

HERRAMIENTA DE SIMULACIÓN PARA OPTIMIZAR LA PROGRAMACIÓN DE  
CARGUE DE UNIDADES EN CAMIONES NODRIZA EN LA EMPRESA  
TRANSPORTES VIGÍA SAS.

Ceron Fuentes Cristian Alexander  
Diaz Latorre Jeiner Farley

Cooperacion universitaria “Minuto de Dios”  
Facultad de Ingenieria  
Tecnologia en informática  
Soacha, Cundinamarca  
2012

HERRAMIENTA DE SIMULACIÓN PARA OPTIMIZAR LA PROGRAMACIÓN DE  
CARGUE DE UNIDADES EN CAMIONES NODRIZA EN LA EMPRESA  
TRANSPORTES VIGÍA SAS

Cerón Fuentes Cristian Alexander

Díaz Latorre Jeiner Farley

Proyecto de grado para optar por título de Tecnólogo en Informática

Asesor Yobany Gomez Garcia Ingeniero de Sistemas

Corporación universitaria "Minuto de Dios"

Facultad de Ingeniería

Tecnología en informática

Soacha, Cundinamarca

2012

Nota de aceptación

---

---

---

---

Presidente del Jurado

---

Jurado

---

Jurado

Soacha 07/03/2012 - (21/06/12)

A nuestros padres  
Por el apoyo brindado  
En el transcurso  
de la carrera.

## AGRADECIMIENTOS

El presente proyecto es un esfuerzo en el cual participan varias personas leyendo, opinando, corrigiendo, teniéndome paciencia.

Agradecemos al profesor Yobany Gomez Garcia por la paciencia y por la dirección de este trabajo. Al Ingeniero Pablo Fonseca por los consejos en la elaboración del documento, el apoyo y el ánimo que nos brindó

## CONTENIDO

	Pág.
INTRODUCCION	
1 PROBLEMA DE INVESTIGACIÓN	15
1.1 ANTECEDENTES	15
1.2 FORMULACIÓN DEL PROBLEMA	15
1.3 JUSTIFICACIÓN	16
1.4 OBJETIVO GENERAL	17
1.5 OBJETIVOS ESPECIFICOS	17
1.6 LIMITACIONES PREVISIBLES	18
2 MARCO DE REFERENCIA	19
2.1 MARCO TÉCNICO	19
2.1.1 Lenguaje php	19
2.1.2 Lenguaje html	19
2.1.3 Aplicación Web	19
2.1.4 Mysql	20
2.1.5 Ajax	20
2.1.6 Java script	21
2.2 MARCO TEÓRICO	21
2.2.1 Simulación	21
2.2.2 Base de datos	22
2.2.3 Estructura de base de datos	22
2.2.4 Análisis del problema	23
2.2.5 Desarrollo de la solución	23
2.2.6 Construcción de la solución	23

2.2.7 Pruebas	23
3. CONSIDERACIONES DEL DISEÑO	24
3.1 REQUERIMIENTOS TÉCNICOS	24
3.2 ALCANCE DEL SISTEMA	24
3.3 REQUERIMIENTOS FUNCIONALES	25
3.3 REQUERIMIENTOS NO FUNCIONALES	25
4. DISEÑO DEL SISTEMA	26
4.1 DESCRIPCIÓN DEL SISTEMA	26
4.2 ARQUITECTURA DEL SISTEMA	26
4.3 MODELO DE CASOS DE USO	27
4.3.1 Descripción general de los Actores.	27
4.3.2 Diagramas del Modelo de Casos de Uso	27
4.3.3 Modelos Objeto	36
4.3.4 Modelos dinámicos	48
4.4 METODOLOGIA	49
4.4.1 Descripción del proceso	50
4.4.2 Variables a tener en cuenta	51
4.4.3 Modelo de simulación	52
5 CONCLUSIONES	54
Bibliografía	55
ANEXOS	56

## LISTADO DE TABLAS

Tabla1. Arquitectura física del Software	26
Tabla 2.Caso de uso. Subsistema modulo admón. lotes.	28
Tabla 3.Caso de uso. Subsistema modulo admón. usuarios.	29
Tabla 4.Caso de uso. Subsistema modulo admón. vehículos nuevos.	31
Tabla 5.Caso de uso. Subsistema. Cambio contraseña.	33
Tabla 6.Caso de uso. Subsistema. Logueo.	34
Tabla 7. Atributos de la clase usuarios.	35
Tabla 8.Atributos de la clase perfiles	37
Tabla 9. Atributos de la clase ciudades.	38
Tabla 10. Atributos de la clase destinos	38
Tabla 11. Atributos de la clase despachos	39
Tabla 12. Atributos de la clase embarques.	39
Tabla 13. Atributos de la clase trailers.	40
Tabla 14. Atributos de la clase vehículos_empresa.	43
Tabla 15. Atributos de la clase lotes.	44
Tabla 16. Atributos de la clase vehículos_lotes.	44
Tabla 17. Atributos de la clase dimencion_vehiculos.	45
Tabla 18. Atributos de la clase modelo.	47
Tabla 19. Atributos de la clase marca.	47

## TABLA DE ILUSTRACIONES

Ilustración 1. Diagrama de caso de uso:sistema general.	28
Ilustración 2. Diagrama de caso de uso: subsistema modulo admon.Lotes.	29
Ilustración 3. Diagrama de caso de uso: subsistema modulo admon.usuarios.	30
Ilustración 4. Diagrama de caso de uso: subsistema modulo admon. Vehículos nuevos.	32
Ilustración 5. Diagrama de caso de uso: subsistema. Cambio contraseña.	33
Ilustración 6. Diagrama de caso de uso: subsistema. Logueo	34
Ilustración 7. Diagrama de Clases.	36
Ilustración 8. Tabla usuarios	37
Ilustración 9. Tabla Perfiles.	38
Ilustración 10. Tabla ciudades.	38
Ilustración 11. Tabla destinos.	39
Ilustración 12. Tabla despachos.	39
Ilustración 13. Tabla embarques.	40
Ilustración 14. Tabla trailers.	41
Ilustración 15. Tabla vehículos_empresa.	43
Ilustración 16. Tabla lotes.	44
Ilustración 17. Tabla vehículos_lotes.	45
Ilustración 18. Tabla dimencion_vehiculos.	45
Ilustración 19. Tabla modelo.	46
Ilustración 20. Tabla marca.	47
Ilustración 21. Modelo Dinámico: secuencia del sistema.	48

## **GLOSARIO.**

**AROS:** Estructura de forma circular adherida por medio de pernos al último eje de la estructura niñera con el fin de servir de palanca para subir o bajar la plataforma superior de la estructura niñera.

**CAPACETE:** Estructura diseñada arriba del cabezote o unidad de tracción que contiene y transporta una única unidad.

**CABEZOTE:** Unidad de tracción de uno o dos ejes diseñada para halar carga pesada (camión).

**CODIGO:** es un conjunto de líneas de texto que son las instrucciones que debe seguir la computadora para ejecutar dicho programa. Por tanto, en el código fuente de un programa está descrito por completo su funcionamiento.

**DEPURAR:** es el proceso de identificar y corregir errores de programación.

**DIAGRAMAS UML:** es un conjunto de herramientas, que permite modelar (analizar y diseñar) sistemas orientados a objetos.

**EQUIPO NODRIZA O NIÑERA:** estructura diseñada y utilizada para contener y transportar unidades (autos) de diferentes características.

**GANCHOS:** Elementos incluidos en las reatas elaborados en hierro o acero de forma curva capaces de sujetar las bandas de rodamiento de las estructuras niñera por medio de huecos en las mismas.

**GUAYAS:** Conjunto de alambres enrollados de forma helicoidal sobre un centro o alma que puede variar entre yute o acero, es utilizado para la elevación mecánica de la rampa superior por medio de forma manual a unos aros sujetos por pernos o tornillos y utilizando el desplazamiento del equipo de tracción (cabezote).

**HUMILLAR:** Termino utilizado en el medio del transporte de autos para describir que la unidad se bajará de forma manual por medio de cinchas o reatas para reducir altura entre las plataformas, este procedimiento solo se realiza a unidades de gran altura como camionetas o vans y solo se puede hacer sosteniendo las unidades del chasis y las plataformas fijas de la estructura niñera.

**MALACATE:** Instrumento mecánico utilizado para enrollar la reata o cincha sobre un cilindro creando presión al halar de forma continua una palanca con engranaje en forma de sierra que le impide devolverse en sentido contrario al que se recoge.

**NAVEGADOR:** El navegador interpreta el código, HTML generalmente, en el que está escrita la página web y lo presenta en pantalla permitiendo al usuario interactuar con su contenido y navegar hacia otros lugares de la red mediante enlaces o hipervínculos.

**PASADORES:** Cilindros metálicos hechos en hierro los cuales pasan las torres para no permitir que la estructura superior caiga al mismo nivel que la primera.

**PROGRAMADOR:** Persona encargada de asignar el cargue de unidades.

**PUNTOS DE ANCLAJE:** Huecos elaborados en las plataformas para que los ganchos, cintas y malacate permitan una correcta sujeción de las llantas de cada unidad a transportar.

**RAMPA DE CARGUE:** Estructura, plano o terreno inclinado que sirve para cargar y descargar las unidades a las plataformas.

**RAMPA DE PASO:** Estructura removible para pasar una unidad ( auto) de la plataforma superior al capacete.

**REATA O CINCHA:** Cinta fabricada en polyester que es un material sintético de alta tenacidad que permite llegar a resistencias altas de fuerza y diseñado para sujetar las unidades a la estructura que moviliza las unidades, cuenta con varios puntos de anclaje llamados ganchos.

**SAS:** Sociedades por Acciones Simplificadas.

**SISTEMA:** es el conjunto de partes interrelacionadas, hardware, software y de recurso humano que permite almacenar y procesar información.

**SOFTWARE:** comprende el conjunto de los componentes lógicos necesarios que hacen posible la realización de tareas específicas.

**TORRE:** Parte de la estructura de la niñera la cual soporta el peso de la rampa superior de cargue y que cuenta con diferentes puntos de graduación para los pasadores.

## RESUMEN

El proyecto **“HERRAMIENTA DE SIMULACIÓN PARA OPTIMIZAR LA PROGRAMACIÓN DE CARGUE DE UNIDADES EN CAMIONES NODRIZA EN LA EMPRESA TRANSPORTES VIGÍA SAS”** tiene como propósito la elaboración de una aplicación web que permita la simulación de cargue de unidades en camiones nodriza, la metodología propuesta en el documento para el desarrollo: **“METODOLOGIA DE DESARROLLO XP (programación extrema)”**.

A lo largo de este documento, se podrá visualizar los componentes principales del proyecto: planteamiento del problema, las consideraciones de diseño y el diseño del sistema donde se encontraran (descripción del sistema, arquitectura del sistema, diagramas del modelo de casos de uso, modelos objeto, modelos dinámicos, descripción del proceso, variables a tener en cuenta, modelo de simulación).

## ABSTRACT

The "**SIMULATION TOOL FOR OPTIMIZING SCHEDULE OF UNITS IN CHARGE NURSE TRUCKS IN THE COMPANY TRANSPORTES VIGÍA SAS**" aims at developing a web application that allows the simulation of charge nurse truck units, the methodology proposed in the document for **development "DEVELOPMENT METHODOLOGY XP (extreme programming)."**

Throughout this document, you can view the project's main components: a problem formulation, design considerations and system design where found (system description, system architecture diagrams use case model, models object, dynamic models, description of the process variables to consider, model simulation).

## INTRODUCCIÓN

Este proyecto busca plasmar la factibilidad de implementar una herramienta informática para la optimización del cargue de unidades en camiones nodriza, debido a que han dejado grandes pérdidas debido a el daño de estos vehículos al momento del cargue, ya que estas unidades son golpeadas al no dejar el espacio adecuado entre ellas.

El semillero de investigación Fénix dentro de los proyectos que se han desarrollado por estudiantes de logística, en búsqueda de mejorar la calidad en el sector logístico de las empresas ha planteado el desarrollo de una herramienta de simulación, para ello ha propuesto la vinculación de algunos estudiantes de tecnología en informática a este semillero de investigación, desarrollando así un proyecto interdisciplinario.

