promueve la ciencia, la cultura y la innovación tecnológica, en correspondencia a las necesidades del desarrollo sostenible en la universalización de la universidad y en los marcos de los Programas de la Revolución.[2]

La UCF está liderada por el rectorado y posee varias áreas. Entre estas aéreas o departamentos se encuentra el departamento de transporte que atiende el control del transporte dentro de la universidad.



Dicho departamento realiza las siguientes funciones:

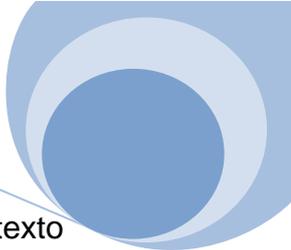
- Control de los recursos asignados a los vehículos de la universidad en caso de roturas.
- Planificación mensual de transportación.
- Transportación tanto de personal como de carga.

El control de los recursos asignados para los vehículos de la Universidad en caso de mantenimientos o roturas se registra mediante la confección de órdenes de trabajo. En este documento se describe el tipo de trabajo a realizar y se enumeran los materiales y piezas a solicitar al almacén. Posteriormente de ser confeccionada esta orden de trabajo se realiza la solicitud de materiales para el almacén de Abastecimiento Técnico Material (ATM). Luego de este procedimiento el mecánico cuenta con los materiales y herramientas necesarias y repara o le da mantenimiento al carro e informa al técnico del departamento el nuevo estado técnico de dicho vehículo.

En el departamento cada mes se elabora el plan de transportación. En este plan se organizan las actividades que son planificadas por la dirección del centro en el plan de trabajo del mes junto con la cantidad de combustible que se presume se pueda consumir en ellas. Estas actividades pueden ser diarias, semanales o por intervalos de días, además pueden tener diferentes destinos. Asimismo puede ocurrir que el combustible planificado para las mismas varíe en dependencia de la disponibilidad de vehículos del departamento.

La transportación tanto de personal como de carga es otra de las funciones del departamento. La transportación del personal se realiza mediante solicitudes que hacen los administradores de cada facultad o área de la Universidad, mientras que el transporte de carga es menos formal y se realiza por una petición telefónica.

Estos datos alcanzan un volumen considerable, que actualmente están guardados en Documentos Excel y Documentos Word. Se necesita además generar reportes, resúmenes e informes mensuales. Asimismo se precisa que esta información esté disponible para que pueda ser consultada por todas las



personas que laboran en las distintas áreas del centro. Todo este contexto **define la problemática** existente.

### **Problema Científico**

La necesidad de controlar la gestión de la información relacionada con el proceso del control del transporte en la Universidad de Cienfuegos de forma digital.

### **Idea a Defender**

La elaboración de una aplicación informática para la gestión de la información relacionada con el control del transporte en la Universidad de Cienfuegos posibilitará una mayor rapidez en el proceso.

### **Objeto de Estudio**

El control del transporte en la Universidad de Cienfuegos.

### **Campo de Acción**

La gestión de la información relacionada con el control del transporte en la Universidad de Cienfuegos.

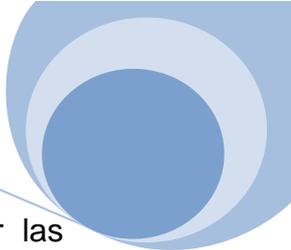
### **Objetivo General**

Elaborar una aplicación informática para la gestión de la información relacionada con el control del transporte en la Universidad de Cienfuegos.

### **Objetivos Específicos**

- Realizar un análisis para la confección del diseño de la aplicación informática que dará solución a la problemática planteada.
- Diseñar la aplicación informática propuesta.
- Implementar una aplicación que se ajuste a las particularidades del Departamento de Transporte de la Universidad de Cienfuegos.
- Validar la aplicación propuesta.

### **Tareas**

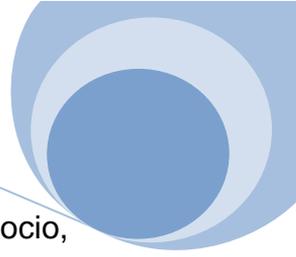
- 
- Caracterización el problema y la situación actual para identificar las particularidades del proceso relacionado al control del transporte en la UCF.
  - Entrevistas a directivos y trabajadores del departamento de transporte para conocer los principales procesos que tienen lugar en la realización de la gestión de los datos.
  - Recopilación los principales aportes teóricos relacionados con los sistemas automatizados existentes asociados al problema.
  - Realizar un estudio de las tendencias y tecnologías actuales del campo de la informática, determinando cuáles utilizar en la solución del problema.
  - Selección de las tecnologías actuales posibles a emplear para dar solución al problema existente, teniendo en cuenta las particularidades del área.
  - Diseño de la interfaz de la aplicación.
  - Análisis y diseñar la base de datos que contendrá la información necesaria para informatizar el control del transporte en la UCF.
  - Instalación del software para su etapa de prueba.

### **Aporte Práctico**

La implementación de la aplicación informática proporciona una vía más rápida para gestionar y organizar el volumen de información referente al campo de acción.

La tesis está estructurada en tres capítulos, además de conclusiones, recomendaciones, referencias bibliográficas, bibliografías utilizadas y los anexos.

**Capítulo 1. Fundamentación Teórica:** Se abordan los aspectos teóricos que se necesitan dominar en la investigación, se analiza el objeto de estudio, los sistemas existentes vinculados a la investigación, así como las tendencias, tecnologías y metodologías actuales a emplear y una explicación de por qué su uso.



**Capítulo 2. Descripción del sistema:** Se describen los procesos del negocio, se enumeran las funcionalidades previstas y los requerimientos no funcionales. Se tienen en cuenta los principios de diseño para crear la interfaz de la aplicación. Se elabora el modelo conceptual de la base de datos, el modelo lógico y físico. Por último se realiza el estudio de la factibilidad.

**Capítulo 3. Construcción del sistema:** En este capítulo se realiza una descripción de la construcción de la solución propuesta haciendo uso de la descripción de la arquitectura y los diagramas de clases por cada capa. Se aborda todo lo relacionado con la seguridad del sistema y el tratamiento de errores. También se realiza la validación del sistema.

Así como los anexos que complementan la información presentada en dichos capítulos.

## 1.1 Introducción

En este capítulo se realiza un estudio sobre los principales conceptos asociados al dominio del problema y sobre los sistemas automatizados existentes vinculados al campo de acción, efectuando una comparación entre ellos. Se desarrolla además un análisis del objeto de estudio y la solución problemática. Concluye con el estudio sobre metodologías, tecnologías y herramientas que se pueden emplear para la elaboración del sistema.

## 1.2 Sistemas automatizados existentes vinculados al campo de acción

En la actualidad el gran avance de las tecnologías de la informática y las comunicaciones, ha permitido desarrollar infinidad de aplicaciones de gestión relacionadas con el control y la recuperación de la información en todo el mundo.

El objetivo principal de dichas aplicaciones es el de almacenar grandes cantidades de información y asegurarse de que su recuperación se efectúe de la forma más rápida y óptima.

La presente investigación pretende enfocarse en la rama del control del transporte en la cual se han desarrollado ya varias aplicaciones. En el ámbito internacional resalta Softflot.

- **Softflot:** Diseñado especialmente para administrar flotillas de cualquier clase de vehículos como puede ser Transporte Público, Transporte Pesado, Transporte de Carga, Auto Tanques Utilitarios, Mudanzas, Automóviles en Arrendamiento, Motocicletas, Maquinaria para Construcción, Montacargas, Retroexcavadoras, etc. [3]

# Capítulo 1. Fundamentación Teórica

- **DPC SmartFleet:** es un sistema que gestiona la aplicación del transporte para aumentar eficacia de dirección rápida, y mejorar la visibilidad de inventario móvil y dirección del recurso. Integra el Sistema del Posicionamiento Global (GPS).
- **QAD Enterprise Applications:** son una colección completa de productos con funcionalidad que se dirige las necesidades de organizaciones de la multinacional. Fue diseñado con funcionamientos para las cadenas de suministro, finanzas, clientes y tecnología. Las soluciones de QAD proporcionan a los fabricantes el acceso fácil a la información que ellos necesitan planear para el futuro.
- **Transportation Solution:** es un sistema diseñado para el control de combustible en empresas que cuenten con una serie de vehículos. Es capaz de realizar cualquier tipo de operaciones con las tarjetas de combustible. Imprime informes mensuales, anuales y lleva toda la contabilidad del tráfico de combustible en una empresa.

Entre los software desarrollados en Cuba relacionados al campo de acción que ameritan nombrarse se encuentra el Tera.

Tera es un sistema diseñado especialmente para la Empresa de Telecomunicaciones de Cuba S.A (ETECSA). Su objetivo es llevar el control de los vehículos de la empresa así como la distribución de los insumos asociados a estos en caso de roturas.

Cuenta con una base de datos diseñada sobre PostgreSQL que cumple con requisitos específicos de ETECSA. [4]

Los sistemas internacionales antes mencionados son software privados, aplicaciones de escritorio dependientes de la plataforma Windows y aunque manejan la gestión de la información relacionada con el transporte en aspectos como la recopilación de los datos, control de mantenimientos y roturas, no se ajustan a las necesidades específicas del departamento de transporte de la Universidad. Pues se necesita una aplicación multiplataforma, donde se aproveche las potencialidades de las aplicaciones Web, teniendo en cuenta los

# Capítulo 1. Fundamentación Teórica

diferentes puntos de acceso y cantidad de estaciones de trabajo que van a acceder al sistema.

Por otra parte se requiere de la elaboración del plan de transportación, de la recepción, revisión y aprobación de cada una de las solicitudes en cada nivel de dirección para luego conformar la lista de viajes de cada semana y ninguno de los software disponibles poseen algunas de estas opciones.

## 1.3 Metodologías de desarrollo

La siguiente tabla muestra una breve comparación entre metodologías de desarrollo ágiles y metodologías de desarrollo pesadas.

<b>Metodologías Ágiles</b>	<b>Metodologías Pesadas</b>
Basadas en heurísticas provenientes de prácticas de producción de código	Basadas en normas provenientes de estándares seguidos por el entorno de desarrollo
Especialmente preparados para cambios durante el proyecto	Especialmente preparados para cambios durante el proyecto
Impuestas internamente (por el equipo)	Impuestas externamente
Proceso menos controlado, con pocos principios	Proceso mucho más controlado, con numerosas políticas/normas
No existe contrato tradicional o al menos es bastante flexible	Existe un contrato prefijado
El cliente es parte del equipo de desarrollo	El cliente interactúa con el equipo de desarrollo mediante reuniones
Grupos pequeños (<10 integrantes) y trabajando en el mismo sitio	Grupos grandes y posiblemente distribuidos
Pocos artefactos	Más artefactos
Pocos roles	Más roles
Menos énfasis en la arquitectura del software	La arquitectura del software es esencial y se expresa mediante modelos

Tabla 1.1 Comparación entre metodologías.

# Capítulo 1. Fundamentación Teórica

## 1.4 FDD

FDD es un proceso diseñado por Peter Coad, Erich Lefebvre y Jeff de Luca y se podría considerar a medio camino entre RUP y XP, aunque al seguir siendo un proceso ligero es más similar a este último.[5]

FDD está pensado para proyectos con tiempo de desarrollo relativamente corto (menos de un año). Se basa en un proceso iterativo de iteraciones cortas que producen un software funcional que el cliente y la dirección de la empresa pueden ver y monitorizar.[5]

Las iteraciones se deciden en base a *features* (de ahí el nombre del proceso) o funcionalidades, que son pequeñas partes del software con significado para el cliente. [5]

Un proyecto que sigue FDD se divide en 5 fases:

- Desarrollo de un modelo general.
- Construcción de una lista de funcionalidades.
- Plan de reléase en base a las funcionalidades a implementar.
- Diseñar en base a las funcionalidades.
- Implementar en base a las funcionalidades.[5]

## 1.5 Lenguaje de modelado, UML

**El lenguaje unificado de modelado (UML)**, es un lenguaje para la especificación, visualización, construcción y documentación de los artefactos de un proceso de sistema intensivo. [6]

UML está consolidado como el lenguaje estándar en el análisis y diseño de sistemas de cómputo. Mediante UML es posible establecer la serie de requerimientos y estructuras necesarias para plasmar un sistema de software previo al proceso intensivo de escribir código.[6]

En otros términos, así como en la construcción de un edificio se realizan planos previos a su construcción, en Software se deben realizar diseños en UML previa codificación de un sistema. Ahora bien, aunque UML es un lenguaje, éste posee más características visuales que programáticas, las que facilitan

# Capítulo 1. Fundamentación Teórica

a los integrantes de un equipo multidisciplinario participar e intercomunicarse fácilmente, estos integrantes siendo los analistas, diseñadores, especialistas de área y desde luego los programadores.[6]

## 1.6 Herramienta de Modelado, Visual Paradigm

Visual Paradigm para UML es una herramienta UML profesional que soporta el ciclo de vida completo del desarrollo de software: análisis y diseño orientados a objetos, construcción, pruebas y despliegue. El software de modelado UML ayuda a una más rápida construcción de aplicaciones de calidad, mejores y a un menor coste. Permite dibujar todos los tipos de diagramas de clases, código inverso, generar código desde diagramas y generar documentación. La herramienta UML CASE también proporciona abundantes tutoriales de UML, demostraciones interactivas de UML y proyectos UML.[7]

## 1.7 Situación actual de la institución

En la Universidad de Cienfuegos los administradores de las diferentes áreas son los encargados de elaborar las solicitudes de transporte. Estas solicitudes deben ser aprobadas y revisadas por los jefes inmediatos superiores de cada área, los que luego de dar su visto bueno las envían al vicerrectorado. En el vicerrectorado todas las solicitudes pasan por un proceso de revisión donde son aprobadas o no. Las que resultan ser aprobadas pasan a formar parte del plan de transportación mensual junto con las actividades previstas en el plan de trabajo.