Aprovechando la revolución tecnología que esta sucediendo en las computadoras, nos permite a través del desarrollo de aplicaciones optimizar algunas de las actividades que son llevadas a cabo en las empresas, evitando el desperdicio de recursos económicos, naturales y humanos.

La solución informática planteada en este proyecto tiene como finalidad brindar una ayuda a los programadores de la empresa transportes vigía SAS en el momento de realizar un despacho de vehículos a otras zonas del país.

# 1. PROBLEMA DE INVESTIGACIÓN

## 1.1 ANTECEDENTES

No se tiene conocimiento de este tipo de sistemas en Colombia, tal vez por lo específico de la aplicación. Empresas como Data-line usan este tipo de sistemas para monitorear los procesos.

El proyecto de la herramienta de simulación es novedoso, ya que plantea la aplicación de herramientas informáticas y web a una programación diaria de un proceso industrial.

El valor agregado de este proyecto es la aplicación de herramientas web a un problema diario dentro de los procesos productivos de una empresa Colombiana.

## 1.2 FORMULACIÓN DEL PROBLEMA

¿Que herramientas técnicas y teóricas se pueden aplicar para mejorar las falencias en la programación de camiones nodriza en la empresa Transportes Vigía SAS?

La programación diaria de unidades (vehículos) por parte de los programadores de carga en la Compañía Transportes Vigía S.A.S, presenta diversos inconvenientes especialmente desde hace 2 años cuando la compañía incremento su capacidad instalada en un 60%.

En primer lugar se ve retrasada la programación pues se realiza conforme a la experiencia y conocimiento empírico del programador, pero no existen herramientas de apoyo que permitan contemplar diferentes opciones para un mismo problema de cargue. Sin embargo dejar la programación en manos de la

experiencia no es sano ya que los errores humanos son diarios y se deben al estado de ánimo de cada persona, o a la capacidad propia de desarrollar métodos eficientes.

Además la programación óptima de unidades en las estructuras desarrolladas para transporte y distribución de las mismas en equipos llamados niñeras o camiones nodrizas, es afectada por diferentes formas en las estructuras, las diferencias en las dimensiones como la longitud, altura, ancho, ángulos de inclinación de sus rampas de cargue, puntos de sujeción de las unidades, distancia mínima y máxima permitida, teniendo en cuenta la diversidad de fabricantes.

Por lo expuesto anteriormente se debe desarrollar una herramienta de simulación de apoyo para el personal de programación que junto con el personal adecuado e instruido pueda responder a los requerimientos y expectativas del mercado.

### **1.3 JUSTIFICACIÓN**

Es necesario que en el proceso de distribución se cometan los mínimos errores para realizar una operación exitosa y así maximizar la distribución y entregas de calidad con un apremiante, aplicar tecnología para desarrollar Herramientas que en cada parte se reduzcan daños de unidades y se aumente la operatividad del cargue de unidades, aumentar la competitividad, agilizar el cargue de unidades, reducir en gran parte los daños, sin embargo hay que tener en cuenta que la naturaleza de estas unidades es estar expuestas a múltiples daños por encontrarse a la vista y sin protección alguna en su integridad física, sin embargo serían menores los daños si se cuenta con la herramienta de distribución por software ya que nos mostraría de manera rápida y segura la ubicación de cada unidad, modelo, peso, distancia entre unidades, puntos de sujeción, se reducen los roces entre los departamentos de operaciones y administración, la optimización en el transporte de unidades desde puertos sería programado con antelación con los equipos idóneos, las reclamaciones por parte de los concesionarios sería menor ya que la calidad se vería reflejada en las entregas, las cualidades de los conductores serían puestas a prueba tomando como base el resultado de la ubicación de las unidades por parte del programa paralelo a las creencias y tendencias de ubicación de cada conductor, en esta parte se ve que

son mayores los beneficios en comparación con la idoneidad del programador empírico.

#### **1.4 OBJETIVO GENERAL.**

Elaborar una herramienta de simulación orientada a la Web y así optimizar la programación de cargue de unidades en camiones nodriza para la empresa transportes Vigía SAS.

#### **1.5 OBJETIVOS ESPECÍFICOS**

- Analizar el completo funcionamiento del sistema que es llevado actualmente por la empresa referente a la programación de cargue de unidades en camiones nodriza.
- Plantear la solución que mejor se adapte a las necesidades y requerimientos de la empresa y los lenguajes de programación mas adecuados para la solución planteada.
- Elaborar diagramas UML y modelos entidad relación sobre las bases de datos en las que se elaborara el sistema informático.
- Codificar los diagramas UML y los modelos entidad relación, en un software con los lenguajes de programación previamente elegidos para la programación.
- Depurar el código y verificar su perfecto funcionamiento sobre los requisitos establecidos por el cliente.

## **1.6 LIMITACIONES PREVISIBLES**

Debido al corto tiempo para el desarrollo del aplicativo web solo será entregado un prototipo funcional que permita la simulación de cargue de unidades en camiones nodriza.

La implementación del prototipo corre por cuenta de la empresa vigía SAS, la adición de nuevas herramientas para el sistema desarrollado no se incluyen en lo pactado con el semillero de investigación FÉNIX, una vez el prototipo sea entregado como por ejemplo ese el modulo ingreso de tráileres.

## 2 MARCO DE REFERENCIA

### 2.1 MARCO TÉCNICO

**2.1.1 Lenguaje php.** Es un lenguaje de programación interpretado (Lenguaje de alto rendimiento), diseñado originalmente para la creación de páginas web dinámicas. Se usa principalmente para la interpretación del lado del servidor (*server-side scripting*) pero actualmente puede ser utilizado desde una interfaz de línea de comandos o en la creación de otros tipos de programas incluyendo aplicaciones con interfaz gráfica usando bibliotecas.

**2.1.2 Lenguaje html.** Lenguaje de marcado de hipertexto, es el lenguaje de marcado predominante para la elaboración de páginas web. Es usado para describir la estructura y el contenido en forma de texto, así como para complementar el texto con objetos tales como imágenes. El HTML se escribe en forma de «etiquetas», rodeadas por corchetes angulares (<,>). HTML también puede describir, hasta un cierto punto, la apariencia de un documento, y puede incluir un *script* (por ejemplo JavaScript), el cual puede afectar el comportamiento de navegadores web y otros procesadores de HTML.

**2.1.3 Aplicación Web.** 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. Las aplicaciones web son populares debido a lo práctico del navegador web como cliente ligero, así como a la facilidad para actualizar y mantener aplicaciones web sin distribuir e instalar software a miles de usuarios potenciales. Existen aplicaciones como los web mails, wikis, web logs, tiendas en línea y la propia Wikipedia que son ejemplos bien conocidos de aplicaciones web. Es importante mencionar que una página Web puede contener elementos que permiten una comunicación activa entre el usuario y la información. Esto permite que el usuario acceda a los datos de modo interactivo, gracias a que la página responderá a cada una de sus

acciones, como por ejemplo rellenar y enviar formularios, participar en juegos diversos y acceder a gestores de base de datos de todo tipo.

**2.1.4 Mysql.** Es un sistema de gestión de bases de datos relacional, multihilo y multiusuario con más de seis millones de instalaciones. MySQL AB desde enero de 2008 una subsidiaria de Sun Microsystems y ésta a su vez de Oracle Corporation desde abril de 2009 desarrolla MySQL como software libre en un esquema de licenciamiento dual.

Por un lado se ofrece bajo la GNU GPL para cualquier uso compatible con esta licencia, pero para aquellas empresas que quieran incorporarlo en productos privativos deben comprar a la empresa una licencia específica que les permita este uso. Está desarrollado en su mayor parte en ANSI C.

**2.1.5 Ajax.** Es realmente muchas tecnologías, cada una floreciendo por su propio mérito, uniéndose en poderosas nuevas formas. AJAX incorpora:

- presentación basada en estándares usando XHTML y CSS;
- exhibición e interacción dinámicas usando el Document Object Model;
- Intercambio y manipulación de datos usando XML and XSLT;
- Recuperación de datos asincrónica usando XMLHttpRequest;
- y JavaScript poniendo todo junto.

El modelo clásico de aplicaciones Web funciona de esta forma: La mayoría de las acciones del usuario en la interfaz disparan un requerimiento HTTP al servidor web. El servidor efectúa un proceso (recopila información, procesa números, hablando con varios sistemas propietarios), y le devuelve una pagina HTML al cliente.

Este es un modelo adaptado del uso original de la Web como un medio hipertextual, pero como fans de “The Elements of User Experience” sabemos, lo que hace a la Web buena para el hipertexto, no la hace necesariamente buena para las aplicaciones de software.

**2.1.6 JavaScript.** Es un lenguaje de programación interpretado, dialecto del estándar ECMA Script. Se define como orientado a objetos, basado en prototipos, imperativo, débilmente tipado y dinámico.

Se utiliza principalmente en su forma del lado del cliente (client-side), implementado como parte de un navegador web permitiendo mejoras en la interfaz de usuario y páginas web dinámicas, en bases de datos locales al navegador. Aunque existe una forma de JavaScript del lado del servidor (Server-side JavaScript o SSJS). Su uso en aplicaciones externas a la web, por ejemplo en documentos PDF, aplicaciones de escritorio es también significativo.

## **2.2 MARCO TEÓRICO**

**2.2.1 Simulación.** Un simulador es por lo regular un aparato o programa informático desarrollado para recrear las sensaciones, condiciones y comportamientos que en realidad no están sucediendo pero que en condiciones normales podrían llegar a darse mostrando muchas veces entornos sintéticos para simular en apariencia de lo que en realidad podría suceder mostrándolo en un computador a más bajo costo que si se probara en la realidad con los equipos a evaluar.(COSS BU, pág. 17).

En la actualidad se cuentan con diferentes tipos de simuladores como son los de conducción, en el cual se puede recrear la forma de conducir desde una motocicleta hasta un camión, también se cuenta con simuladores de vuelo que permiten dominar el mundo de la aviación, simuladores de vida, que son aquellos que permiten controlar una persona virtual y su vida diaria, simuladores de negocios en los cuales se pueden tomar varios roles en las funciones típicas de los negocios, simuladores médicos con los cuales puede darse un dictamen sobre un paciente virtual con diferentes casos clínicos, este le permite a los futuros médicos prepararse para cuando en realidad encuentre un paciente con estos mismos síntomas, simuladores musicales los cuales reproducen los sonidos de un instrumento real en uno de juguete.(COSS BU, págs. 18,19).

La simulación es una técnica numérica para conducir experimentos en una computadora digital, comprendiendo relaciones lógicas y matemáticas, necesarias para describir el comportamiento de sistemas complejos en periodos de tiempo (COSS BU, 1999).

La técnica de la simulación se utiliza desde hace mucho tiempo como herramienta fundamental para el diseñador, es herramienta viable para experimentar con diseños específicos, esto quiere decir que se puede imitar el desempeño de un trabajo controlado con el fin de estimar cuál sería su desempeño real, después de diseñar el modelo y de probarse se le pueden dar los últimos detalles al diseño final. Para utilizar la simulación en un estudio de investigación siempre irá precedido de un análisis teórico preliminar para el desarrollo de un diseño básico, por eso la simulación es una técnica de muestreo estadístico controlada para estimar desempeño de sistemas estocásticos. (Frederick S. Hiller, 1997, págs. 900-901)

Así bien la simulación es una herramienta habitualmente usada para estimar desempeños en sistemas complejos cuando se requiere de usar nuevos diseños o políticas de operación, la simulación es una herramienta muy adaptable pero de ninguna manera es la verdad absoluta, es una técnica imprecisa que solo proporciona estimaciones estadísticas y no resultados exactos, la simulación permite experimentar con las políticas o los sistemas propuestos pero sin tener que hacer cambios en el sistema real.

**2.2.2 Bases de datos.** Es un conjunto de datos pertenecientes a un mismo contexto y almacenados sistemáticamente para su posterior uso. En este sentido, una biblioteca puede considerarse una base de datos compuesta en su mayoría por documentos y textos impresos en papel e indexados para su consulta.

**2.2.3 Estructura de una bases de datos.** La estructura de la base de datos es bastante sencilla. Las convenciones utilizadas aparecen implícitamente en este documento. Por ejemplo, la mayoría de los objetos se indexan con un entero auto incrementado cuyo nombre es de tipo id\_objet, y que se declara como clave primaria en la tabla apropiada.

**2.2.4 Análisis del problema.** Su propósito general es ayudar al programador a llegar a la comprensión de la naturaleza del problema ingresando los datos necesarios para la resolución del mismo y dando paso al proceso con el cual los datos ingresados empiezan a formar la solución. (BUCARELLY, 2005, pág. 143)

**2.2.5 Desarrollo de la solución.** Los problemas pueden resolverse más fácilmente en un ordenador cuando se subdividen en segmentos más fáciles de resolver que en el original, a esto se le llama diseño descendente (top down design). (BUCARELLY, 2005, pág. 148)

**2.2.6 Construcción de la solución.** En esta fase se compilan todos los datos ingresados y se revisa que no existan errores para dar a correr el programa. (BUCARELLY, 2005, pág. 148)

**2.2.7 Prueba.** En esta parte se da a correr el programa y se revisa que los resultados sean satisfactorios y lo que en realidad el usuario requiere, se debe realizar la búsqueda de errores normales como son la semántica, o de tipo lógico o de ejecución. (BUCARELLY, 2005, pág. 148)

### **3. CONSIDERACIONES DEL DISEÑO**

#### **3.1 REQUERIMIENTOS TÉCNICOS.**

- **Aplicación Web.**

El aplicativo deberá ser desarrollado de tal forma que sea accesible vía internet. Deberá ser accesible a través del navegador Mozilla Firefox.

- **Conexiones con base de datos.**

El aplicativo debe diseñarse de tal forma que pueda acceder a la base de datos elegida por el semillero de investigación FENIX, que se encuentra implementada en Mysql.

- **Desarrollo en php y ajax.**

El prototipo debe ser desarrollado utilizando php y Ajax debido a que el aplicativo debe de ser de rápido acceso a los registros alojados en la base de datos, especialmente en el momento de actualizar o el ingreso de nuevos datos.

#### **3.2 ALCANCE DEL SISTEMA**

El aplicativo podrá ser accedido por los usuarios desde el navegador web Mozilla Firefox disponible desde cualquier sistema operativo moderno.

Para asegurar la integridad del sistema ya que con otros navegadores puede generar inconvenientes con los css, el aplicativo solo puede ser accedido por personal autorizado por la empresa transportes vigía SAS, al cual previamente se le ha asignado un usuario y clave de accesos.

El alcance global de este proyecto puede solucionar los problemas de cargue no solo de la empresa vigia SAS, sino que tambien puede ser implementado en otras empresas de transporte de unidades en camiones nodriza. Con algunos ajustes en el software este sistemas puede ser implementado en otros sectores de la industria de transporte en tractocamiones.

### **3.3 REQUERIMINETOS FUNCIONALES**

- El software debe contar con un manual de usuario que describa todas sus funciones y herramientas.
- El software debe permitir realizar cualquier clase de ingreso de nuevos vehículos, así como la interacción de los vehículos almacenados en lotes.
- El software debe entregar los resultados de simulación sobre la manera mas optima de carga de los vehículos encontrados en los lotes.

### **3.4 REQUERIMIENTOS NO FUNCIONALES**

- Personal capacitado para el funcionamiento del aplicativo.
- Experiencia del usuario.
- Identificar políticas de mantenimiento de información.
- El sistema debe visualizarse y funcionar en cualquier navegador, especialmente en Mozilla y crome.

## 4. DISEÑO DEL SISTEMA

### 4.1 DESCRIPCIÓN DEL SISTEMA

El sistema consiste en una aplicación web, es decir; aquella que puede accederse desde cualquier navegador de internet, aunque de preferencia se debe acceder a través de Mozilla Firefox. Este sistema accede a la base de datos alojada en el host.

A través del aplicativo se realizara el ingreso de nuevas marcas de vehículos por casa matriz para la creación de lotes de unidades y por medio de la simulación nos arrojará el cargue mas optimo para ser enviado al destino seleccionado.

### 4.2 ARQUITECTURA DEL SISTEMA

Componente	Tecnología
Sistema operativo Cliente	Cualquiera que acepte TCP/IP
Sistema operativo servidor de base de datos	Windows Xp
Sistema operativo servidor web	Windows Xp
Software en el lado del cliente	Un navegador web avanzado que soporte peticiones AJAX preferiblemente Mozilla Firefox por la compatibilidad de css3
Software del lado del servidor	Servidores Apache Base de datos Msql.

Tabla1. Arquitectura física del Software.

### **4.3 MODELO DE CASOS DE USO**

**4.3.1 Descripción general de los Actores.** El aplicativo está dirigido a la empresa de transportes vigía SAS, para la optimización de carga de unidades a camiones nodriza. No es necesario conocer el funcionamiento total para manejar correctamente el aplicativo; pero solo puede ser utilizado por el programador. El programador podrá acceder al aplicativo desde cualquier lugar con acceso a internet, sin embargo, deberán contar con una cuenta de usuario, la cual le será entregada por el administrador del sistema. Dado que el aplicativo está definido bajo 2 tipos de roles, es posible por parte del administrador restringir los permisos o herramientas a los que el programador tiene acceso.

**4.3.2 Diagramas del Modelo de Casos de Uso.** Se incluye la especificación de cada caso de uso con una descripción general por cada modulo, al igual que un curso normal de eventos. El modelo de casos de uso es el punto de partida para la construcción del diagrama de clases.



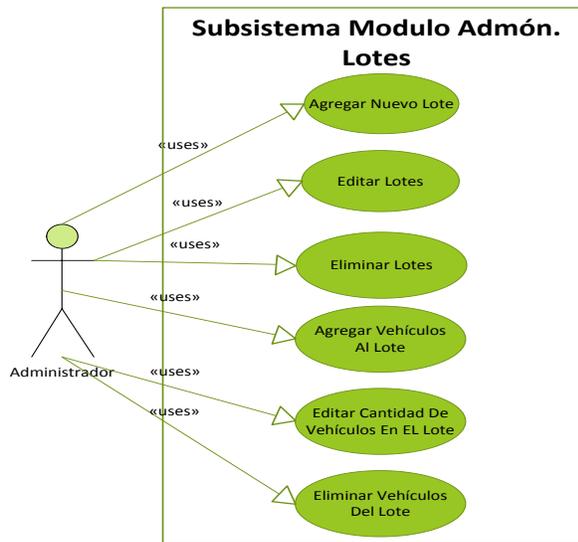


Ilustración 2. Diagrama de caso de uso: subsistema modulo admon.Lotes.

<b>SELECCIÓN PRINCIPAL</b>	
Caso de uso	subsistema modulo admon.Lotes
Actores	Administrador
Propósito	<p>El administrador puede crear, editar y eliminar lotes.</p> <p>También se puede agregar, editar y eliminar vehículos de lotes.</p>
Descripción	<p>Este subsistema nos permite apreciar como el administrador maneja el modulo de administración de lotes permitiendo el agregar nuevos lotes, edición de lotes y eliminación de lotes.</p>
<b>ACCIONES DEL SUBSISTEMA</b>	
1. Agregar nuevos Lotes	2. Editar Lotes
3. eliminar Lotes	4. Agregar vehículos a lotes
5. Editar vehículos en lotes	6. Eliminar vehículos en lotes

Tabla 2.Caso de uso. Subsistema modulo admón. lotes.

Adicionalmente el administrador puede agregar vehículos al lote, editar cantidad de vehículos en el lote y eliminación de vehículos en el lote.

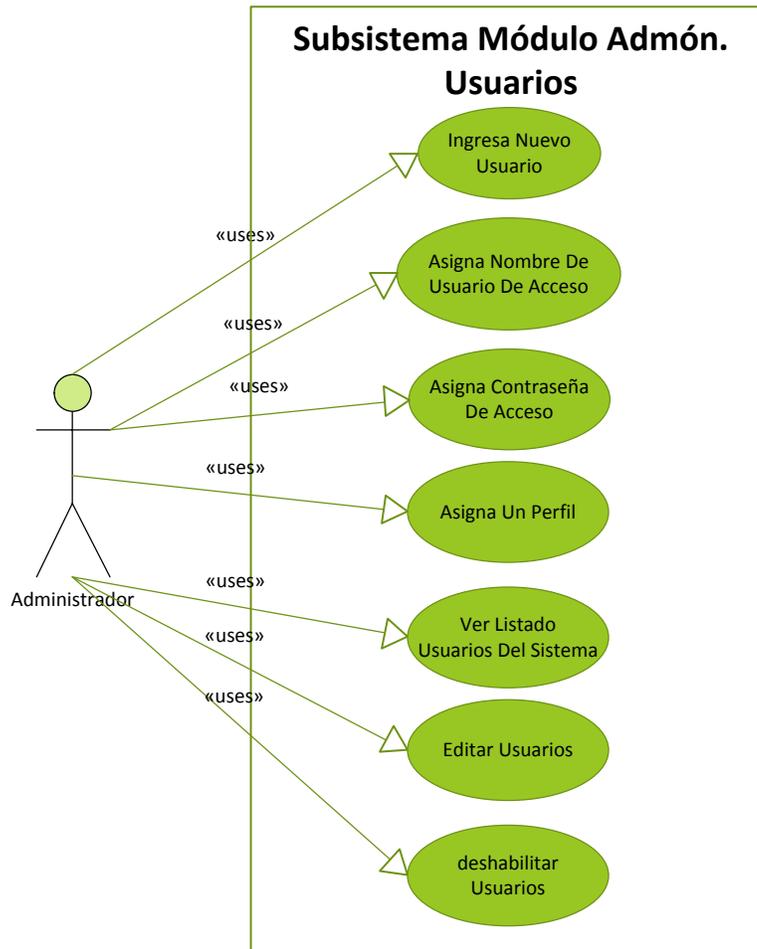


Ilustración 3. Diagrama de caso de uso: subsistema modulo admón. usuarios.

SELECCIÓN PRINCIPAL	
Caso de uso	Subsistema modulo admón. usuarios.
Actores	Administrador
	El administrador puede crear, asignar

Propósito	<p>nombre y contraseña de acceso.</p> <p>También se puede ver listado de usuarios, editar y deshabilitar usuarios.</p>
Descripción	<p>Este subsistema nos permite apreciar como el administrador maneja el modulo de administración de usuarios permitiendo el agregar nuevos usuarios, edición de usuarios, asignar nombre de usuario de acceso, asignar contraseña de acceso, asignar un perfil, ver el estado de usuario del sistema y deshabilitar de usuarios.</p>
<b>ACCIONES DEL SUBSISTEMA</b>	
1. Ingresar nuevos usuarios	2. Asignar nombre y contraseña de acceso
3. Asignar perfil	4. Ver listado de usuarios
5. Editar usuarios	6. deshabilitar usuarios

**Tabla 3.Caso de uso. Subsistema modulo admón. usuarios.**

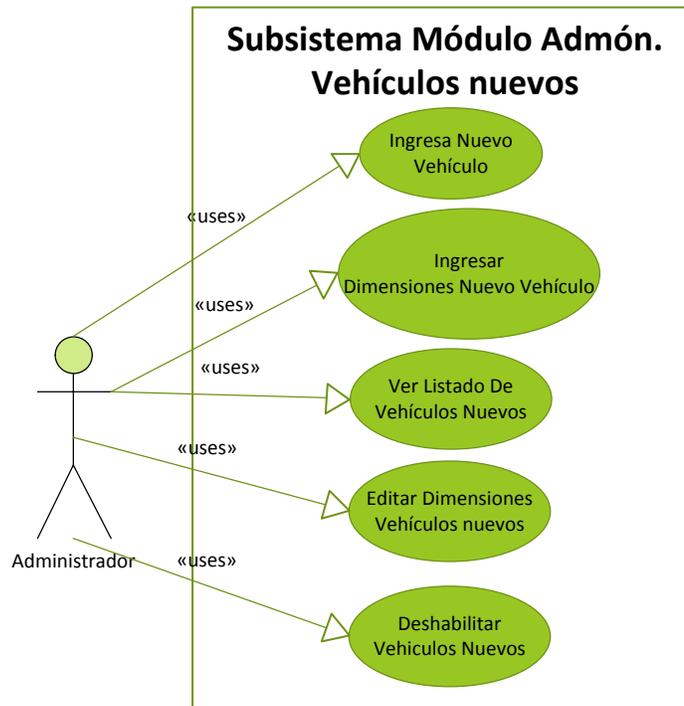


Ilustración 4. Diagrama de caso de uso: subsistema modulo admón. vehículos nuevos.