En la actualidad todo el proceso se realiza manualmente por lo que consume gran cantidad de material de oficina (papel, lapiceros, lápices, marcadores, etc...) esta situación contribuye a la lentitud en el proceso, la información no es segura y no siempre llega a su destino final, a veces por errores humanos. Todo este flujo no es almacenado lo que contribuye a que se pierda gran cantidad de información. El proceso por tanto no es eficiente, la posibilidad de errores es mayor que si el proceso se realizara de manera digital. Además es mucho más difícil el control de cada una de las etapas del proceso por la dirección.

# Capítulo 1. Fundamentación Teórica

Por la situación actual descrita anteriormente surge la necesidad apremiante de construir una aplicación Web que permita el acceso a la información de manera remota para todos los usuarios relacionados con este proceso. Permitiendo además la implantación del sistema garantizar el control estricto de la misma por parte del Vicerrectorado. Se requiere un sistema multiplataforma por el gran número de estaciones de trabajo.

Asimismo es imprescindible un potente gestor de base de datos que se mantenga estable ante la presencia de volúmenes masivos de información. De manera general se precisa de una aplicación ágil, sencilla, interactiva y con gran cantidad de módulos. Se cuenta con un tiempo limitado de desarrollo por lo que se debe hacer una selección correcta de lenguajes a emplear en la creación del software de forma tal que se facilite la rapidez de la implementación.

## 1.8 Aplicaciones web

En la ingeniería de software se denomina aplicación web a aquellas aplicaciones que los usuarios pueden utilizar accediendo a un servidor web a través de Internet o de una intranet mediante un navegador. En otras palabras, es una aplicación software que se codifica en un lenguaje soportado por los navegadores web en la que se confía la ejecución al navegador.

Características:

- Portabilidad de los datos.
- Solución a problemas reales.
- Multiplataforma.
- Deberían tener una visión.
- No deberían molestar al usuario.
- Deberían estar en constante mejora.
- Deberían tener una excelente documentación.
- Deberían ser amistosas.
- Visualmente atractiva.

# Capítulo 1. Fundamentación Teórica

## 1.9 Elementos de arquitectura

El concepto de arquitectura de software está muy extendido, sin embargo no existe una definición única y estándar para este concepto. Un autor, cuya contribución en el mundo de las arquitecturas de software y las componentes ha sido muy aceptada, es David Garlan. Su definición de arquitectura es una de las más utilizadas en la literatura.

“La arquitectura de software está compuesta por la estructura de los componentes de un programa o sistema, sus interrelaciones, y los principios y reglas que gobiernan su diseño y evolución a lo largo del tiempo.”

Por las características del proyecto y por las tendencias de las tecnologías en la actualidad se pretende implementar una arquitectura Modelo - Vista - Controlador y se empleará la programación orientada a objetos pues la aplicación se desarrollará con PHP5, este es un paradigma de programación que usa objetos y sus interacciones para diseñar aplicaciones, está basado en varias técnicas, incluyendo herencia, modularidad, polimorfismo, y encapsulamiento. Se hará uso además de frameworks que organicen toda la estructura de la aplicación y minimicen el tiempo de desarrollo.

### Modelo Vista Controlador (MVC)

Es un patrón de arquitectura de software que separa los datos de una aplicación, la interfaz de usuario, y la lógica de negocio en tres componentes distintos. El patrón se observa frecuentemente en aplicaciones web, donde la vista es la página HTML y el código que provee de datos dinámicos a la página. El modelo es el Sistema de Gestión de Base de Datos y la capa de acceso a datos, y el controlador es el responsable de recibir los eventos de entrada desde la vista y de la lógica del negocio.[8]

## 1.10 Servidor web, Apache

El servidor Web es una herramienta que implementa el protocolo HTTP (Hypertext Transfer Protocol). Está elaborado para traducir lenguajes y sentencias a una interfaz entendible por el usuario, entre ellos: textos

# Capítulo 1. Fundamentación Teórica

complejos con enlaces, figuras, formularios, botones y objetos incrustados como animaciones o reproductores de sonido.

El servidor Apache es un servidor HTTP de código abierto para varias plataformas. Presenta mensajes de error altamente configurables, Base de Datos de Autenticación y negociado de contenidos. Es el servidor HTTP más usado en la actualidad. El mismo es capaz de transformar lenguaje PHP a lenguaje HTML que es el que soporta el navegador con el cual se le presenta la interfaz al usuario.

## **1.11 Herramienta para el almacenamiento de la información, MySQL**

MySQL es uno de los Sistemas Gestores de bases de datos más populares desarrolladas bajo la filosofía de código abierto.

Las principales virtudes del MySQL son su gran velocidad, robustez y facilidad de uso. Fue desarrollado inicialmente para manejar grandes bases de datos mucho más rápidamente que las soluciones existentes y ha sido usado exitosamente por muchos años en ambientes de producción de alta demanda. A través de constante desarrollo, MySQLServer ofrece hoy una rica variedad de funciones.[9]

## **1.12 Lenguaje utilizado del lado del servidor, PHP**

PHP es un lenguaje de programación muy potente que, junto con HTML, permite crear sitios Web dinámicos. PHP se instala en el servidor y funciona con versiones de Apache, MicrosoftIIs, Netscape Enterprise Server y otros.[10]

La forma de usar PHP es insertando código del lenguaje dentro del código HTML de un sitio Web. Cuando un cliente visita la página Web que contiene éste código, el servidor lo ejecuta y el cliente sólo recibe el resultado. Su ejecución, es por tanto en el servidor, a diferencia de otros lenguajes de programación que se ejecutan en el navegador.[10]

PHP permite la conexión a numerosas bases de datos, incluyendo MySQL, Postgres, Oracle, ODBC, IBMDB2, MicrosoftSQLServer y SQLite. PHP tiene la

# Capítulo 1. Fundamentación Teórica

capacidad de ser ejecutado en la mayoría de los sistemas operativos tales como UNIX, Linux, Windows y MacOSX, y puede interactuar con los servidores de Web más populares.[10]

## 1.13 Framework del lado del servidor, CodeIgniter

CodeIgniter es un conjunto de herramientas para personas que construyen su aplicación web usando PHP. Su objetivo es permitirle desarrollar proyectos mucho más rápido de lo que podría si lo escribiese desde cero, proveyéndole un rico juego de librerías para tareas comúnmente necesarias, así como una interface simple y estructura lógica para acceder a esas librerías. CodeIgniter le permite creativamente enfocarse en su proyecto minimizando la cantidad de código necesaria para una tarea dada. [11]

CodeIgniter se encuentra bajo una licencia open source Apache/BSD-style, así que puede ser usado en cualquier lugar.

CodeIgniter está escrito para ser compatible con PHP 4. Aunque hubiese sido mucho mejor tomar ventaja del manejo de objetos en PHP 5 ya que hubiese simplificado algunas cosas, se ha tenido que buscar soluciones creativas. Los principales proveedores de Sistemas Operativos como RedHat se están moviendo lentamente a soportar PHP 5, y es poco probable que lo hagan en el corto plazo, así que no serviría a los mejores intereses de la comunidad de PHP escribir CodeIgniter en PHP 5.[11]

## 1.14 Lenguaje utilizado del lado del cliente, JavaScript

JavaScript es un lenguaje interpretado, al igual que Visual Basic Script, Perl, TCL. (Lenguajes de script) sin embargo, posee una característica que lo hace especialmente idóneo para trabajar en Web, porque es el navegador que soporta la carga de procesamiento. Gracias a su compatibilidad con la mayoría de los navegadores modernos, es el lenguaje de programación del lado del cliente más utilizado. Con JavaScript podemos crear efectos especiales en las páginas y definir interactividades con el usuario. El navegador del cliente es el encargado de interpretar las instrucciones JavaScript y ejecutarlas para realizar estos efectos e interactividades, de modo que el mayor recurso, y tal vez el único, con que

# Capítulo 1. Fundamentación Teórica

cuenta este lenguaje es el propio navegador.[12]

JavaScript es el siguiente paso, después del HTML, que puede dar un programador de la Web que decida mejorar sus páginas y la potencia de sus proyectos. Es un lenguaje de programación bastante sencillo y pensado para hacer las cosas con rapidez, a veces con ligereza. Incluso las personas que no tengan una experiencia previa en la programación podrán aprender este lenguaje con facilidad y utilizarlo en toda su potencia con solo un poco de práctica.

“Javascript es un lenguaje orientado a objetos. El modelo de objetos JavaScript está reducido y simplificado, pero incluye los elementos necesarios para que los Scripts puedan acceder a la información de una página y puedan interactuar sobre la interfaz del navegador”.

Entre las acciones típicas que se pueden realizar en JavaScript tenemos dos vertientes. Por un lado los efectos especiales sobre páginas Web, para crear contenidos dinámicos y elementos de la página que tengan movimiento, cambien de color o cualquier otro dinamismo. Por el otro, JavaScript nos permite ejecutar instrucciones como respuesta a las acciones del usuario, con lo que podemos crear páginas interactivas con programas como calculadoras, agendas, o tablas de cálculo.[12]

## 1.15 Framework del lado del cliente, Dojo

**Dojo** es un framework que contiene APIs y widgets (controles) para facilitar el desarrollo de aplicaciones Web que utilicen tecnología AJAX. Contiene un sistema de empaquetado inteligente, los efectos de UI, drag and drop APIs, widget APIs, abstracción de eventos, almacenamiento de APIs en el cliente, e interacción de APIs con AJAX.

Resuelve asuntos de usabilidad comunes como pueden ser la navegación y detección del navegador, soportar cambios de URL en la barra de URLs para luego regresar a ellas (bookmarking), y la habilidad de degradar cuando AJAX/JavaScript no es completamente soportado en el cliente. Es conocido

# Capítulo 1. Fundamentación Teórica

como "la navaja suiza del ejército de las bibliotecas Javascript". Proporciona una gama más amplia de opciones en una sola biblioteca JavaScript y es compatible con navegadores antiguos.

**jQuery** es una biblioteca de JavaScript, creada inicialmente por John Resig, que permite simplificar la manera de interactuar con los documentos HTML, manipular el árbol DOM, manejar eventos, desarrollar animaciones y agregar interacción con la técnica AJAX a páginas web. Fue presentada el 14 de enero de 2006 en el BarCamp NYC.

jQuery es software libre y de código abierto, posee un doble licenciamiento bajo la Licencia MIT y la Licencia Pública General de GNU v2, permitiendo su uso en proyectos libres y privativos. jQuery, al igual que otras bibliotecas, ofrece una serie de funcionalidades basadas en JavaScript que de otra manera requerirían de mucho más código, es decir, con las funciones propias de esta biblioteca se logran grandes resultados en menos tiempo y espacio.