<b>SELECCIÓN PRINCIPAL</b>	
Caso de uso	subsistema modulo admón. vehículos nuevos.
Actores	Administrador
Propósito	<p>El administrador puede ingresar vehículos y sus dimensiones.</p> <p>También le permite ver el listado de los vehículos, editar sus dimensiones y deshabilitar vehículos.</p>
Descripción	<p>Este subsistema nos permite apreciar como el administrador maneja el modulo de administración de vehículos nuevos permitiendo el ingreso de vehículo, ingreso de dimensiones de vehículo, ver listado de vehículos, editar dimensiones vehículos y deshabilitar vehículos.</p>

ACCIONES DEL SUBSISTEMA	
1. Ingresar nuevos vehículos	2. ingresar dimensiones de vehículos
3. ver listado de vehículos	4. editar dimensiones de vehiculos
5. deshabilitar vehículos	

Tabla 4.Caso de uso. Subsistema modulo admón. vehículos nuevos.

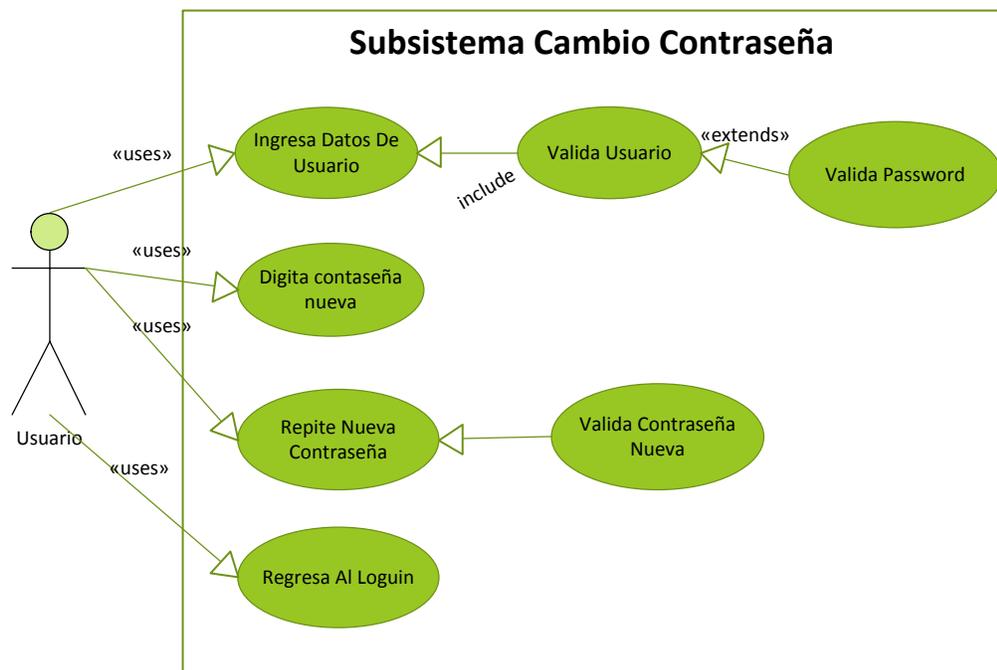


Ilustración 5. Diagrama de caso de uso: subsistema. Cambio contraseña.

SELECCIÓN PRINCIPAL	
Caso de uso	Subsistema. Cambio contraseña.
Actores	Administrador
Propósito	Ingreso datos de usuario al sistema. Cambio de contraseña.
	Este subsistema nos permite apreciar

Descripción	como el usuario interactúa con el subsistema de cambio de contraseña, permite el ingreso de datos de usuario, validando el usuario, validando el password. Posteriormente el sistema pide una contraseña nueva, seguido de la validación de la nueva contraseña y el regreso al login.
<b>ACCIONES DEL SUBSISTEMA</b>	
1. ingreso de datos de usuario	2. validar usuario y password.
3. validar nueva contraseña	4. regreso al login.

Tabla 5.Caso de uso. Subsistema. Cambio contraseña.

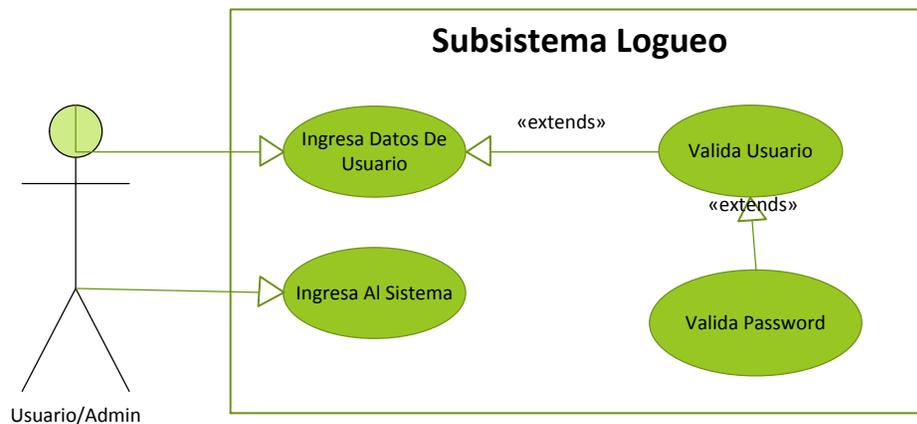


Ilustración 6. Diagrama de caso de uso: Subsistema. Logueo.

<b>SELECCIÓN PRINCIPAL</b>	
Caso de uso	Subsistema. Logueo
Actores	Administrador/usuario
Propósito	Ingreso datos de usuario al sistema, validar usuario y password e ingresar al sistema.

Descripción	Este subsistema nos permite apreciar como el usuario y el administrador realizan el logueo para ingresar al sistema, primero se realiza el ingreso del usuario y la clave, el sistema valida el usuario junto con la clave si es correcto ingresa al sistema.
<b>ACCIONES DEL SUBSISTEMA</b>	
1. ingresar usuario	2. validar usuario y password.
3. ingreso al sistema	

**Tabla 6.Caso de uso. Subsistema. Logueo.**

### 4.3.3 Modelos Objeto

A continuación se describe el diagrama de clases del aplicativo.

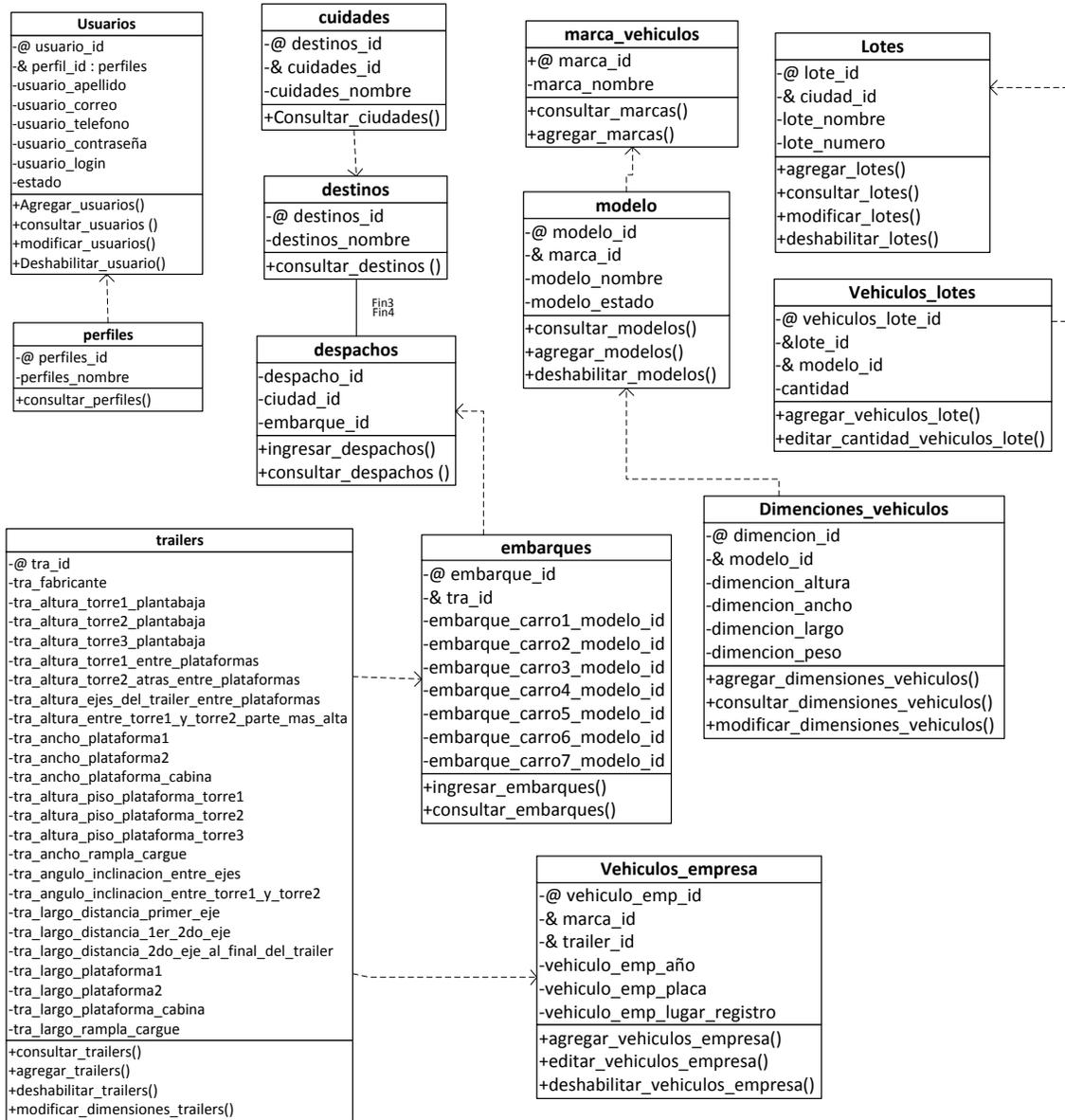


Ilustración 7. Diagrama de Clases.

Usuarios
-@ usuario_id -& perfil_id : perfiles -usuario_apellido -usuario_correo -usuario_telefono -usuario_contraseña -usuario_login -estado
+Agregar_usuarios() +consultar_usuarios () +modificar_usuarios() +Deshabilitar_usuario()

Ilustración 8. Tabla usuarios

**Clase Usuarios:** Esta clase se encarga de almacenar los usuarios que pueden acceder a sistema.

Nombre del campo	Tipo	Descripción
<u>usuario_id</u>	int	Identificador único de la tabla
usuario_nombre	String	Nombre del usuario
usuario_apellido	String	Apellido del usuario
usuario_email	String	Correo electrónico del usuario
usuario_tel	String	Teléfono del usuario
usuario_contraseña	binary	Contraseña de logeo
usuario_login	String	Nombre del usuario
perfil_id	int	Id del perfil del usuario
estado	int	estado del usuario

Tabla 7. Atributos de la clase usuarios.

<b>perfiles</b>
-@ perfiles_id
-perfiles_nombre
+consultar_perfiles()

**Ilustración 9. Tabla Perfiles.**

**Clase perfiles:** esta clase se encarga de guardar el tipo de perfil del usuario.

<b>Nombre del campo</b>	<b>Tipo</b>	<b>Descripción</b>
<u>perfil_id</u>	int	Identificador único de la tabla
perfil_nombre	String	Nombre del perfil

Tabla 8. Atributos de la clase perfiles.

<b>cuidades</b>
-@ destinos_id
-& ciudades_id
-ciudades_nombre
+Consultar_ciudades()

**Ilustración 10. Tabla ciudades.**

**Clase ciudades:** esta tabla se encarga de almacenar las ciudades en las que se realizarán descargues de unidades.

<b>Nombre del campo</b>	<b>Tipo</b>	<b>Descripción</b>
<u>ciudad_id</u>	int(11)	Identificador único de la tabla
destino_id	int(11)	id de la tabla destinos
ciudad_nombre	string	Nombre de la ciudad

Tabla 9. Atributos de la clase ciudades.

<b>destinos</b>
-@ destinos_id
-destinos_nombre
+consultar_destinos ()

Ilustración 11. Tabla destinos.

**Clase destinos:** esta tabla se encarga de almacenar los tipos de destinos en los que se encuentran las ciudades.

<b>Nombre del campo</b>	<b>Tipo</b>	<b>Descripción</b>
<u>destinos_id</u>	int	Identificador único de la tabla
destinos_nombre	string	Nombre del destino

Tabla 10. Atributos de la clase destinos

<b>despachos</b>
-despacho_id
-ciudad_id
-embarque_id
+ingresar_despachos()
+consultar_despachos ()

Ilustración 12. Tabla despachos.

**Clase despachos:** esta tabla se encarga de almacenar todos los despachos realizados por el programador.

<b>Nombre del campo</b>	<b>Tipo</b>	<b>Descripción</b>
<u>despachos_id</u>	int	Identificador único de la tabla
Ciudad_id	Int	Id de la ciudad de destino
Embarque_id	Int	Id del embarque

Tabla 11. Atributos de la clase despachos.

<b>embarques</b>
-@ embarque_id
-& tra_id
-embarque_carro1_modelo_id
-embarque_carro2_modelo_id
-embarque_carro3_modelo_id
-embarque_carro4_modelo_id
-embarque_carro5_modelo_id
-embarque_carro6_modelo_id
-embarque_carro7_modelo_id
+ingresar_embarques()
+consultar_embarques()

Ilustración 13. Tabla embarques.

**Clase de embarques:** esta tabla se encarga de almacenar los vehículos que son despachados hacia un destino.

<b>Nombre del campo</b>	<b>Tipo</b>	<b>Descripción</b>
Embarque_id	int	Identificador único de la tabla
tra_id	Int	Id del trailer
embarque_carro1	Int	Id del modelo del carro 1
embarque_carro2	Int	Id del modelo del carro 2
embarque_carro3	Int	Id del modelo del carro 3
embarque_carro4	Int	Id del modelo del carro 4
embarque_carro5	Int	Id del modelo del carro 5
embarque_carro6	Int	Id del modelo del carro 6
embarque_carro7	Int	Id del modelo del carro 7
embarque_carro8	Int	Id del modelo del carro 8
embarque_carro9	Int	Id del modelo del carro 9

Tabla 12. Atributos de la clase embarques.

<b>trailers</b>
-@ tra_id -tra_fabricante -tra_altura_torre1_plantabaja -tra_altura_torre2_plantabaja -tra_altura_torre3_plantabaja -tra_altura_torre1_entre_plataformas -tra_altura_torre2_atras_entre_plataformas -tra_altura_ejes_del_trailer_entre_plataformas -tra_altura_entre_torre1_y_torre2_parte_mas_alta -tra_ancho_plataforma1 -tra_ancho_plataforma2 -tra_ancho_plataforma_cabina -tra_altura_piso_plataforma_torre1 -tra_altura_piso_plataforma_torre2 -tra_altura_piso_plataforma_torre3 -tra_ancho_rampla_cargue -tra_angulo_inclinacion_entre_ejes -tra_angulo_inclinacion_entre_torre1_y_torre2 -tra_largo_distancia_primer_eje -tra_largo_distancia_1er_2do_eje -tra_largo_distancia_2do_eje_al_final_del_trailer -tra_largo_plataforma1 -tra_largo_plataforma2 -tra_largo_plataforma_cabina -tra_largo_rampla_cargue
+consultar_trailers() +agregar_trailers() +deshabilitar_trailers() +modificar_dimensiones_trailers()

Ilustración 14. Tabla trailers.

**Clase trailers:** esta tabla se encarga de guardar todas las dimensiones que hacen parte de los trailers.

Nombre del campo	Tipo	Descripción
<u>tra_id</u>	int	Identificador único de la tabla
tra_fabricante	String	Nombre de el fabricante
tra_altura_torre1_planta_baja	int	Dimensión altura trailer
tra_altura_torre2_planta_baja	int	Dimensión altura trailer
tra_altura_torre3_planta_baja	int	Dimensión altura trailer
tra_altura_piso_plataforma_torre1	int	Dimensión altura trailer
tra_altura_piso_plataforma_torre2	int	Dimensión altura trailer
tra_altura_piso_plataforma_torre3	int	Dimensión altura trailer
tra_altura_torre1_entre_plataformas	int	Dimensión altura trailer
tra_altura_torre2_atras_entre_plataformas	int	Dimensión altura trailer
tra_altura_ejes_del_trailer_entre_plataformas	int	Dimensión altura trailer
tra_altura_entre_torre1_y_torre2_parta_mas_alta	int	Dimensión altura trailer
tra_ancho_plataforma1	int	Dimensión ancho trailer
tra_ancho_plataforma2	int	Dimensión ancho trailer
tra_ancho_plataforma_cabina	int	Dimensión ancho trailer
tra_acho_rampla_cargue	int	Dimensión ancho trailer
tra_angulo_inclinacion_entre_ejes	int	Dimensión angulo inclinacion
tra_angulo_inclinacion_entre_torre1_y_torre2	int	Dimensión angulo

		inclinación
tra_largo_distancio_primer_eje	int	Dimensión largo trailer
tra_largo_distancia_1er_2do_eje	int	Dimensión largo trailer
tra_largo_distancia_2do_eje_al_final_del_trailer	int	Dimensión largo trailer
tra_largo_plataforma1	int	Dimensión largo trailer
tra_largo_plataforma_2	int	Dimensión largo trailer
tra_largo_plataforma_cabina	int	Dimensión largo trailer
tra_largo_rampla_cargue	int	Dimensión largo trailer

Tabla 13. Atributos de la clase trailers.

<b>Vehiculos_empresa</b>
-@ vehiculo_emp_id
-& marca_id
-& trailer_id
-vehiculo_emp_año
-vehiculo_emp_placa
-vehiculo_emp_lugar_registro
+agregar_vehiculos_empresa()
+editar_vehiculos_empresa()
+deshabilitar_vehiculos_empresa()

Ilustración 15. Tabla vehículos\_empresa.

**Clase vehículos\_empresa:** esta tabla almacena los datos de las tracto mulas de la empresa usadas para trasportar los lotes de vehículos.

Nombre del campo	Tipo	Descripción
vehiculo_emp_id	int	Identificador único de la tabla
<u>tra_id</u>	int	Id del tráiler
<u>marca_id</u>	int	Id de la marca del vehículo
vehiculo_emp_año	string	Año de fabricación del vehículo
vehiculo_emp_placa	string	Placa del vehículo
vehiculo_emp_lugar_registro	string	Lugar de registro del vehículo

Tabla 14 . Atributos de la clase vehiculos\_empresa.

Lotes
-@ lote_id
-& ciudad_id
-lote_nombre
-lote_numero
+agregar_lotes()
+consultar_lotes()
+modificar_lotes()
+deshabilitar_lotes()

Ilustración 16. Tabla lotes.

**Clase lotes:** esta tabla guarda los lotes para posteriormente insertar vehículos para ser asignados a un tráiler y realizar su entrega.

Nombre del campo	Tipo	Descripción
lotes_id	int	Identificador único de la tabla
Ciudad_id	int	Id del tráiler
<u>lote_nombre</u>	string	Nombre del lote
lote_numero	int	Numero de lote

Tabla 15. Atributos de la clase lotes.

<b>Vehiculos_lotes</b>
-@ vehiculos_lote_id -&lote_id -& modelo_id -cantidad
+agregar_vehiculos_lote() +editar_cantidad_vehiculos_lote()

Ilustración 17. Tabla vehículos\_lotes.

**Clase vehículos\_lotes:** esta tabla almacena los vehículos en los lotes ingresados previamente.

<b>Nombre del campo</b>	<b>Tipo</b>	<b>Descripción</b>
Vehículos_lote_id	int	Identificador único de la tabla
lotes_id	int	Id del lote
Modelo_id	int	Id del modelo de vehiculo
cantidad	int	Cantidad de vehiculos

Tabla16. Atributos de la clase vehículos\_lotes.

<b>Dimensiones_vehiculos</b>
-@ dimension_id -& modelo_id -dimension_altura -dimension_ancho -dimension_largo -dimension_peso
+agregar_dimensiones_vehiculos() +consultar_dimensiones_vehiculos() +modificar_dimensiones_vehiculos()

Ilustración 18. Tabla dimension\_vehiculos.

**Clase dimencion\_vehiculos:** esta tabla almacena las dimensiones de los modelos.

Nombre del campo	Tipo	Descripción
dimencion_id	int	Identificador único de la tabla
Modelo_id	int	Id del modelo
Dimencion_altura	int	Altura del vehículo
Dimencion_ancho	int	Ancho vehículo
Dimencion_largo	int	Largo vehículo
Dimencion_peso	Int	Peso del vehículo

Tabla 17. Atributos de la clase dimencion\_vehiculos.

modelo
-@ modelo_id -& marca_id -modelo_nombre -modelo_estado
+consultar_modelos() +agregar_modelos() +deshabilitar_modelos()

Ilustración 19. Tabla modelo.

**Clase modelo:** esta tabla almacena los modelos de vehículos.

Nombre del campo	Tipo	Descripción
Modelo_id	int	Identificador único de la tabla
Marca_id	int	Id del marca
Modelo_nombre	string	Nombre del modelo
Modelo_estado	string	Estado del modelo

Tabla 18. Atributos de la clase modelo.

marca_vehiculos
+@ marca_id
-marca_nombre
+consultar_marcas()
+agregar_marcas()

Ilustración 20. Tabla marca.

**Clase marca:** esta tabla almacena los nombre de las casa matrices.

Nombre del campo	Tipo	Descripción
Marca_id	int	Identificador único de la tabla
Marca_nombre	string	Nombre del modelo

Tabla 19. Atributos de la clase marca.