## 1.16 Entorno de desarrollo integrado, NetBeans

**NetBeans** es un entorno de desarrollo integrado libre, hecho principalmente para el lenguaje de programación Java. Existe además un número importante de módulos para extenderlo. NetBeans IDE es un producto libre y gratuito sin restricciones de uso.[13]

NetBeans es un proyecto de código abierto de gran éxito con una gran base de usuarios, una comunidad en constante crecimiento, y con cerca de 100 socios en todo el mundo. SunMicroSystems fundó el proyecto de código abierto NetBeans en junio de 2000 y continúa siendo el patrocinador principal de los proyectos.[13]

La plataforma NetBeans permite que las aplicaciones sean desarrolladas a partir de un conjunto de componentes de software llamados *módulos*. Un módulo es un archivo Java que contiene clases de java escritas para interactuar con las APIs de NetBeans y un archivo especial (manifest file) que lo identifica como módulo. Las aplicaciones construidas a partir de módulos pueden ser extendidas agregándole nuevos módulos. Debido a que los módulos pueden ser desarrollados independientemente, las aplicaciones

# Capítulo 1. Fundamentación Teórica

basadas en la plataforma NetBeans pueden ser extendidas fácilmente por otros desarrolladores de software.

## 1.17 Herramienta de diseño gráfico, GIMP

El tratamiento de imágenes se hace necesario cada vez más en un entorno de trabajo donde la elaboración de documentos electrónicos es básica para cualquier tarea diaria. GIMP es una herramienta, de gran potencia y reconocido prestigio en el entorno del software libre. Se trata de un programa que permite retocar fotografías, crear imágenes para la Web y otras muchas utilidades. Se presenta bajo licencia libre, lo que permite al usuario distribuirlo y adaptarlo a sus necesidades. Otra característica interesante es su disponibilidad tanto para plataformas Linux como Windows con lo que se garantiza la posibilidad de una implantación progresiva y adaptada a las necesidades de la administración. La interfaz de GIMP está disponible en varios idiomas, entre ellos: español, inglés, catalán, gallego, euskera, francés, italiano, ruso, sueco, noruego, coreano y neerlandés.[14]

GIMP es también conocido por ser quizás la primera gran aplicación libre para usuarios no profesionales o expertos. Productos originados anteriormente, como GCC, el núcleo Linux, etc., eran principalmente herramientas de programadores para programadores. GIMP es considerado por algunos como una demostración fehaciente de que el proceso de desarrollo de software libre puede crear aplicaciones que los usuarios comunes, no avanzados, pueden usar de manera productiva. De esta forma, Gimp ha abierto el camino a otros proyectos como KDE, GNOME, Mozilla Firefox, OpenOffice.org y otras aplicaciones posteriores.

## 1.18 Conclusiones

Del estudio realizado anteriormente se puede concluir que dada la importancia que se le atribuye hoy en día a la gestión de planes de trabajo en el mundo empresarial, se hace necesaria la aplicación de las nuevas tecnologías de información, con el fin de perfeccionar este proceso logrando una mayor calidad en la gestión.

# Capítulo 1. Fundamentación Teórica

Para ello se ha realizado un profundo estudio teórico, capaz de definir las necesidades existentes las cuales llevarán consigo a la implementación de un sistema automatizado que se convertirá en una herramienta muy útil para el control de la información.

Al finalizar este estudio, se definieron además las metodologías a utilizar, así como los lenguajes, tecnologías y herramientas que se emplearán en el desarrollo de la aplicación.

## 2.1 Introducción

La descripción de las distintas fases de la metodología FDD, la descripción del flujo actual de los procesos involucrados en el campo de acción junto con el objeto de automatización, los principales conceptos asociados al dominio del sistema, la elaboración del modelo de objeto, la especificación de los requerimientos que debe tener el sistema, el estudio de factibilidad de la aplicación son los aspectos que se tomarán en cuenta para la confección del siguiente capítulo.

## 2.2 Desarrollo de un modelo global.

Cuando esta fase comienza, los expertos del dominio ya tienen una idea del contexto y los requerimientos del sistema. Es probable que el documento de especificación de requerimientos ya exista. Sin embargo, FDD no hace énfasis en la recolección y administración de los requerimientos. [5]

El dominio global es dividido en diferentes áreas y se realiza un informe detallado para cada una de dichas áreas por parte de los expertos del dominio. Luego de cada informe, un grupo de desarrolladores realizan un modelo de objetos para el área del dominio. Además, el equipo de desarrollo discute y decide cual es el modelo de objetos más apropiado para cada área del dominio. Simultáneamente, el modelo global del sistema es construido.[5]

### 2.2.1 Flujo actual de los procesos involucrados en el campo de acción.

En el área de transporte de la universidad de Cienfuegos (UCF) se llevan a cabo varias actividades entre las que se encuentran la transportación del personal y el mantenimiento de los vehículos siendo estas las que se desean automatizar.

# Capítulo 2. Descripción del Sistema

La actividad de transportación funciona de la siguiente forma:

En el departamento de transporte de la Universidad de Cienfuegos cada mes se elabora el plan de transportación. En este plan se organizan las actividades que son planificadas por la dirección del centro en el plan de trabajo del mes junto con la cantidad de combustible que se presume se pueda consumir en ellas. Estas actividades pueden ser diarias, semanales o por intervalos de días, además pueden tener diferentes destinos. Asimismo puede ocurrir que el combustible planificado para las mismas varíe en dependencia de la disponibilidad de vehículos del departamento. Por otro lado semanalmente los administradores de las distintas áreas de la universidad realizan solicitudes de viaje en dependencia de las demandas de su área. Dicha acción se realiza mediante una planilla con el nombre de la facultad que lo solicita, destino, hora de partida, cantidad de pasajeros y el propósito del viaje. Después que se realiza esta solicitud pasa a ser aprobada por el decano de la facultad o el máximo directivo del área que lo solicita. En caso de no ser aprobada a nivel de área se desecha y se notifica al administrador, de lo contrario se envía al Jefe de Transporte el cual recopila durante toda la semana esta información. La semana comienza y termina los miércoles a las 12 meridiano y los viajes solicitados son analizados en las reuniones de transporte que se efectúan cada jueves, donde son procesados y analizados para luego darles respuesta a los administradores. Las solicitudes que no son aprobadas se desechan y se le informa al solicitante y las que se aprueban en la reunión pasan a formar parte del plan de transportación junto con otros datos adicionales (poner datos). En este plan de transportación se organizan todas las solicitudes de las áreas. Adicionalmente se anexan a este plan algunas actividades de urgencia, es decir, que no forman parte de ninguna solicitud, por pedido de algún directivo del centro. Cuando un viaje se realiza adicionalmente se anexa a la actividad planificada el estado de realizado, el combustible real consumido y el chofer recoge en un modelo las opiniones de los pasajeros para posteriormente ser revisadas por el Jefe de Transporte.

La actividad de mantenimiento de los vehículos funciona de la siguiente forma:

# Capítulo 2. Descripción del Sistema

El departamento de transporte de la Universidad de Cienfuegos cuenta con múltiples vehículos de diferentes tipos (Ligero, Moto, Camión, Autobús, Tractor). A cada vehículo que pasa a formar parte de este departamento se le confecciona un expediente en el cual se guardan los siguientes datos:

- Chapa
- Color
- Marca
- Modelo
- No de Carrocería
- No Motor.
- Kms recorridos.
- Kms recorridos desde el último mantenimiento.
- Tipo (Ligero, Moto, Camión, Autobús, Tractor)
- Estado.

En la medida en que los vehículos realizan viajes del centro se actualiza su kilometraje recorrido. Los vehículos que alcancen los 8000 Kms recorridos se les realiza un mantenimiento que consiste en el cambio del filtro de aceite y de aceite. En caso de que el almacén tenga algún otro insumo en existencia se le aumenta a dicho mantenimiento.

Para cada mantenimiento o rotura que surja de imprevisto se confecciona una orden de trabajo en la cual se describe el tipo de trabajo a realizar (rotura o mantenimiento) y se enumeran los materiales y piezas a solicitar al almacén. Posteriormente de ser confeccionada esta orden de trabajo se realiza la solicitud de materiales para el almacén de ATM. En ATM al asignar los materiales y piezas al departamento se confeccionan dos copias del vale de salida el cual lleva impreso el número de la orden de trabajo a la cual pertenece. Una de las copias es para archivar en el departamento la otra se archiva en ATM.

Luego de este procedimiento el mecánico cuenta con los materiales y herramientas necesarias y repara o le da mantenimiento al carro e informa al técnico del departamento el nuevo estado técnico de dicho vehículo.

# Capítulo 2. Descripción del Sistema

Las roturas surgen de imprevisto y se reparan en dependencia de las piezas de repuestos con las que cuente el almacén en ese momento.

Las roturas se encuentran organizadas como se muestra en la siguiente figura:

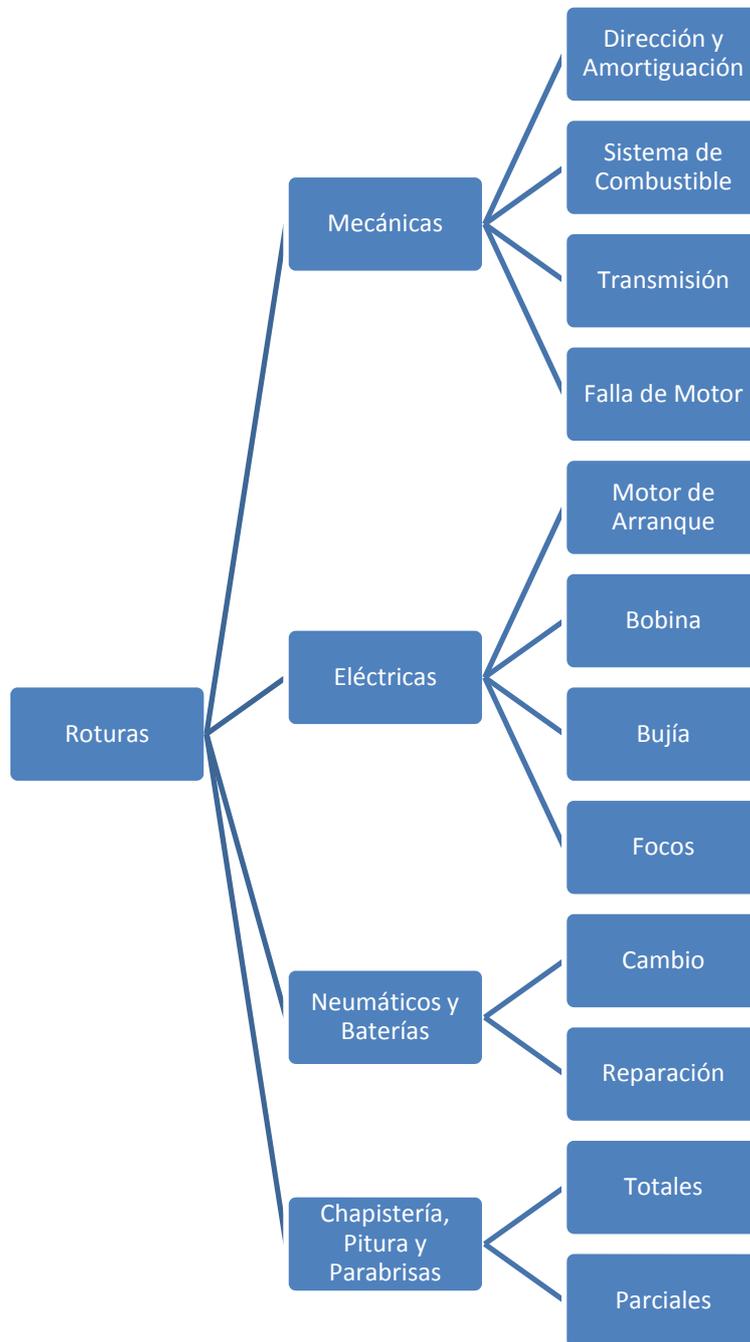


Figura 2.1 Jerarquía de reparaciones.

## 2.2.2 Objeto de automatización.

Se desea automatizar los procesos relacionados con la transportación del personal. Entre los cuales resalta el proceso de elaboración de las solicitudes

# Capítulo 2. Descripción del Sistema

por el administrador, aprobación por la decana, aprobación en el vicerrectorado, inclusión en el plan de trabajo, asignación de vehículo(s) y combustible y realización del viaje. Asimismo la elaboración de planes de transportación mensual y gráficos de cumplimiento del plan de transportación. Además se pretende automatizar el control de los mantenimientos y roturas de los distintos vehículos con los que cuenta la Universidad de Cienfuegos. Indicando al técnico de transporte el listado de los vehículos del centro que necesitan mantenimiento, en adición se almacenará el registro de cada uno de los mantenimientos realizados con sus respectivas órdenes de trabajo en cada caso.