### 4.3.4 Modelos Dinámicos

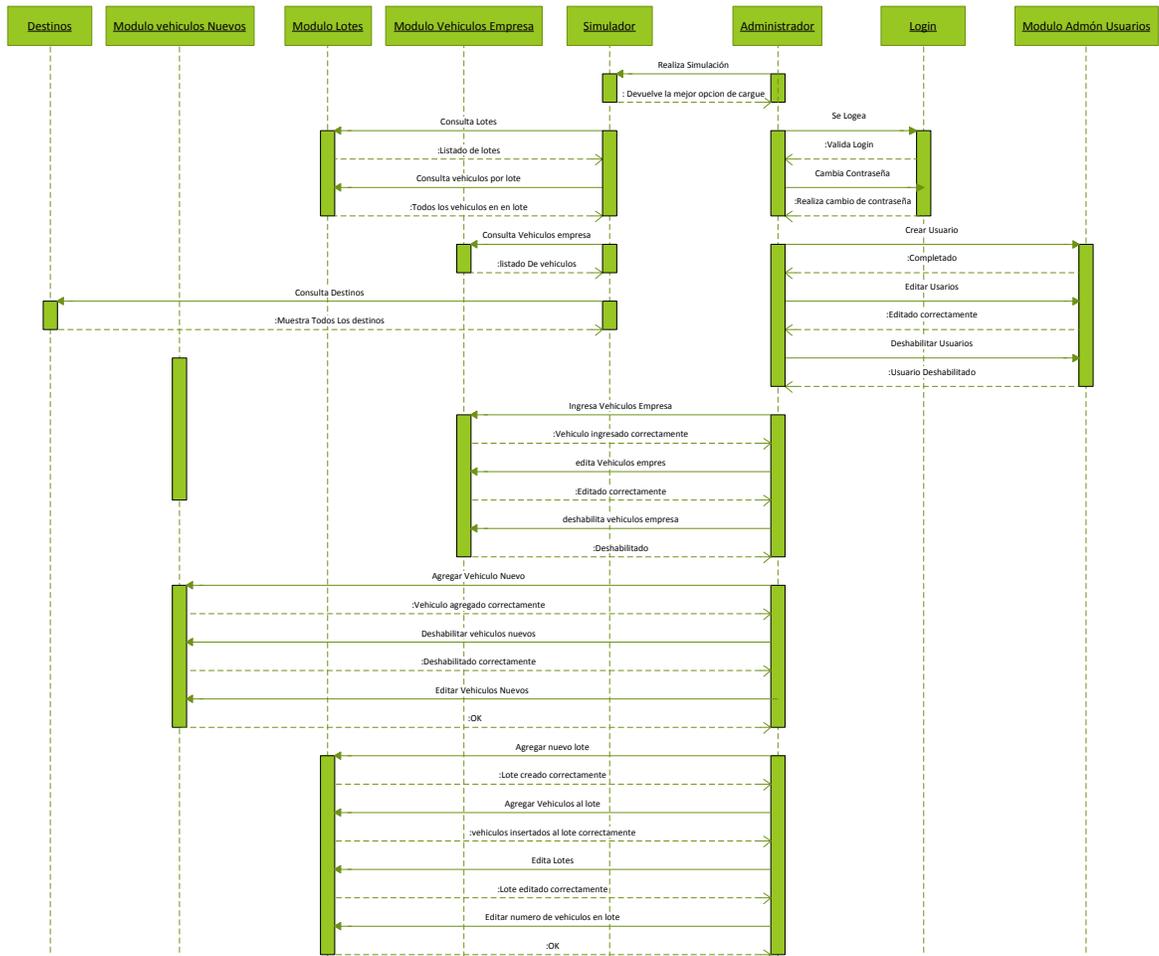


Ilustración 21. Modelo Dinámico: secuencia del sistema.

## 4.4 METODOLOGIA

Este proyecto se realizara bajo la metodología de Programación Extrema la cual es una metodología ligera de desarrollo de software que se basa en la simplicidad, la comunicación y la realimentación del código desarrollado.

Fases del desarrollo:

**I. Planificación:** 1. Historias de Usuario, 2. Plan de Entregas, 3. Velocidad de Proyecto,

4. Iteraciones, 5. Rotaciones, 6. Reuniones.

**II. Diseño:** 1. Metáfora del Sistema, 2. Tarjetas CRC, 3. Soluciones Puntuales,

4. Funcionalidad mínima, 5. Reciclaje.

**III. Desarrollo:** 1. Disponibilidad del cliente, 2. Unidad de Pruebas, 3. Programación por parejas,

4. Integración.

**IV. Pruebas:** 1. Implantación, 2. Pruebas de Aceptación

Ventajas:

- Programación organizada.
- Menor tasa de errores.
- Satisfacción del programador.

Desventajas:

- Es recomendable emplearlo solo en proyectos a corto plazo.
- Altas comisiones en caso de fallar.

El análisis de variables permitirá construir algunos modelos ofrecidos en datos estadísticos mediante gráficos para obtener una mejor visualización de los eventos ocurridos en la compañía. De esta manera favorece la evaluación de diferentes escenarios para finalmente brindar la mejor solución al problema planteado.

**4.4.1 Descripción del proceso.** Actualmente la empresa Transportes Vigía cuenta con una gran cantidad de tráileres o estructuras diseñadas para el transporte de las unidades (autos) a los diferentes concesionarios o puntos de entrega de estas unidades de esta gran cantidad se han extraído para el moldeo de la herramienta cuatro (4) estructuras que son las más utilizadas y las de mayor movimiento para este tipo de operación, los fabricantes de estas estructuras son TRAILERS RODRIEMOLQUES, TRAILERS SANTANDER, ESTRUCTURAS METALMECANO, TRAILERS HIDROAMERICA, la mayoría de esas estructuras cuentan con dimensiones diferentes como ejemplo diremos que los TRAILERS SANTANDER son los de dimensiones menores y sus sistemas de accionamiento son mecánicos y se componen de aros de sujeción y guayas con alma de yute para la elevación y descenso del segundo nivel, sus dimensiones son las más pequeñas en estos tipos de estructuras al contrario de las ESTRUCTURAS METALMECANO que cuentan con sistemas de elevación hidráulicos accionados por bombas de presión hidráulica y sus dimensiones son las de mejor elaboración y fueron elaboradas según los nuevos requerimientos de la empresa Transportes Vigía S.A.S.

De igual forma se tomaron varias líneas de autos para desarrollar el proceso de selección con el cual se trabajará el aplicativo; actualmente la empresa Transportes Vigía cuenta con clientes importantes como General Motors, Compañía Colombiana Automotriz (Mazda) Ssanyong Motor Company, Distribuidora Nacional Nissan, Distoyota estas siendo las más representativas entre un gran número de clientes; las unidades de cada compañía generadora y productora de autos cuenta con unidades de dimensiones y pesos diferentes según su carrocería como se verá más adelante en una tabla diseñada para ver estas diferencias, entre las líneas que manejan las compañías se encuentran los vehículos utilitarios, las panel, los deportivos, autos convencionales, camiones, vehículos de alta gama, familiares entre otros, para el desarrollo del aplicativo utilizaremos como **método** la mayor rotación en el mercado basados en el histórico de unidades movilizadas mes a mes durante el primer semestre del año

en curso junto con sus características primordiales teniendo en cuenta la tabla desarrollada para tal fin.

Los **módulos** a tener en cuenta para el desarrollo del aplicativo son varios y sus variables son igualmente complejas, sin embargo las mencionaremos de acuerdo a los requerimientos de los programadores, las estructuras deben tener la capacidad de contener y transportar como mínimo seis (06) unidades de las de longitudes, ancho, altura y peso diferentes según sean los requerimientos del cliente y las unidades asignadas por el mismo, las distancias entre unidades deben ser mínimo de **15cms** para garantizar que no se contaran con averías durante su trayecto, la altura de las unidades cargadas en las estructuras no puede sobrepasar **los 4.2 mts**, los ángulos de cargue de las rampas de cargue no pueden superar los **14<sup>a</sup>** y los puntos de sujeción deben ser mínimo de tres amarres uno en cada rueda, las unidades de mayor altura serán reducidas (humilladas) disponiendo de reatas y malacates de sujeción enganchadas desde el chasis o la estructura más rígida dependiendo del tipo de unidad. Como podremos ver en la siguiente tabla las unidades cuentan con diferencias en sus dimensiones y pesos para lo cual la estructura debe estar diseñada para contener cualquiera de ellas y en un mínimo de 6 unidades.

**4.4.2 Variables a tener en cuenta.** De acuerdo a la descripción realizada se establece que las variables a tener en cuenta son:

- Tipo de tráiler (esto restringe la longitud total).
- Longitud total del vehículo.
- Distancias mínimas de separación.

Para la solución de este problema no se tendrá en cuenta la altura de los vehículos, sino que se establecerá un factor de corrección de la longitud en la quinta rueda, lugar donde la longitud se ve afectada por la altura entre las dos plataformas ya que se ve reducido ostensiblemente. El número de trailers se debe restringir en el momento de efectuar la asignación.

3.3.3 Modelo de simulación. Entendiendo la simulación como el Modelamiento matemático de un proceso real y que es representado mediante una herramienta de software, definiremos cada uno de los elementos aplicados para este caso.

- **Modelo matemático.** El modelo matemático propuesto es:

$$C_t = \sum L_x + (n - 1) * 0.15$$

Dónde:

$C_t =$  Capacidad del trailer.

$L_x =$  Longitud de la unidad a cargar.

$(n - 1) * 15:$  Funcion que define las distancias optimas de cargue entre unidades.

Todas las anteriores medidas de longitud en unidades del sistema internacional. En primer lugar se define la capacidad del tráiler total, para luego realizar un algoritmo de asignación que busca optimizar el modelo. La optimización se encuentra al ocupar el mayor espacio con unidades con un mínimo sobrante.

El proceso real. Para esta asignación se toman las unidades más transportadas (vehículos) que son cargadas y transportadas de acuerdo al procedimiento establecido. Se toman además 4 tipos de trailers diferentes, que son los más utilizados y se tienen en cuenta las distancias óptimas de separación entre unidades cargadas. Cuando esta distancia de separación no es observada existe un alto riesgo de daños en los vehículos transportados y en consecuencia pérdidas para la empresa que es la situación que se presenta en la actualidad.

Herramienta de software. Para representar el sistema utilizaremos el lenguaje de programación PHP5, el cual cuenta con las herramientas de procesamiento matemático, implementaremos un desarrollo en AJAX que permitirán el desarrollo de una herramienta interactiva para visualizar diferentes escenarios de la solución.

Para representar la solución realizaremos el siguiente proceso:

- Realizar la programación diaria de cada uno de los módulos del software

Se Asignara una macro que organice los destinos de forma discreta (máximo 7 destinos).

- Se determina número de unidades para cada destino.
- Se consolidan destinos para que sea factible optimizar la asignación.
- Se asigna los tráileres disponibles para cada destino.
- Se realiza tantas asignaciones como sea posible apoyados en un formulario realizado en HTML.
- Se determina el punto óptimo de asignación.
- Se podrán ingresar a las bodegas tantos vehículos como guste el programador.

## CONCLUSIONES

- Los objetivos propuestos en el proyecto de grado fueron cumplidos de manera exitosa y presenta el cargue optimo de unidades.
- Aunque el aplicativo es de mucha ayuda para la empresa transportes vigía SAS esta aun en una etapa de mejoramiento debido a que se trata de un prototipo funcional.
- El semillero de investigación Fénix, ha podido probar con gran éxito que la interdisciplinaridad en la universidad deja grandes frutos, ya que por medio de la investigación realizada por parte de los estudiantes de logística y los estudiantes de informática, se ha podido dar solución parcial a un problema que afectaba la empresa transportes vigía SAS, a través de un aplicativo de simulación.

## BIBLIOGRAFIA

[1] Sommerville, Ian. Ingeniería de software. Pearson educación. Séptima de edición.2005.

[2] MENDEZ ALVAREZ. Carlos Eduardo. Metodología Diseño y Desarrollo del Proceso De Investigación. Tercera Edición. Bogota: Editorial **Limusa**, páginas 103 a la 108 y 125 a la 128.

[3] INSTITUTO COLOMBIANO DE NORMAS TÉCNICAS Y CERTIFICACION. Trabajos escritos: presentación y referencias bibliográficas. Sexta actualización. Bogotá: ICONTEC, 2008 110.

[4] GÓMEZ GALLEGO Andrés Mateo. Documento proyecto de grado. Pereira.

<http://es.scribd.com/doc/50414646/Documento-de-Proyecto-de-Grado-SOFTWARE-DE-PERTINENCIA-ACADEMICA>

[5] Reference Manual. Referencia manual base de datos MsqI.

<http://dev.mysql.com/doc/refman/5.0/es/index.html>.

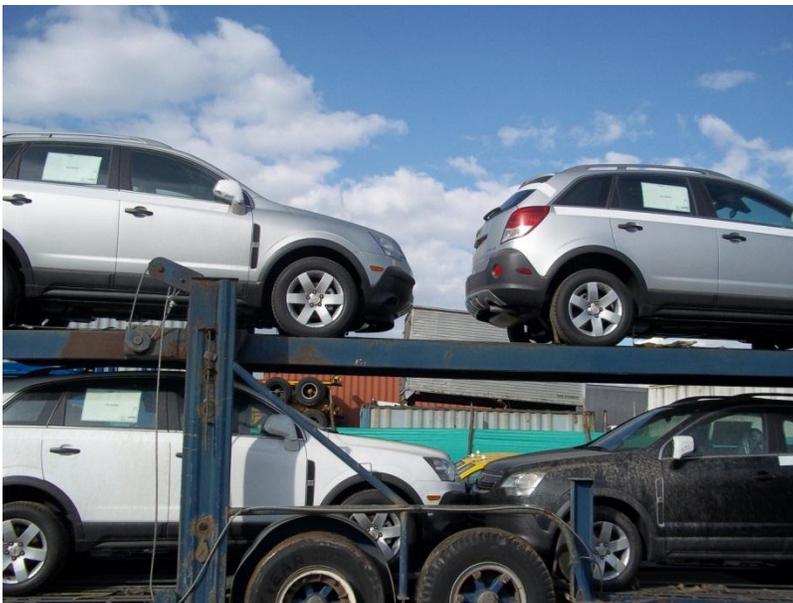
[6] COLOMBIANO, E. (03 de 09 de 2011). BANCO DE CONOCIMIENTO. *AUMENTO DE VENTAS EN VEHICULOS*, pág. 13.

[7] COSS BU, R. (1999). *Simulacion Un enfoque practico*. Balderas, Mexico, DF.: Limusa.

- [8] Frederick S. Hiller, G. j. (1997). *INTRODUCCION A LA INVESTIGACION DE OPERACIONES*. Mexico D.F.: Mc Graw Hill.
- [9] Long, D. (2006). *Logística Internacional: Administración de la cadena de abastecimiento global*. Mexico: Limusa.
- [10] Marquina, R. (04 de 04 de 2002). *Microsoft Visual Basic 6.0*. Recuperado el 01 de 08 de 2011, de Microsoft Visual Basic 6.0: <http://www.hacienda.go.cr/cifh/sidovih/uploads/Curso/Visual%20Basic%206.0-Guia.pdf>
- [11] P., R. S. (14 de 06 de 2008). *BPM CHILE BUSSINES PROCESS MANAGEMENT GROUP*. Recuperado el 31 de 08 de 2011, de BPM CHILE BUSSINES PROCESS MANAGEMENT GROUP.: <http://blog.bpmchile.org/2008/06/optimizacin-de-procesos-parte-i.html>
- [12] Solutions, C. (12 de 06 de 2009). *TUTORES SOLUCIONES COMERCIALES*. Recuperado el 05 de 09 de 2011, de TUTORES SOLUCIONES COMERCIALES: <http://www.tutores.com.co/articulos/que-es-sap.html>
- [13] VANGUARDIA, D. L. (04 de 09 de 2011). LA VENTA DE AUTOS. *ECONOMIA*, pág. 7.

## ANEXOS

Fotografías de unidades cargadas en los equipos sin control de calidad.







UNIDADES EN ZONA QUINTA RUEDA:





EJEMPLO DE UNIDADES HUMILLADAS:





ESPECIFICACIONES TECNICAS TRAILERS			
Fabricante	Variables	Descripción	Metros
Hidroamerica	Altura	Torre 1	2,8
Hidroamerica	Altura	Torre 2	2,8
Hidroamerica	Altura	Torre 3	2,8
Hidroamerica	Altura	Torre 1 Piso a la plataforma	1,2
Hidroamerica	Altura	Torre 2 Piso a la plataforma	0,4
Hidroamerica	Altura	Torre 3 Piso a la plataforma	0,4
Hidroamerica	Altura	Torre 1 entre plataformas	1,55
Hidroamerica	Altura	Torre 2 y Parte Atrás entre plataformas	2,1
Hidroamerica	Altura	Ejes del trailer entre plataformas	1,75
Hidroamerica	Altura	Entre torre 1 y 2 Parte mas alta de la plataforma	2,7
Hidroamerica	Ancho	Plataforma 1	2,18
Hidroamerica	Ancho	Plataforma 2	2,18
Hidroamerica	Ancho	Plataforma Cabina	2,18
Hidroamerica	Ancho	Ramplas de Cargue	0,35
Hidroamerica	Angulo de inclinacion	de Entre torre 1 y 2	3,4

<b>Hidroamerica</b>	Angulo de inclinacion	de Entre los ejes	0,15
<b>Hidroamerica</b>	Largo	Distancia al primer eje	Plataforma 1 2,7
<b>Hidroamerica</b>	Largo	Distancia 1ro y 2do Eje	Plataforma 1 1,35
<b>Hidroamerica</b>	Largo	Distancia 2do eje al fin del trailer	Plataforma 1 6,85
<b>Hidroamerica</b>	Largo	Plataforma 1	14,45
<b>Hidroamerica</b>	Largo	Plataforma 2	14,45
<b>Hidroamerica</b>	Largo	Plataforma Cabina	3,8
<b>Hidroamerica</b>	Largo	Ramplas de Cargue	1,5
<b>Metalmeccano</b>	Altura	Torre 1	2,9
<b>Metalmeccano</b>	Altura	Torre 2	2,9
<b>Metalmeccano</b>	Altura	Torre 3	2,9
<b>Metalmeccano</b>	Altura	Torre 1 Piso a la plataforma	1,2
<b>Metalmeccano</b>	Altura	Torre 2 Piso a la plataforma	0,4
<b>Metalmeccano</b>	Altura	Torre 3 Piso a la plataforma	0,4
<b>Metalmeccano</b>	Altura	Torre 1 entre plataformas	1,65
<b>Metalmeccano</b>	Altura	Torre 2 y Parte Atrás entre plataformas	2,2
<b>Metalmeccano</b>	Altura	Ejes del trailer entre plataformas	1,85

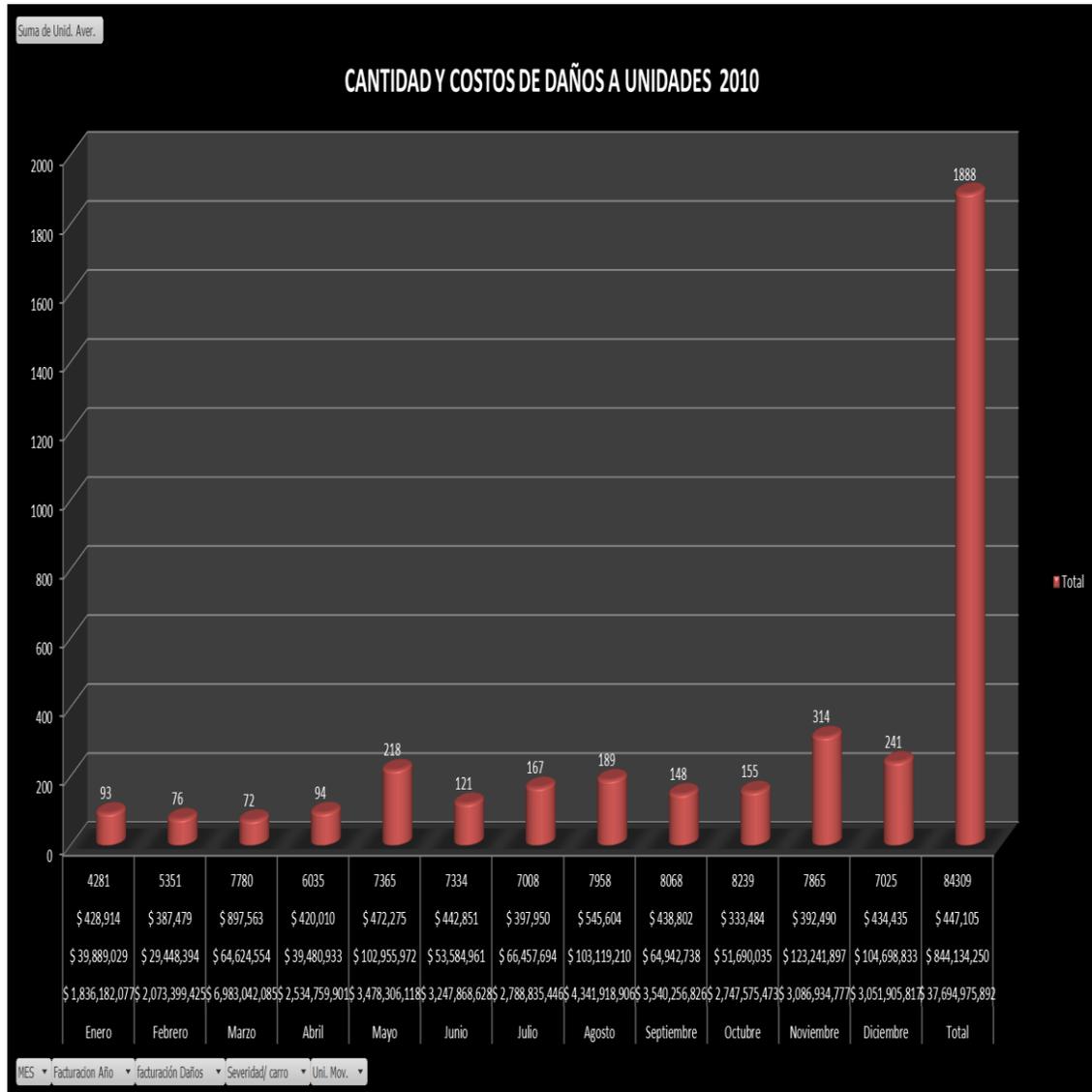
<b>Metalmecano</b>	Altura	Entre torre 1 y 2	Parte mas alta de la plataforma	2,8
<b>Metalmecano</b>	Ancho		Plataforma 1	2,24
<b>Metalmecano</b>	Ancho		Plataforma 2	2,24
<b>Metalmecano</b>	Ancho		Plataforma Cabina	2,24
<b>Metalmecano</b>	Ancho		Ramplas de Cargue	0,35
<b>Metalmecano</b>	Angulo de inclinacion	de	Entre torre 1 y 2	3,4
<b>Metalmecano</b>	Angulo de inclinacion	de	Entre los ejes	0,15
<b>Metalmecano</b>	Largo		Distancia al primer eje Plataforma 1	3,1
<b>Metalmecano</b>	Largo		Distancia 1ro y 2do Eje Plataforma 1	1,35
<b>Metalmecano</b>	Largo		Distancia 2do eje al fin del trailer Plataforma 1	6,85
<b>Metalmecano</b>	Largo		Plataforma 1	14,85
<b>Metalmecano</b>	Largo		Plataforma 2	14,85
<b>Metalmecano</b>	Largo		Plataforma Cabina	3,8
<b>Metalmecano</b>	Largo		Ramplas de Cargue	1,5
<b>Pachon</b>	Altura		Torre 1	2,9
<b>Pachon</b>	Altura		Torre 2	2,9