## 2.2.3 Modelo de dominio. Diagrama de objetos.

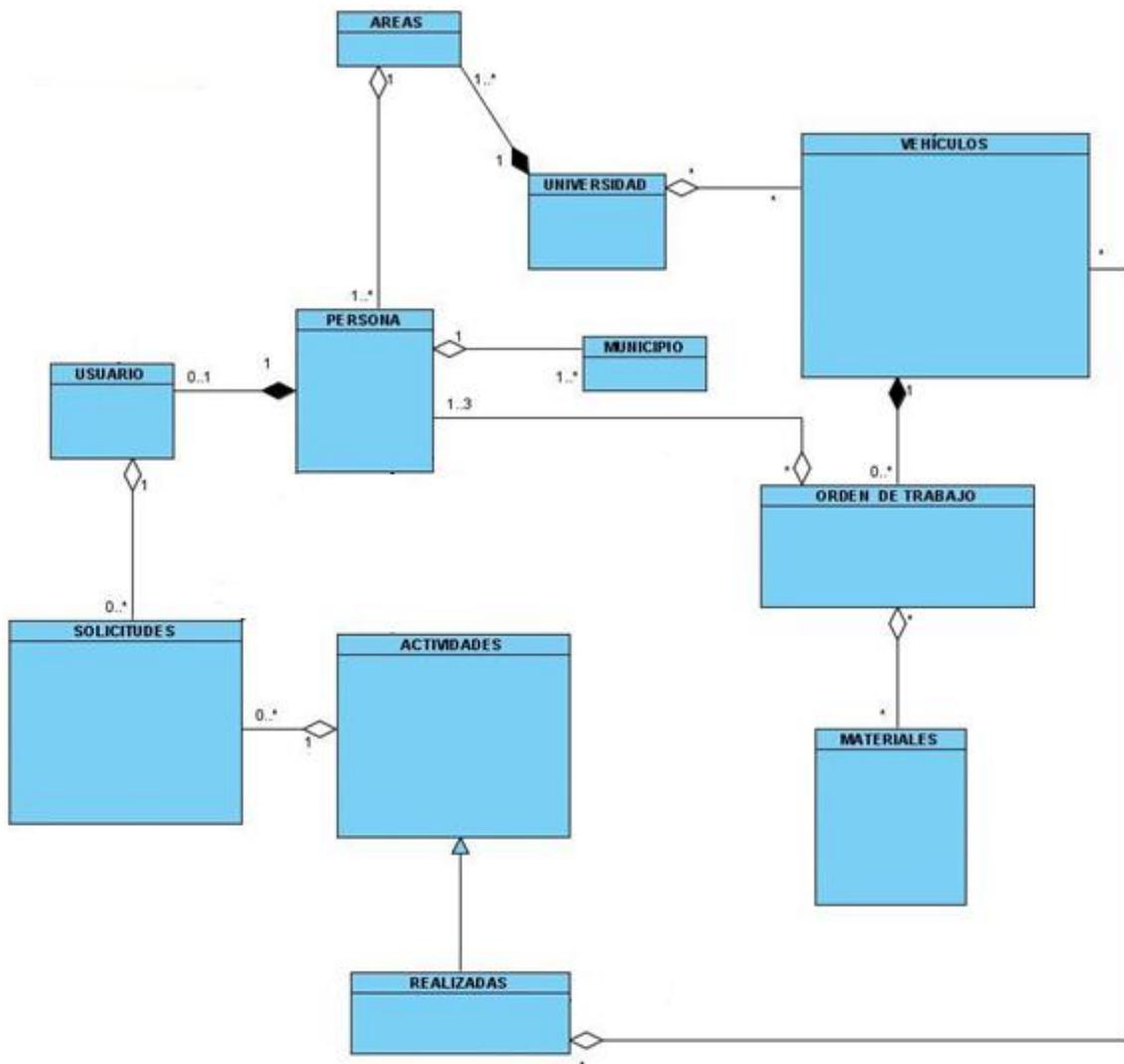


Figura 2.2 Modelo de objetos.

# Capítulo 2. Descripción del Sistema

## 2.2.4 Reglas a considerar.

- Cada mes se elabora un plan de transportación.
- Las actividades que se conciben en reuniones quedan bajo la responsabilidad del Jefe de transporte o del Técnico del departamento para ser incluidas en el plan de transportación.
- Las solicitudes para viajes son realizadas por los administradores de cada área.
- Dichas solicitudes pasan a ser aprobadas a nivel de área por el máximo directivo de la misma.
- El Técnico del departamento es el encargado de recepcionar los vehículos.
- El mantenimiento de un vehículo se realiza cada 8000 kms recorridos.

## 2.3 Construcción de una lista de funcionalidades.

El modelo de objetos y los requerimientos existentes ofrecen una buena base para construir una lista de funcionalidades que resuma la funcionalidad del sistema a ser desarrollado. En dicha lista, el equipo de desarrolladores presenta cada una de las funcionalidades evaluadas por el cliente. Las funcionalidades son presentadas por cada área del dominio y éstas forman un Lista de Funcionalidades. Dicha lista es dividida en subconjuntos en base a la funcionalidad. Estas representan diferentes actividades con un área específica del dominio. La lista de funcionalidades es revisada por los usuarios y sponsors del sistema para su validación y aprobación.[5]

### 2.3.1 Requerimientos funcionales.

Los requerimientos funcionales permiten expresar una especificación más detallada de las responsabilidades del sistema que se propone. Ellos permiten determinar, de una manera clara, lo que debe hacer el mismo. Son capacidades o condiciones que el sistema debe cumplir definiendo el comportamiento interno del software así como los comportamientos del sistema.

### Funcionalidades de Transportación de Personal:

# Capítulo 2. Descripción del Sistema

Nro.	Funcionalidad	Prioridad
1	Insertar solicitud	A
2	Eliminar solicitud	A
3	Buscar solicitud	A
4	Cambiar estado de solicitud por VREA	A
5	Cambiar estado de solicitud	A
6	Mostrar plan de transportación de un mes	A
7	Insertar actividades	A
8	Eliminar actividades	A
9	Buscar actividades por rango de fecha	A
10	Asignar a actividad vehículo y combustible	A
11	Eliminar asignación de vehículo y combustible a una actividad	A
12	Listar viajes dada la evaluación	A
13	Realizar viaje	A
14	Unir viaje	
15	Eliminar realización	A
16	Listar solicitudes aprobadas por rango fecha y por área	A
17	Listar dado un usuario la cantidad de solicitudes realizadas	A
18	Listar viajes realizados en un mes	A
19	Calcular cantidad de viajes realizados en un mes	B
20	Listar solicitudes por intervalo de fecha	A
21	Graficar planificación de viajes contra real	B
22	Calcular cantidad de viajes planificados y realizados	B
23	Graficar viajes por evaluación en un mes	B
24	Imprimir solicitud	B

Tabla 2.1 Primer bloque de funcionalidades.

## Funcionalidades de Mantenimientos y Roturas:

25	Insertar vehículos	A
26	Modificar vehículos	A
27	Eliminar vehículos	A
28	Buscar vehículos	A
29	Insertar trabajador	A
30	Modificar trabajador	A

## Capítulo 2. Descripción del Sistema

31	Eliminar trabajador	A
32	Buscar trabajador	A
33	Insertar materiales	A
34	Modificar materiales	A
35	Eliminar materiales	A
36	Buscar materiales	A
37	Insertar área	A
38	Modificar área	A
39	Eliminar área	A
40	Buscar áreas	A
41	Insertar orden de trabajo	A
42	Modificar orden de trabajo	A
43	Eliminar orden de trabajo	A
44	Buscar orden de trabajo	A
45	Imprimir orden de trabajo	A
46	Anexar materiales a orden de trabajo	A
47	Eliminar anexo de materiales a orden de trabajo	A
48	Anexar trabajadores a orden de trabajo	A
49	Eliminar anexo de trabajadores a orden de trabajo	A
50	Generar solicitud de materiales	A
51	Imprimir solicitud de materiales	A
52	Listar vehículos que necesitan mantenimiento	A
53	Actualizar estado de vehículo	A
54	Actualizar kms recorridos de un vehículo	A
55	Visualizar los mantenimientos por períodos	B
56	Visualizar mantenimientos por tipo de vehículos	B
57	Visualizar mantenimientos por vehículos	B
58	Visualizar las roturas por períodos	B
59	Visualizar las roturas por tipo de vehículos	B
60	Visualizar las roturas por vehículos	B

Tabla 2.2 Segundo bloque de funcionalidades.

### Funcionalidades de administración:

61	Insertar usuario	A
----	------------------	---

# Capítulo 2. Descripción del Sistema

62	Modificar usuario	A
63	Eliminar usuarios	A
64	Listar usuarios	A
65	Autenticarse	A
66	Cambiar contraseña	A

Tabla 2.3 Tercer bloque de funcionalidades.

## 2.3.2 Requerimientos no funcionales.

Los requisitos no funcionales son propiedades o cualidades que el producto debe tener. Debe pensarse en estas propiedades como las características que hacen al producto atractivo, usable, rápido o confiable. En muchos casos los requisitos no funcionales son fundamentales en el éxito del producto. Normalmente están vinculados a requisitos funcionales, es decir una vez que se conozca lo que el sistema debe hacer se puede determinar cómo ha de comportarse, qué cualidades debe tener o cuán rápido o grande debe ser.[6]

### Requerimientos de apariencia o interfaz externa:

- El sistema debe tener una interfaz sencilla, legible y simple de usar.
- La interfaz debe ser diseñada respetando parámetros de diseño.

### Requerimientos de usabilidad:

- La interfaz de usuario deberá ser tan familiar como sea posible a otras aplicaciones similares que los usuarios hayan utilizado.

### Requerimientos de rendimiento:

- El sistema propuesto debe ser rápido en el procesamiento de la información así como a la hora de dar respuesta solicitudes de los usuarios. No debe tardarse más de 5 segundos para realizar acciones complejas, que no dependan del acceso a las colecciones de datos.

### Requerimiento de Portabilidad:

- La aplicación propuesta será desarrollada en la plataforma Windows. No obstante los clientes podrán instalar la aplicación en el Sistema Operativo Linux o Windows desde versión 2000.

# Capítulo 2. Descripción del Sistema

## Requerimientos de seguridad

Se debe garantizar un control estricto sobre la seguridad de la información teniendo en cuenta el establecimiento de niveles de acceso. Se define una política de usuarios con privilegios de acuerdo a su rol, lo que asegura que la información pueda ser consultada de acuerdo a su nivel de acceso.

Accederán de manera rápida y operativa los usuarios sin que los requerimientos de seguridad se conviertan en un retraso para usar el sistema.

Se utilizará un mecanismo de encriptación para las contraseñas que por cuestiones de seguridad no deben viajar al servidor en texto plano. Se guardará encriptada en la base de datos usando para ello la función hash MD5.

La función hash es un tipo de algoritmo que calcula un mensaje implícito o valor hash para cualquier mensaje que se le indique. El valor generado por el algoritmo no es importante, lo importante es que el resultado sea fijo, es decir, que sea el mismo cada vez que se utiliza una entrada dada, que sea pequeño y que el algoritmo sea rápido. La función MD5 calcula y devuelve el hash MD5 que consiste en una cadena de 32 números en formato hexadecimal.[10]

## Requerimiento de Hardware:

- Las máquinas que utilizarán el software tienen que estar conectadas a la red.
- Los requerimientos mínimos para una estación de trabajo deben ser de 128 MB de RAM, 800 MHz, 2GB de disco duro e impresora Epson LX300+ o superior.
- Los requerimientos mínimos para el servidor del software deben ser de 1 GB de RAM, 1.4 GHz, 10 GB de disco duro.

## Requerimientos de software:

- Los requerimientos mínimos para una estación de trabajo deben ser:
  - Navegador Mozilla Firefox 7.0 o superior.
- Se requiere además, una máquina servidor que cuente con:

# Capítulo 2. Descripción del Sistema

- La versión 2.2.6 o superior del servidor Web Apache.
- Lenguaje de programación PHP 5.2.5
- Servidor de base de datos MySQL 5.0.45 o superior.
- Navegador Mozilla Firefox 7.0 o superior.

## 2.4 Estudio de la factibilidad.

Para la estimación del tamaño de un sistema a partir de sus requerimientos, una de las técnicas más difundidas es el Análisis de Puntos de función. Esta técnica permite cuantificar el tamaño de un sistema en unidades independientes del lenguaje de programación, las metodologías, plataformas y/o tecnologías utilizadas. [15]

Por otro lado, el SEI (del inglés, Software Engineering Institute) propone desde hace algunos años un método para la estimación del esfuerzo llamado COCOMO II. Éste método está basado en ecuaciones matemáticas que permiten calcular el esfuerzo a partir de ciertas métricas de tamaño estimado, como el Análisis de Puntos de Función y las líneas de código fuente (en inglés SLOC, Source Line Of Code).[15]

En este epígrafe se propone la utilización combinada de dos métodos (Puntos de Función y Cocomo) los cuales tienden a proporcionar una estimación más precisa de la siguiente forma:

Primero la aplicación del método de Puntos de Función para determinar las sentencias de código del proyecto software.

En segundo lugar la aplicación del método COCOMO II partiendo de la información producida por el anterior para llegar a una estimación precisa de las horas hombre a aplicar y fundamentalmente a la estimación de la duración total del proyecto.

Después clasificar cada una de las funcionalidades previstas se arribó a los siguientes resultados:

# Capítulo 2. Descripción del Sistema

Nombre de la entrada externa	Cantidad de ficheros Referenciados	Cantidad de Elementos de datos	Complejidad (Simple, Medio y Complejo)
1	3	7	Medio
2	1	1	Simple
4	1	2	Simple
5	1	2	Simple
7	3	8	Medio
8	1	1	Simple
10	2	3	Simple
11	1	1	Simple
12	1	3	Simple
15	3	10	Medio
16	1	1	Simple
24	2	10	Medio
25	1	10	Medio
26	1	1	Simple
28	3	6	Medio
29	1	6	Medio
30	1	1	Simple
32	1	7	Medio
33	1	7	Medio
34	1	1	Simple
36	2	1	Simple
37	1	1	Simple
38	1	1	Simple
40	4	4	Medio
41	1	4	Medio
42	1	1	Simple
45	3	2	Simple
46	1	2	Simple
47	3	2	Simple
48	1	2	Simple
52	1	2	Simple
53	1	2	Simple
60	2	3	Simple
61	1	3	Simple
62	1	1	Simple
65	1	2	Simple

Tabla 2.4 Entradas externas.

# Capítulo 2. Descripción del Sistema

Nombre de la salida externa	Cantidad de ficheros referenciados	Cantidad de Elementos de datos	Complejidad (Simple, Medio y Complejo)
21	2	6	Medio
22	1	6	Medio

Tabla 2.5 Salidas externas.

Nombre de la consulta externa	Cantidad de ficheros Referenciados	Cantidad de Elementos de datos	Complejidad (Simple, Medio y Complejo)
3	1	8	Medio
6	1	9	Medio
9	1	9	Medio
13	2	11	Medio
17	1	10	Medio
18	1	7	Medio
19	2	10	Medio
20	2	11	Medio
24	1	11	Medio
27	1	12	Medio
31	1	11	Medio
35	1	8	Medio
39	1	4	Medio
43	5	10	Medio
39	1	7	Medio
49	0	8	Medio
50	1	8	Medio
51	1	11	Medio
54	1	8	Medio
55	1	8	Medio
56	1	8	Simple
57	1	8	Simple
58	1	8	Medio
59	1	8	Medio
63	1	6	Medio
64	1	2	Medio
66	1	3	Medio

Tabla 2.6 Consultas externas.

Nombre del fichero lógico interno	Cantidad de ficheros referenciados	Cantidad de Elementos de datos	Complejidad (Simple, Medio y Complejo)
Universidad	1	3	Simple
Áreas	2	3	Simple
Vehículos	2	11	Medio
Orden_de_trabajo	3	8	Medio
Materiales	1	7	Simple
Personas	4	9	Medio
Municipios	1	2	Simple
Usuarios	3	6	Medio

# Capítulo 2. Descripción del Sistema

Solicitudes	4	12	Medio
Actividades	1	11	Medio
Realizadas	1	5	Simple

Tabla 2.7 Archivos lógicos internos.

Elementos	Baja	X Peso	Media	X Peso	Alta	X Peso	Subtotal de puntos de función
Entradas externas	25	3	11	4	2	6	131
Salidas externas	0	4	2	5	0	7	10
Consultas externas	2	3	25	3	0	6	81
Ficheros lógicos internos	4	7	6	10	0	15	88
						<b>Total</b>	310

Tabla 2.8 Puntos de función.

$$UFP = 310$$

$$PM_{nominal} = A \times (Size)^B$$

**A:** se toma el valor por defecto del modelo, ajustado en 2.94.

**Size:** se calcula como el producto de los puntos de función sin ajustar por un factor de conversión que depende del lenguaje a utilizar en el desarrollo del sistema. Se utiliza Java (factor de conversión medio para lenguajes orientados a objetos = 30 SLOC/UFP). Entonces: [15]

$$Size = UFP \times FC$$

$$Size = 310 \times 30 = 9300 \text{ SLOC} = 9.3 \text{ KLOC}$$

**B:** se calcula ponderando las variables escalares mediante la ecuación siguiente:

$$B = 0.91 + 0.01 \times \sum (W_i)$$

donde las **W<sub>i</sub>** muestran en la siguiente tabla:

# Capítulo 2. Descripción del Sistema

Variable	Ponderación	Valor
PREC	Alto	2.40
FLEX	Nominal	3.04
RESL	Alto	2.83
TEAM	Alto	3.29
PMAT	Nominal	4.68

Tabla 2.9 Variables escalares.

## Factores de escala

$$SF = \sum (W_i) = PREC + FLEX + RESL + TEAM + PMAT$$

$$SF = \sum (W_i) = 2.4 + 3.04 + 2.83 + 3.29 + 4.68 = 16.24$$

$$B = 0.91 + 0.01 \times 16.24$$

$$B = 1.0724 \approx 1.07$$

Luego el esfuerzo nominal resulta:

$$PM_{\text{nominal}} = A \times (\text{Size})^B$$

$$PM_{\text{nominal}} = 2.94 \times (9.3 \text{ KLOC})^{1.07} \approx 31.96 \text{ Meses} - \text{Hombre}$$

Para completar la estimación hay que ajustar el esfuerzo nominal de acuerdo a las características del proyecto. El ajuste se efectúa aplicando la ecuación siguiente [38]:

$$PM_{\text{ajustado}} = PM_{\text{nominal}} \times \prod (ME_i)$$

Donde los  $\prod (ME_i)$  (multiplicadores de esfuerzo) varían en función del modelo de estimación seleccionado (Diseño Preliminar o Post arquitectura). En este caso se aplica el modelo de Diseño preliminar. Entonces, se cuantifican los multiplicadores de esfuerzo para este modelo de la siguiente forma [38]:

# Capítulo 2. Descripción del Sistema

Multiplicador	Ponderación	Valor
PERS	Alto	0.83
RCPX	Nominal	1
RUSE	Nominal	1
PDIF	Bajo	0.87
PREX	Alto	0.87
SCED	Nominal	1
FCIL	Muy Alto	0.73

Tabla 2.10 Multiplicadores de esfuerzo.

Multiplicador de esfuerzos

$$E_M = \prod (ME_i) = PERS \times RCPX \times RUSE \times PDIF \times PREX \times SCED \times FCIL$$

$$E_M = \prod (ME_7) = 0.83 \times 1 \times 1 \times 0.87 \times 0.87 \times 1 \times 0.73$$

$$E_M = 0.45860571 \approx 0.46$$

Con estos valores, el ajuste del esfuerzo resulta:

$$PM_{ajustado} = 31.96 \times 0.46 = 14.7 \text{ Meses} - \text{Hombre} \approx 15 \text{ Meses} - \text{Hombre}$$

Este resultado se interpreta como el tiempo requerido para que una persona desarrolle los cuatro módulos de la aplicación.

**Para el costo:**

Se asume como salario promedio mensual \$450.00.

CH: Cantidad de hombres = 3.

Tiempo: Tiempo total de desarrollo del software.

$$\text{Tiempo} = PM_{ajustado} \div CH = 15 \div 3 = 5 \text{ Meses}$$

$$\text{Costo} = \text{SalarioPromedio} \times CH \times \text{Tiempo} = 450 \times 3 \times 5 \text{ Meses} = \$6750$$

Cálculo	Valor
Esfuerzo nominal	31.96 Meses – Hombre
Esfuerzo ajustado	15 Meses – Hombre
Cantidad de hombres	3
Salario promedio	\$450.00
Tiempo	5 Meses
Costo	\$6750

# Capítulo 2. Descripción del Sistema

Tabla 2.11 Resumen de resultados.

Sobre los resultados obtenidos se interpreta que con tres hombres trabajando en la aplicación se desarrolla en 5 meses y su costo total se estima que sea \$6750.00.

## **2.4.1 Beneficios tangibles e intangibles.**

Entre los beneficios tangibles obtenidos con la realización del software se pueden mencionar un ahorro sustancial de recursos humanos y materiales de oficina, lo que permite una disminución del nivel de errores y los retrasos en el logro de los resultados finales. Todo lo anteriormente mencionado puede resumirse en la posibilidad de gestionar rápida y eficientemente la información asociada a los procesos ya mencionados, siendo la misma segura y confiable, brindando de manera significativa ayuda a los directivos del centro en la toma de decisiones. La aplicación y el ahorro de tiempo constituyen los beneficios intangibles que además se traducen en un ahorro económico considerable permitiendo que los empleados dediquen menos tiempo en este proceso y que un trabajo que antes requería de varias horas sea terminado en pocos minutos.

## **2.4.2 Análisis de costos y beneficios.**

La incorporación del sistema propuesto traerá grandes beneficios como se evidenció en el epígrafe anterior. Permitirá entre otros reducir la pérdida de información por deterioro de documentación, mayor rapidez y confiabilidad del proceso, así como la obtención de reportes y gráficos de fácil entendimiento para los especialistas y para asistir a los directivos en la toma de decisiones. El desarrollo no supone grandes gastos de recursos y las tecnologías empleadas no requieren pagos de licencias.

Después del análisis del costo del proyecto y los numerosos beneficios que traerá consigo su despliegue se ha concluido que es factible la realización de este proyecto.

## **2.5 Planeación por funcionalidad.**

En esta etapa se incluye la creación de un plan de alto nivel, en el cual la lista de funcionalidades es ordenada en base a la prioridad y a la dependencia entre

# Capítulo 2. Descripción del Sistema

cada funcionalidad. Además, las clases identificadas en la primera etapa son asignadas a cada programador.[5]

<b>Nro.</b>	<b>Funcionalidad</b>	<b>Prioridad</b>	<b>Fecha Inicial</b>	<b>Fecha Final</b>	<b>Días</b>
1	Insertar solicitud	A	3/10/2011	5/10/2011	2
2	Eliminar solicitud	A	5/10/2011	7/10/2011	2
3	Buscar solicitud	A	10/10/2011	12/10/2011	2
4	Cambiar estado de solicitud por VREA	A	12/10/2011	14/10/2011	2
5	Cambiar estado de solicitud	A	17/10/2011	19/10/2011	2
6	Mostrar plan de transportación de un mes	A	19/10/2011	21/10/2011	2
7	Insertar actividades	A	24/10/2011	26/10/2011	2
8	Eliminar actividades	A	26/10/2011	28/10/2011	2
9	Buscar actividades por rango de fecha	A	31/10/2011	2/11/2011	2
10	Asignar a actividad vehículo y combustible	A	2/11/2011	4/11/2011	2
11	Eliminar asignación de vehículo y combustible a una actividad	A	7/11/2011	9/11/2011	2
12	Listar viajes dada la evaluación	A	9/11/2011	11/11/2011	2
13	Realizar viaje	A	14/11/2011	16/11/2011	2
14	Unir viaje		16/11/2011	18/11/2011	2
15	Eliminar realización	A	21/11/2011	23/11/2011	2
16	Listar solicitudes aprobadas por rango fecha y por área	A	23/11/2011	25/11/2011	2
17	Listar dado un usuario la cantidad de solicitudes realizadas	A	28/11/2011	30/11/2011	2
18	Listar viajes realizados en un mes	A	30/11/2011	2/12/2011	2
19	Calcular cantidad de viajes	B	5/12/2011	7/12/2011	2

# Capítulo 2. Descripción del Sistema

	realizados en un mes				
20	Listar solicitudes por intervalo de fecha	A	7/12/2011	9/12/2011	2
21	Graficar planificación de viajes contra real	B	12/12/2011	15/12/2011	3
22	Calcular cantidad de viajes planificados y realizados	B	15/12/2011	19/12/2011	2
23	Graficar viajes por evaluación en un mes	B	19/12/2011	22/12/2011	3
24	Imprimir solicitud	B	22/12/2011	27/12/2011	3

## 2.6 Diseño por funcionalidad.

Un conjunto de funcionalidades son seleccionadas de la lista de funcionalidades.

El diseño y construcción de la funcionalidad es un proceso iterativo durante el cual las funcionalidades seleccionadas son producidas. Una iteración puede llevar desde unos pocos días a un máximo de dos semanas. Este proceso iterativo incluye tareas como inspección del diseño, codificación, prueba unitaria, integración e inspección del código. Luego que la iteración llega a su fin se realiza una construcción de la funcionalidad en el cual esta es integrada.[5]

### 2.6.1 Principios de diseño.

Los elementos gráficos o textuales que componen la interfaz son claros y de fácil identificación. Se hizo uso de términos y conceptos que se tomaron de la experiencia de las personas que más utilizarán el sistema. Adicionalmente se tuvieron presente los siguientes aspectos:

- Las operaciones comparables se activan de la misma forma.
- El sistema no provoca sorpresa a los usuarios.
- Se incluyeron mecanismos para permitir a los usuarios recuperarse de los errores mediante la confirmación de acciones destructivas.

# Capítulo 2. Descripción del Sistema

- La interfaz provee retroalimentación significativa y características de ayuda sensible al contexto.
- La interfaz provee características de interacción apropiada para los diferentes tipos de usuarios.
- La manipulación directa: Interacción directa con los objetos de la pantalla, rápida e intuitiva y fácil de aprender.
- Se requiere teclear poco.
- Pocas opciones en cada menú.
- Introducción de datos sencilla en los campos de un formulario.

## 2.6.2 Modelo conceptual de la base de datos.

En general los modelos conceptuales por su nivel de abstracción y riqueza semántica constituyen una interfaz útil entre el informático y los usuarios finales en las primeras etapas del proceso de diseño de bases de datos, ya que reflejan en mayor medida la semántica, el significado de los datos y sus interrelaciones.[16]

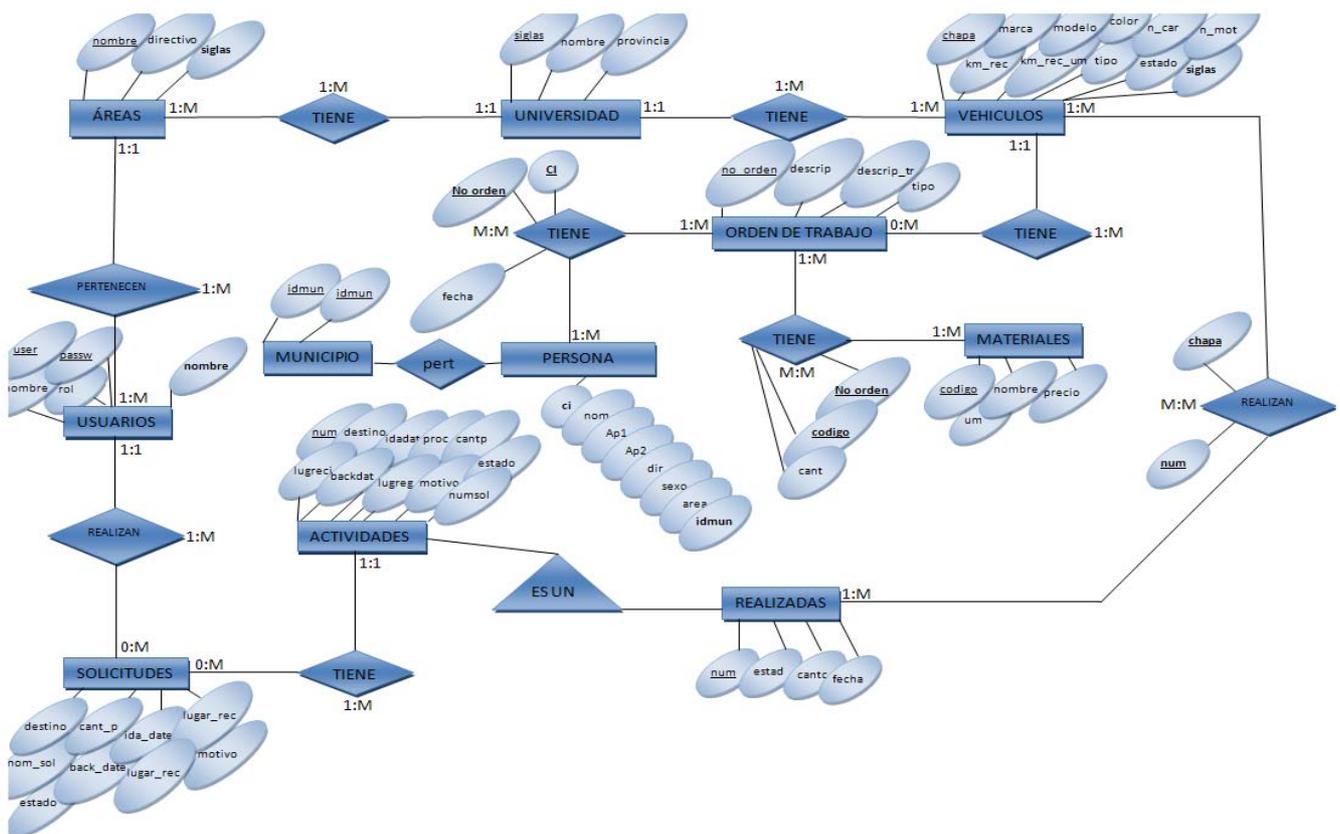


Figura 2.3 Modelo conceptual de la base de datos.

# Capítulo 2. Descripción del Sistema

## 2.6.3 Modelo lógico de la base de datos.

El diagrama del modelo lógico de datos o diagrama de clases persistentes, muestra las clases capaces de mantener su valor en el espacio y en el tiempo.[17]

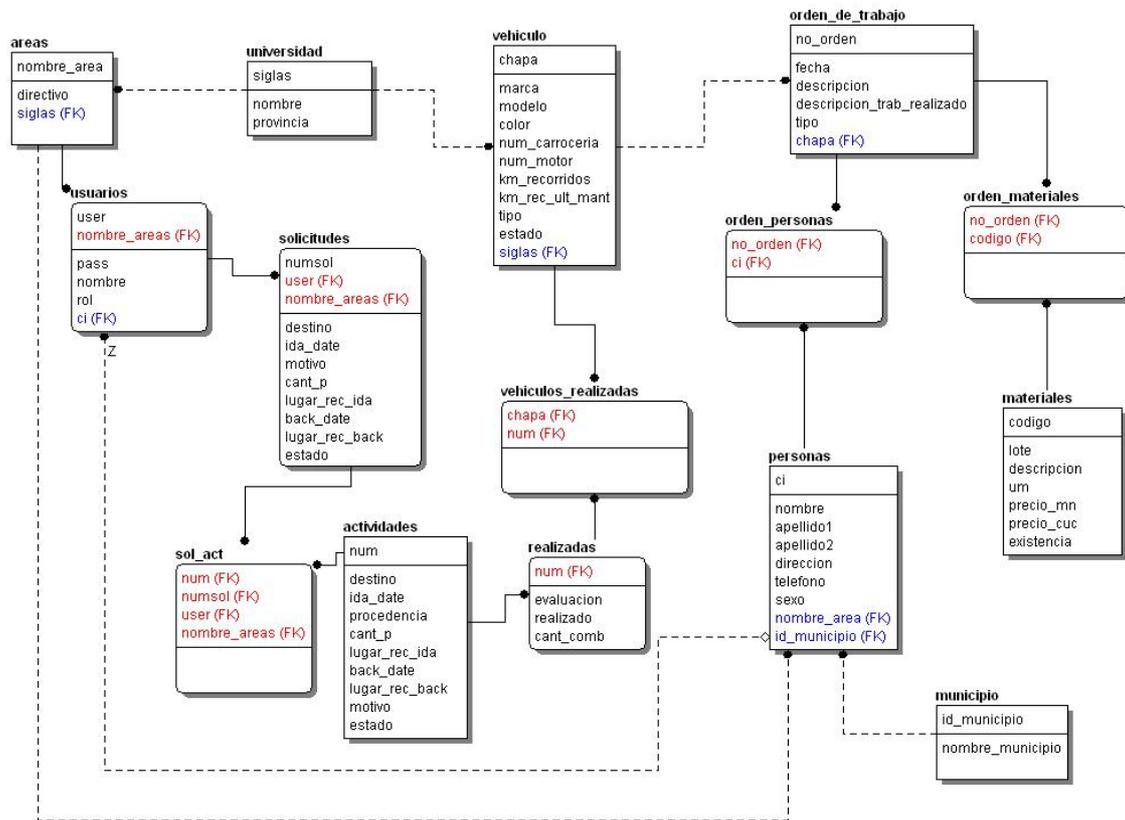


Figura 2.4 Modelo lógico de la base de datos.

## 2.6.4 Modelo físico de la base de datos.

El modelo físico de datos incluye todos los aspectos de diseño de un modelo de base de datos, que se pueden modificar sin cambiar los componentes de la aplicación.[17]

# Capítulo 2. Descripción del Sistema

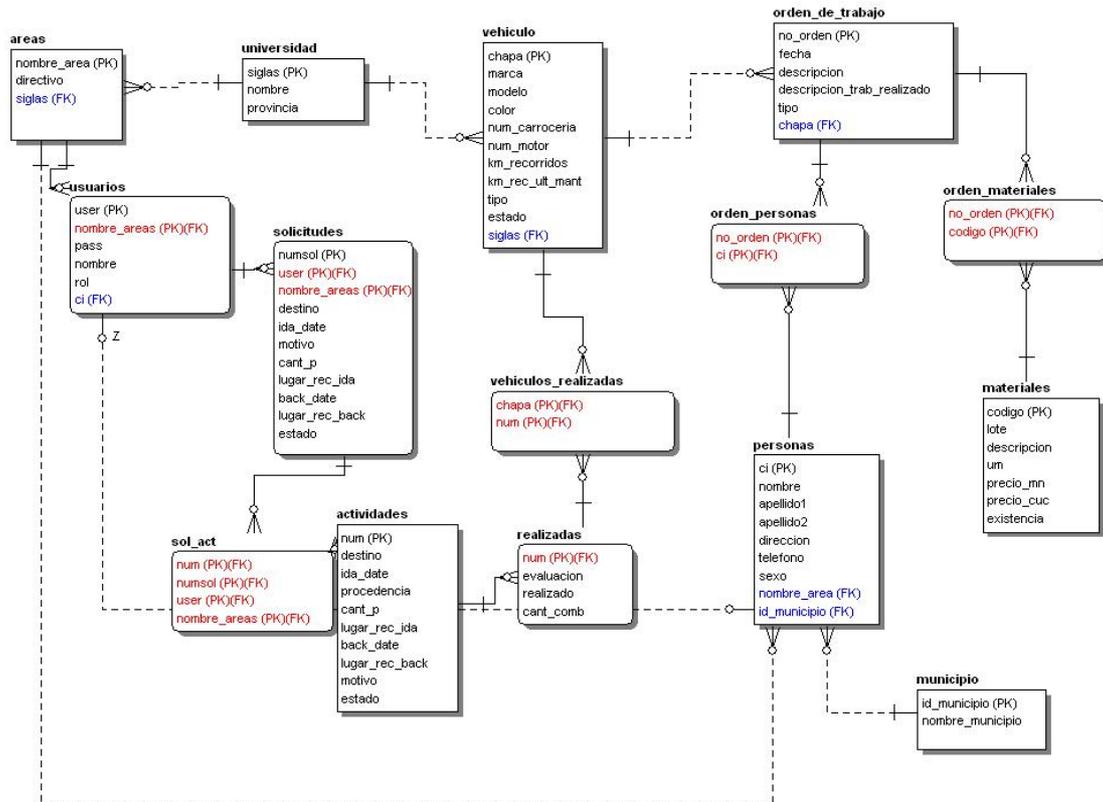


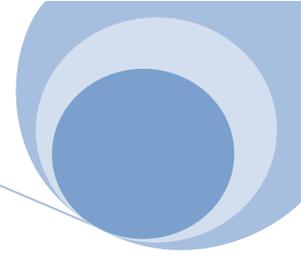
Figura 2.5 Modelo físico de la base de datos.

## 2.7 Conclusiones.

En este capítulo se crearon los artefactos correspondientes a las fases que propone la metodología FDD utilizando notación UML. Se analizó y describió el flujo de los procesos involucrados en la gestión de la información que se genera en el departamento de transporte de la UCF. Se especificaron los requerimientos que debe tener el sistema, se desarrolló la planeación y como parte del diseño por funcionalidades, se elaboró el modelo conceptual, lógico y físico de la base de datos.

Asimismo se realizó un estudio de factibilidad del sistema el cual arrojó un costo estimado de \$6750.00 con un tiempo de desarrollo por tres persona de aproximadamente cinco meses. Analizando el costo y los beneficios de la implantación del sistema y las mejoras que introduce en los procesos del departamento de transporte de la UCF se concluyó que el desarrollo del proyecto es factible.

# Capítulo 3. Construcción del Sistema.



## Capítulo 3

### 3.1 Introducción

En este capítulo se realiza una descripción del diseño del sistema utilizando los diagramas de clases. Se aborda todo lo relacionado con la seguridad del sistema y el tratamiento de errores. Adicionalmente se realiza la validación de la solución propuesta.

### 3.2 Descripción de la arquitectura propuesta

CodeIgniter está basado en el patrón de desarrollo Modelo-Vista-Controlador. MVC es una aproximación al software que separa la lógica de la aplicación de la presentación.

El Modelo representa la estructura de datos. Típicamente sus clases de modelo contendrán funciones que lo ayudarán a recuperar, insertar y actualizar información en su base de datos. Se implementó una clase del modelo para cada tabla de la base de datos.[11]

La Vista es la información que es presentada al usuario. La Vista normalmente será una página web, pero en CodeIgniter, una vista también puede ser un fragmento de una página como un encabezado o un pie de página. Las vistas de la aplicación poseen elementos comunes que fueron definidos en la plantilla además de las vistas diseñadas para cada una de las funcionalidades.[11]

El Controlador sirve como un intermediario entre el Modelo, la Vista y cualquier otro recurso necesario para procesar la petición HTTP y generar una página web. Se definieron doce clases controladoras, de manera tal que cada una gestiona la información relacionada con una clase entidad, además una para la

# Capítulo 3. Construcción del Sistema.

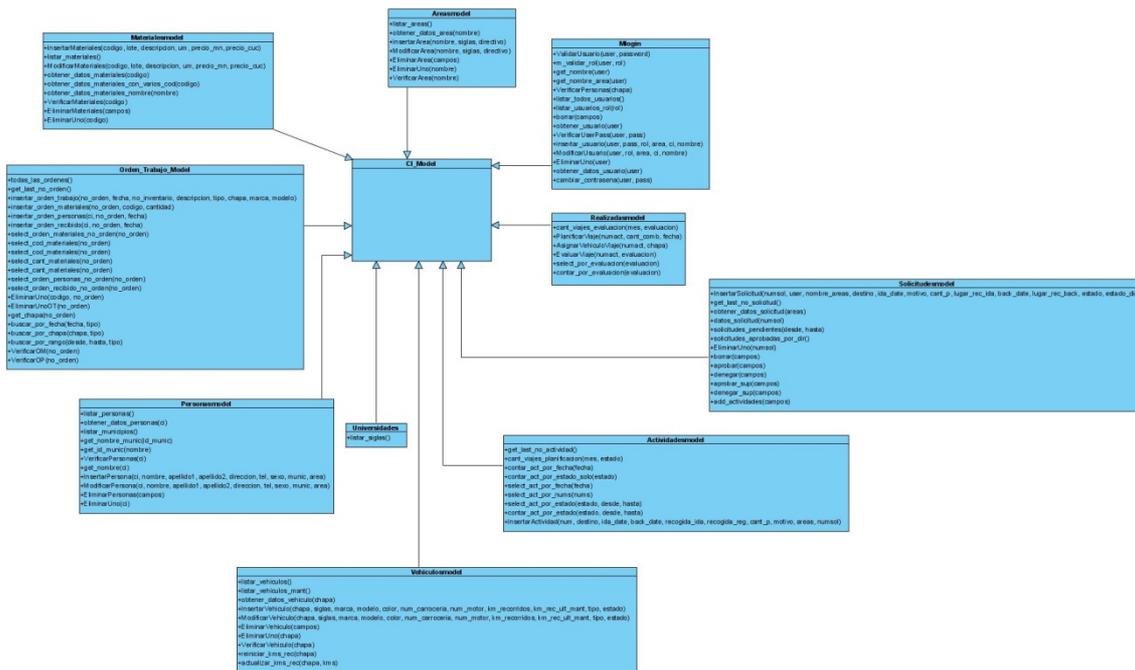
administración y la principal que es la que se configura para ser cargada por la vista principal del sistema.[11]

Teniendo en cuenta que la aplicación se implementó utilizando como base CodeIgniter desde el punto de vista técnico y arquitectónico posee las siguientes características:

- **Bajo Acomplamiento:** Acomplamiento es el grado que los componentes de un sistema dependen entre ellos. Mientras menos componentes dependan de otro, más reusable y flexible el sistema se vuelva. [11]
- **Singularidad del Componente:** Singularidad es el grado que más componentes tienen un propósito en el que enfocarse más estrecho. Cada clase y sus funciones son altamente autónomas para permitir máxima utilidad.[11]

De manera general el sistema es poco acoplado con gran singularidad de componente. Posee simplicidad, flexibilidad y buen rendimiento.

## 3.3 Diagrama de clases



# Capítulo 3. Construcción del Sistema.

Figura 3.1 Diagrama de clases del modelo.

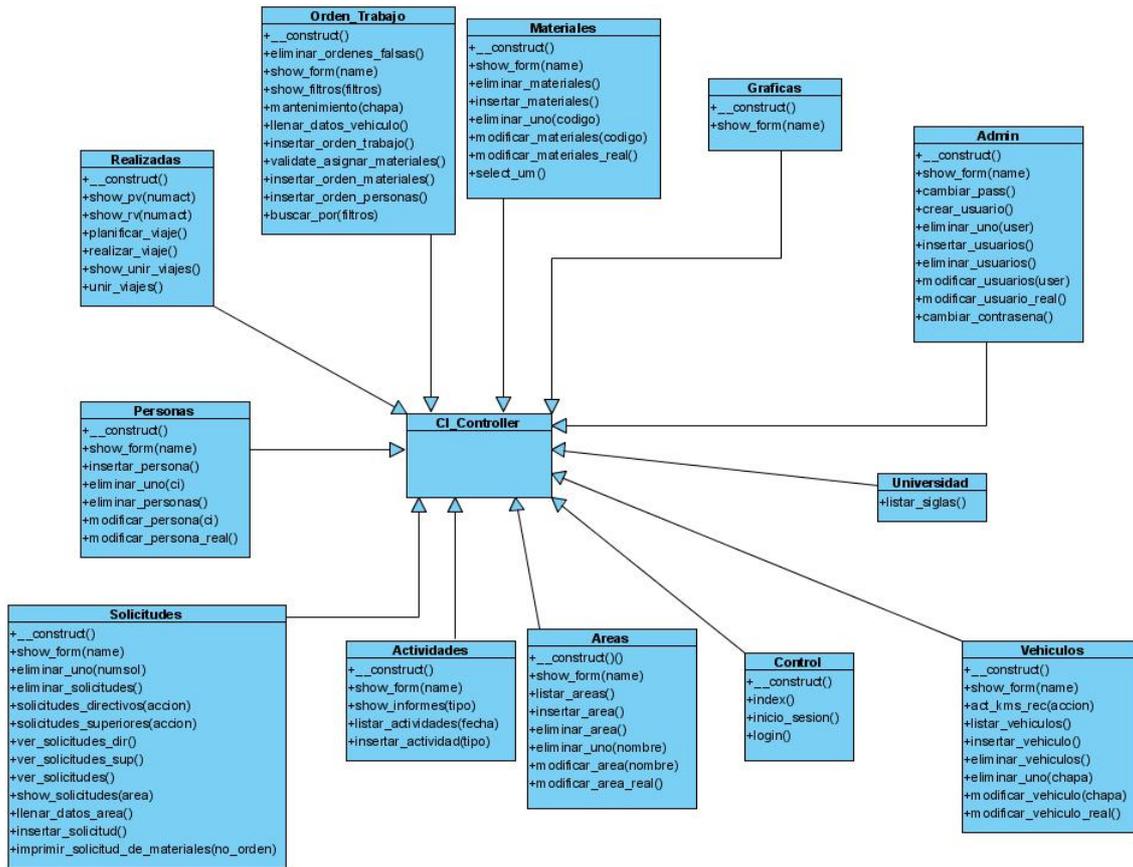
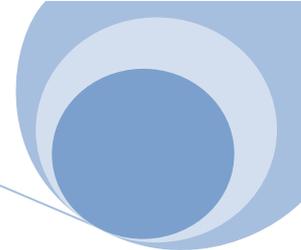


Figura 3.2 Diagrama de clases del controlador.

## 3.4 Estándar de implementación

Para el adecuado mantenimiento del código del sistema es necesario establecer un estándar de codificación a emplear. Las variables, nombres de funciones, de clases y objetos fueron declarados en las páginas son cortos, claros, y describen claramente su propósito. Los objetos se nombran según el valor de su contenido. Los inicios ({} y cierre (}) de ámbito se encuentran alineados debajo de la declaración a la que pertenecen. Los signos lógicos y de operación se separan por un espacio antes y después de los mismos. Se emplean comentarios para añadir información sobre determinadas líneas de código en las páginas y así ayudar a entender el objetivo del mismo. Se emplean además nombres similares entre clases de manera tal que se

# Capítulo 3. Construcción del Sistema.



establezca claramente la relación entre la clase del modelo, la clase controladora que maneja sus datos y la tabla de la base de datos de donde obtienen la información necesaria.

## 3.5 Seguridad del sistema

- El mecanismo de seguridad y protección de la aplicación se basa en el empleo del nombre de usuario y la contraseña para acceder a las funcionalidades a las que tienen acceso.
- Cada usuario tiene definido a que módulo e información puede acceder, teniendo en cuenta la política de seguridad del departamento y de la Universidad.
- Los usuarios una vez registrados en su módulo no podrán acceder haciendo uso de la url a otra página a la cual no tiene acceso.
- Los usuarios una vez que se desconectan de su módulo no podrán regresar y realizar acciones dentro del mismo sin antes autenticarse.
- Se utilizaron mecanismos de encriptación de los datos usando para ello la función hash MD5.

## 3.6 Tratamiento de errores

A menudo las personas que usan sistemas informáticos cometen errores que pueden afectar el adecuado funcionamiento de los mismos. Por lo que es necesario siempre realizar un adecuado tratamiento para estas situaciones excepcionales que pueden suceder. Como por ejemplo la entrada de datos incorrectos o en un formato no soportado por el sistema.

El sistema está diseñado e implementado de forma tal, que las posibilidades de introducir información errónea sean mínimas. Aunque en algunos casos es necesario teclear datos y seleccionar elementos, se mantiene un nivel elevado de validación. Se muestran comentarios sobre el formato de la información a introducir en las áreas de textos de las pantallas. Sobre los botones se colocan etiquetas informativas para brindarle información al usuario sobre cada

# Capítulo 3. Construcción del Sistema.

funcionalidad. Adicionalmente a la validación los mensajes de error que muestra el sistema se encuentran escritos en un lenguaje de fácil comprensión para los usuarios.

## 3.7 Diagrama de despliegue

El diagrama de despliegue es un modelo de objetos que describe la distribución física del sistema en términos de cómo se distribuye la funcionalidad entre los nodos de cómputo. Es una colección de nodos y arcos; donde cada nodo representa un recurso de cómputo, normalmente un procesador o un dispositivo de hardware similar.[18]

El siguiente diagrama muestra la configuración hardware del sistema y los nodos físicos que lo componen. El sistema estará estructurado según la arquitectura cliente - servidor. En el lado del servidor estarán en funcionamiento, el servidor de bases de datos MySQL y el servidor Web Apache. Esta se comunicará con el cliente de la Intranet a través del protocolo HTTP. El cliente podrá visualizar la aplicación con el Mozilla Firefox 7.0 o superior. Además se cuenta con una impresora conectada a la máquina cliente que le brindará la posibilidad al usuario de imprimir la información que desee.

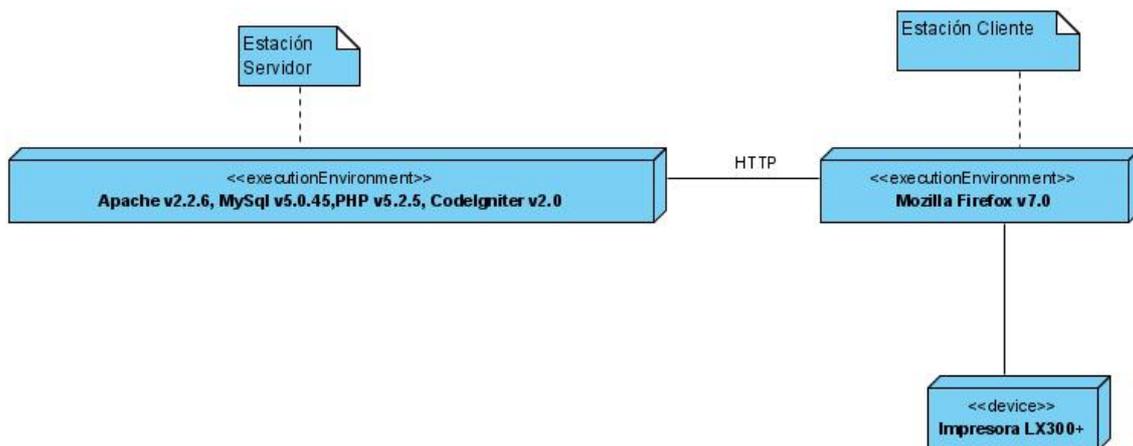
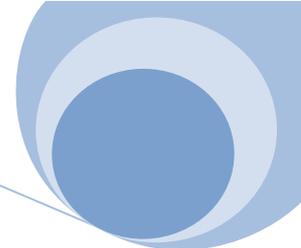


Figura 3.3 Diagrama de despliegue del sistema.

## 3.8 Validación de la solución propuesta

En la fase de validación se realizaron diferentes mediciones de las variables de tiempo promedio de duración de los procesos que abarcan desde la confección

# Capítulo 3. Construcción del Sistema.



de la solicitud hasta que se informa si fue aprobada a nivel de universidad para las distintas áreas de la universidad (variable 1) y del proceso de generar un informe de la relación de viajes planificados contra viajes realizados para todos los meses del año (variable 2) antes y después de la configuración del sistema en la red de la universidad.

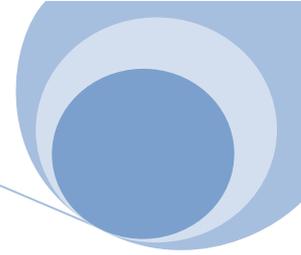
## 3.8.1 Resultados estadísticos de las mediciones realizadas

Se efectuaron mediciones del tiempo que demoraba el proceso de la variable 1 antes y después de la configuración del sistema. Se obtuvieron 15 muestras de una población total de 100. Los datos recopilados se procesaron en el programa SPSS.

Tiempo (minutos) de realización del proceso de la variable 1 y 2, antes y después de la utilización del sistema para varias áreas de la universidad:

Tiempos antes y después de la realización de los procesos				
Muestra	Procesos			
	Variable 1		Variable 2	
	Antes	Después	Antes	Después
	90	20	66	1
95	24	60	0.2	
92	23	68	0.5	
94	20	70	0.9	
89	24	65	0.7	
93	25	67	1	
97	21	62	0.4	
96	20	68	0.2	
91	26	62	0.3	
90	24	61	0.7	
98	19	70	0.9	
100	23	65	1	
91	23	58	0.4	
94	24	48	0.8	
97	23	60	1	

# Capítulo 3. Construcción del Sistema.



Media	93.8	22.6	63.3	4.2
-------	------	------	------	-----

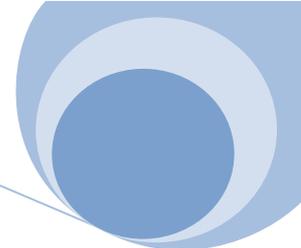
Tabla 3.1 Tiempos antes y después de la realización de los procesos.

Con los datos antes tabulados se puede observar a simple vista la diferencia significativa que existe entre las medias correspondientes a los valores de antes y después para cada una de las variables pero se hace necesario comprobarlo estadísticamente para poder fundamentar con más precisión que verdaderamente existe una diferencia significativa entre estos.

A continuación se muestra el análisis realizado utilizando la prueba de Kolmogorov-Smirnov para comprobar que nuestros valores siguen una distribución de contraste normal.

Dicha prueba realiza un contraste entre la hipótesis nula y la hipótesis alternativa planteando que la variable sigue una distribución normal y que la variable no sigue una distribución normal correspondientemente. Se tomó un nivel de significación de 5%, es decir  $\alpha = 0.05$ , si este es menor que la significación asintótica, entonces se rechaza la hipótesis nula, de lo contrario se acepta.

# Capítulo 3. Construcción del Sistema.



La prueba de Kolmogorov-Smirnov para la Variable 1 utilizando PASW Statistic 18 arrojó el siguiente reporte:

		Antes	Después
N		15	15
Parámetros Normales <sup>a,b</sup>	Media	93,8000	22,6000
	Desviación Estándar	3,32093	2,09762
Diferencia de Extremos	Absoluta	,134	,242
	Positiva	,134	,159
	Negativa	-,099	-,242
Kolmogorov-Smirnov Z		,518	,938
Significación Asintótica. (2-tailed)		,951	,342
a. La prueba sigue una distribución normal.			
b. Calculado desde los datos.			

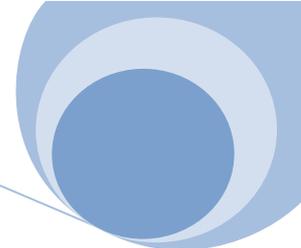
Tabla 3.2 Prueba de Kolmogorov-Smirnov para la Variable 1.

La prueba de Kolmogorov-Smirnov para la Variable 2 utilizando PASW Statistic 18 arrojó el siguiente reporte:

		Antes	Después
N		15	15
Parámetros Normales <sup>a,b</sup>	Media	63,3333	4,2667
	Desviación Estándar	5,69043	3,03472
Diferencia de Extremos	Absoluta	,149	,172
	Positiva	,121	,172
	Negativa	-,149	-,149
Kolmogorov-Smirnov Z		,575	,668
Significación Asintótica. (2-tailed)		,895	,764
a. La prueba sigue una distribución normal.			
b. Calculado desde los datos.			

Tabla 3.3 Prueba de Kolmogorov-Smirnov para la Variable 2

# Capítulo 3. Construcción del Sistema.



Puede concluirse que no se rechaza la hipótesis nula, demostrando que ambas variables siguen una distribución normal, por tanto al cumplirse este supuesto puede realizarse la Prueba T.

La Prueba T se realiza con el objetivo de demostrar que existe una diferencia significativa de un antes con respecto a un después.

Considerando como hipótesis nula que no hay diferencias significativas entre ellas y la hipótesis alternativa que existen diferencias significativas entre ambas variables y utilizando un nivel de significación de 5%, es decir  $\alpha = 0.05$ , se obtuvo el siguiente reporte de datos.

**Paired Samples Test**

	Paired Differences					t	df	Sig. (2-tailed)
	Mean	Std. Deviation	Std. Error Mean	95% Confidence Interval of the Difference				
				Lower	Upper			
Pair 1 antes - despues	71,20000	4,53872	1,17189	68,68654	73,71346	60,756	14	,000

Tabla 3.4 Resultado de la Prueba T para la variable 1.

**Paired Samples Test**

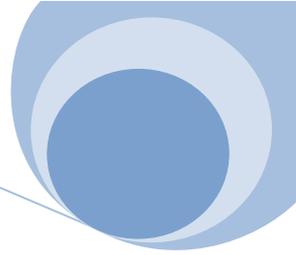
	Paired Differences					t	df	Sig. (2-tailed)
	Mean	Std. Deviation	Std. Error Mean	95% Confidence Interval of the Difference				
				Lower	Upper			
Pair 1 antes - despues	59,06667	6,52979	1,68598	55,45059	62,68274	35,034	14	,000

Tabla 3.5 Resultado de la prueba T para la variable 2.

Puede concluirse que se rechaza la hipótesis nula, y se acepta la hipótesis alternativa, ya que el nivel de significación es mayor que la significación asintótica demostrando que hay diferencias significativas entre ambas medias.

Por lo analizado y demostrado anteriormente se puede afirmar que el sistema realiza la gestión de la información relacionada con el control del transporte en la Universidad de Cienfuegos en menos tiempo que antes de ser informatizado.

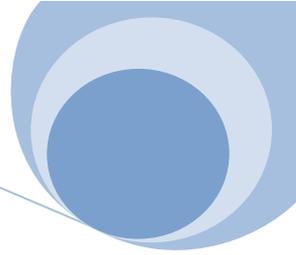
# Capítulo 3. Construcción del Sistema.



## 3.9 Conclusiones

En este capítulo se abordaron temas como la descripción de la arquitectura propuesta, diagrama de clases, estándar de implementación, seguridad del sistema, tratamiento de errores y se elaboró el diagrama de despliegue. Adicionalmente se realizó la validación de la solución propuesta utilizando los métodos estadísticos de KS y Prueba T.

# Capítulo 3. Construcción del Sistema.

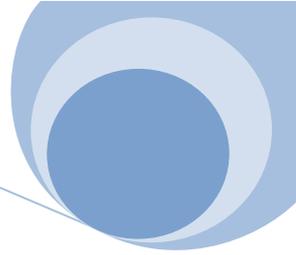


## Conclusiones

Luego de la culminación de la investigación para la automatización de los procesos relacionados con la gestión del transporte en la Universidad de Cienfuegos se puede afirmar que se obtuvo los siguientes resultados:

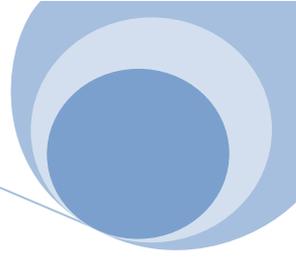
- Se realizó el análisis y diseño de la aplicación propuesta utilizando el lenguaje de modelado UML y la metodología FDD, haciendo uso de la herramienta Visual Paradigm.
- Se implementó una aplicación ajustada a las particularidades del Departamento de Transporte de la Universidad de Cienfuegos utilizando el lenguaje de programación PHP, el gestor de bases de datos MySQL, el IDE de programación NetBeans 6.8 y se aprovecharon las potencialidades del framework CodeIgniter.
- Se validó la aplicación propuesta utilizando métodos estadísticos para demostrar la existencia de una diferencia significativa entre los tiempos antes y después de desplegar el sistema.

# Capítulo 3. Construcción del Sistema.



## Recomendaciones

- Adoptar esta investigación como el comienzo de un proyecto mayor donde se terminen de automatizar todas las funciones del departamento de transporte.
- Desplegar la aplicación a todos los departamentos de transporte de las universidades del país.



## Referencias Bibliográficas

- [1] A. L. S. Chitolie, "Sistema de Gestión de los Datos de Producción en la Empresa Glucosa de Cienfuegos.," Práctica, Departamento de Informática, Universidad de Cienfuegos, Cienfuegos, 2011.
- [2] D. V. Pires. (2011, 12/12/2011). *Universidad de Cienfuegos "Carlos Rafael Rodríguez"*. Available: [www.ucf.edu.cu](http://www.ucf.edu.cu)
- [3] (2011, 26/09/2011). *Intera System*. Available: [www.usa.iterasystem.com/soffflot](http://www.usa.iterasystem.com/soffflot)
- [4] M. D. Q. Alahmar, "Sistema Informático para la Gestión y control de los insumos en la actividad de transporte en la Empresa de Telecomunicaciones de Cuba Sociedad Anónima ETECSA.," Práctica, Departamento de Informática, Universidad de Cienfuegos, Cienfuegos, 2011.
- [5] A. Molpeceres, "Procesos de desarrollo: RUP,XP y FDD."
- [6] G. B. I. Jacobson, y J. Rumbaugh, " El Proceso Unificado de Desarrollo de Software," ed: Addison Wesley, 1999.
- [7] *Visual Paradigm for UML*. Available: <http://www.freedownloadmanager.org/es/>
- [8] U. y. P. Craig Larman, "Una introducción al análisis y diseño orientado a objetos y al proceso unificado," 2000.
- [9] C. Romero. *Sistema Gestor de Bases de Datos*. Available: <http://personal.lobocom.es/claudio/sql001.htm>
- [10] L. w. y. I. thomson. (2012, *Desarrollo web con PHP y MySQL*.
- [11] R. Ellis, "Guia Definitiva de CodeIgniter."
- [12] M. Á. Álvarez. *Qué es Javascript*. Available: <http://www.desarrolloweb.com/articulos/25.php>
- [13] (2010, *Netbeans 6.8 liberado*. Available: <http://blogcultura.com/java/netbeans-6-8-liberado>
- [14] (2011, *GIMP 2.8*. Available: <http://www.gimp.org/>
- [15] M. Peralta, "ESTIMACIÓN DEL ESFUERZO BASADA EN CASOS DE USO."
- [16] L. Toledo, "Conferencia 2 Diseño Conceptual de Bases de Datos," ed, 2006.
- [17] "Manual de SQL," 2005.
- [18] G. B. I. Jacobson, y J. Rumbaugh, "El Proceso Unificado de Desarrollo de Software," 2002.