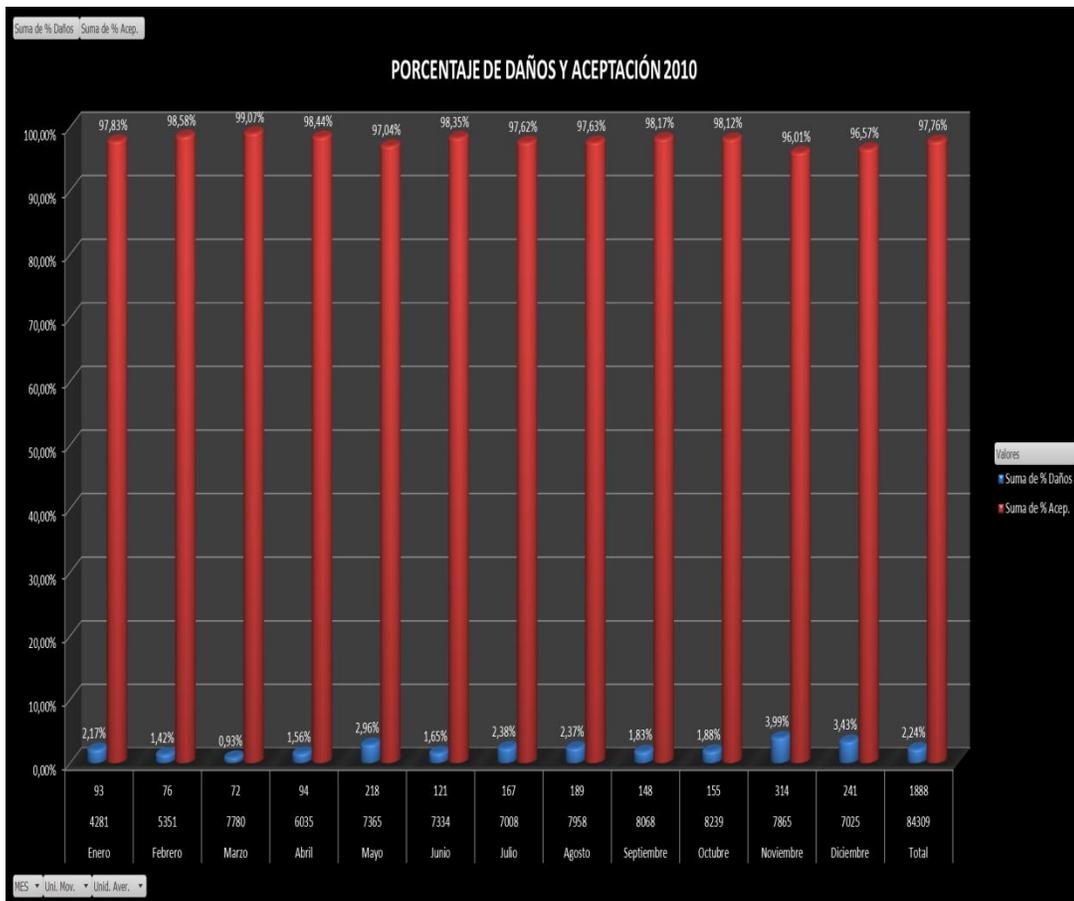
Pachon	Altura	Torre 3	2,9
Pachon	Altura	Torre 1 Piso a la plataforma	1,2
Pachon	Altura	Torre 2 Piso a la plataforma	0,4
Pachon	Altura	Torre 3 Piso a la plataforma	0,4
Pachon	Altura	Torre 1 entre plataformas	1,65
Pachon	Altura	Torre 2 y Parte Atrás entre plataformas	2,2
Pachon	Altura	Ejes del trailer entre plataformas	1,85
Pachon	Altura	Entre torre 1 y 2 Parte mas alta de la plataforma	2,8
Pachon	Ancho	Plataforma 1	2,18
Pachon	Ancho	Plataforma 2	1,95
Pachon	Ancho	Plataforma Cabina	2,18
Pachon	Ancho	Ramplas de Cargue	0,35
Pachon	Angulo de inclinacion	de Entre torre 1 y 2	3,4
Pachon	Angulo de inclinacion	de Entre los ejes	0,15
Pachon	Largo	Distancia al primer eje Plataforma 1	3,25
Pachon	Largo	Distancia 1ro y 2do Eje Plataforma 1	1,35
Pachon	Largo	Distancia 2do eje al fin del trailer	6,85

		Plataforma 1	
<b>Pachon</b>	Largo	Plataforma 1	15
<b>Pachon</b>	Largo	Plataforma 2	15
<b>Pachon</b>	Largo	Plataforma Cabina	3,8
<b>Pachon</b>	Largo	Ramplas de Cargue	1,5
<b>Santander</b>	Altura	Torre 1	3,2
<b>Santander</b>	Altura	Torre 2	3,2
<b>Santander</b>	Altura	Torre 3	3,2
<b>Santander</b>	Altura	Torre 1 Piso a la plataforma	1,2
<b>Santander</b>	Altura	Torre 2 Piso a la plataforma	0,4
<b>Santander</b>	Altura	Torre 3 Piso a la plataforma	0,4
<b>Santander</b>	Altura	Torre 1 entre plataformas	1,85
<b>Santander</b>	Altura	Torre 2 y Parte Atrás entre plataformas	2,4
<b>Santander</b>	Altura	Ejes del trailer entre plataformas	2,05
<b>Santander</b>	Altura	Entre torre 1 y 2 Parte mas alta de la plataforma	2,8
<b>Santander</b>	Ancho	Plataforma 1	2,3
<b>Santander</b>	Ancho	Plataforma 2	2,3
<b>Santander</b>	Ancho	Plataforma Cabina	2,3

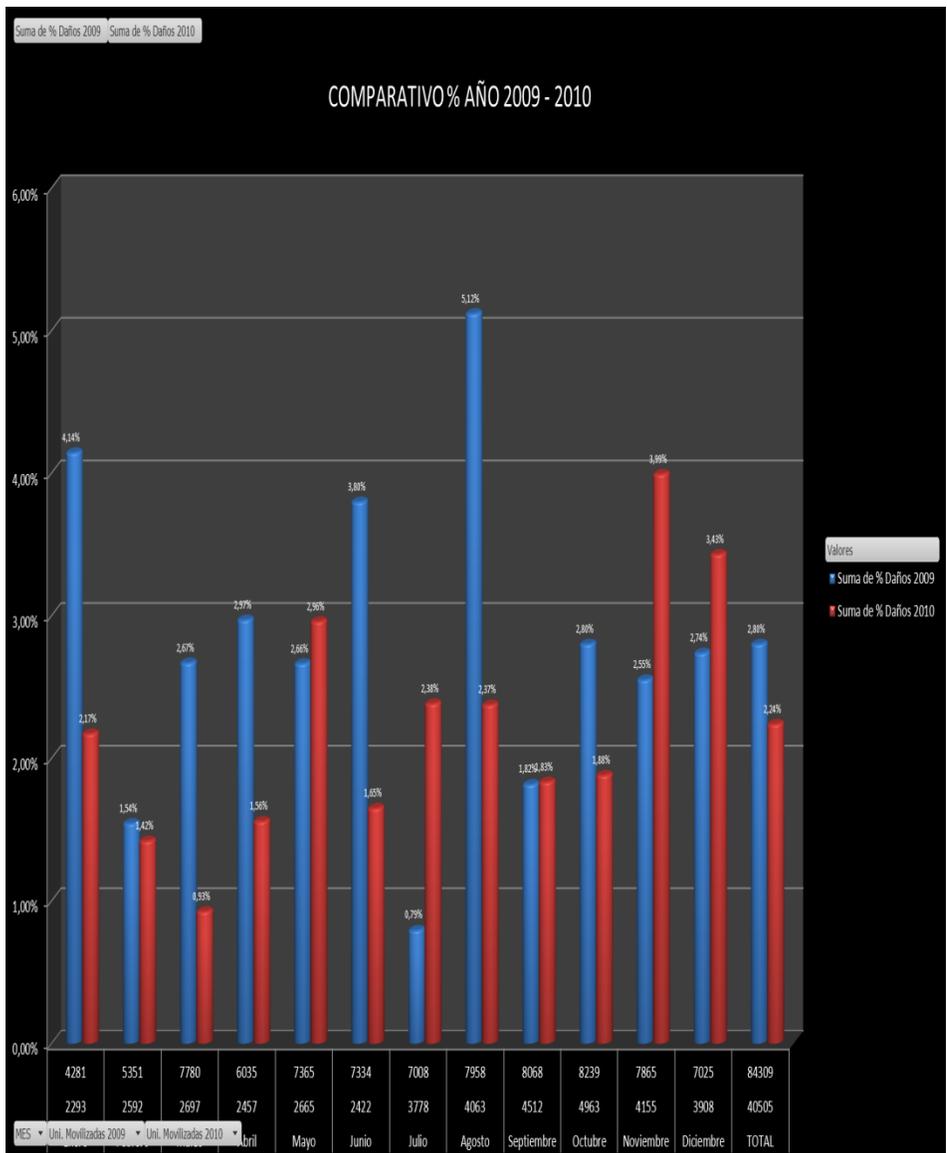
Santander	Ancho	Ramplas de Cargue	0,35
Santander	Angulo de inclinacion	de Entre torre 1 y 2	3,4
Santander	Angulo de inclinacion	de Entre los ejes	0,15
Santander	Largo	Distancia al primer eje Plataforma 1	2,55
Santander	Largo	Distancia 1ro y 2do Eje Plataforma 1	1,35
Santander	Largo	Distancia 2do eje al fin del trailer Plataforma 1	6,85
Santander	Largo	Plataforma 1	14,3
Santander	Largo	Plataforma 2	14,3
Santander	Largo	Plataforma Cabina	3,8
Santander	Largo	Ramplas de Cargue	1,5

## GRAFICAS INDICADORES DE CALIDAD 2009 Y 2010

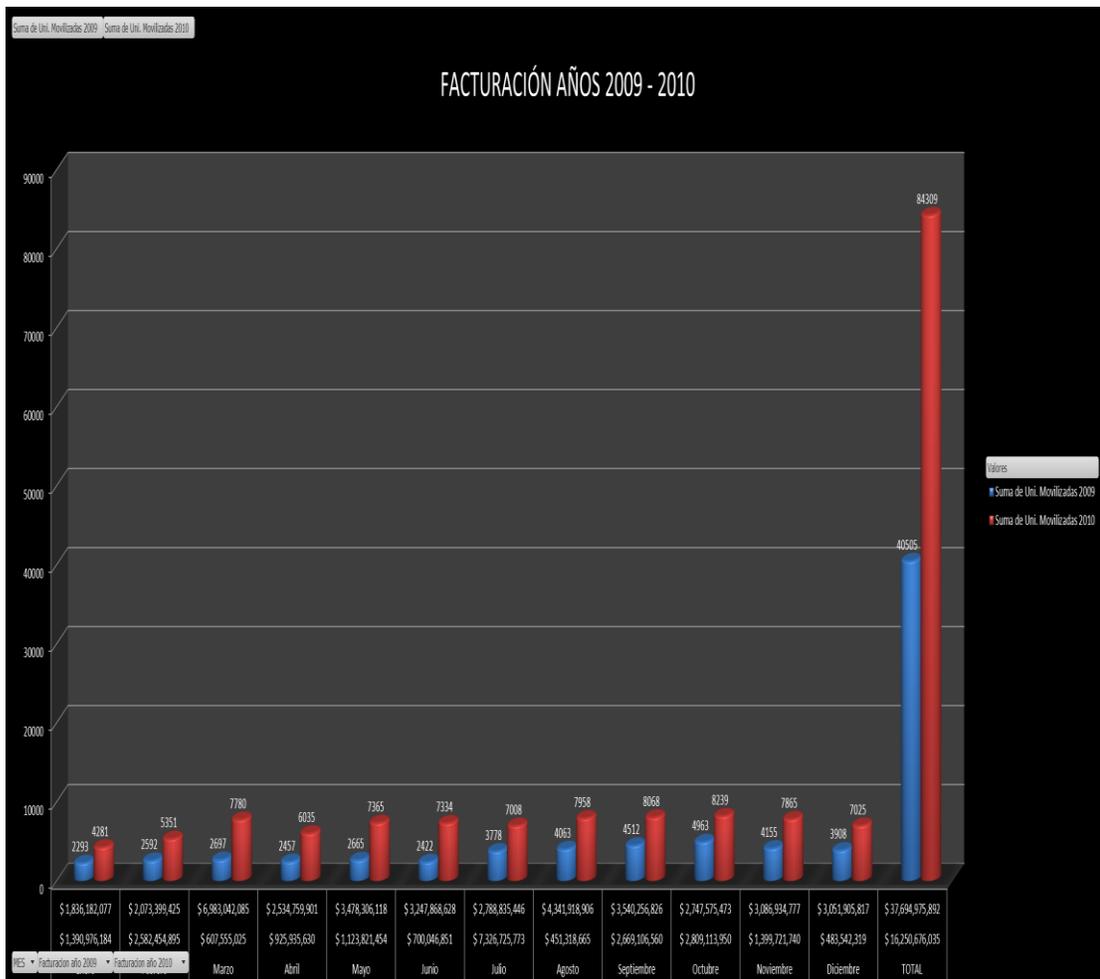




Este grafico muestra la relación entre las unidades movilizadas mes a mes y porcentaje de aceptación.

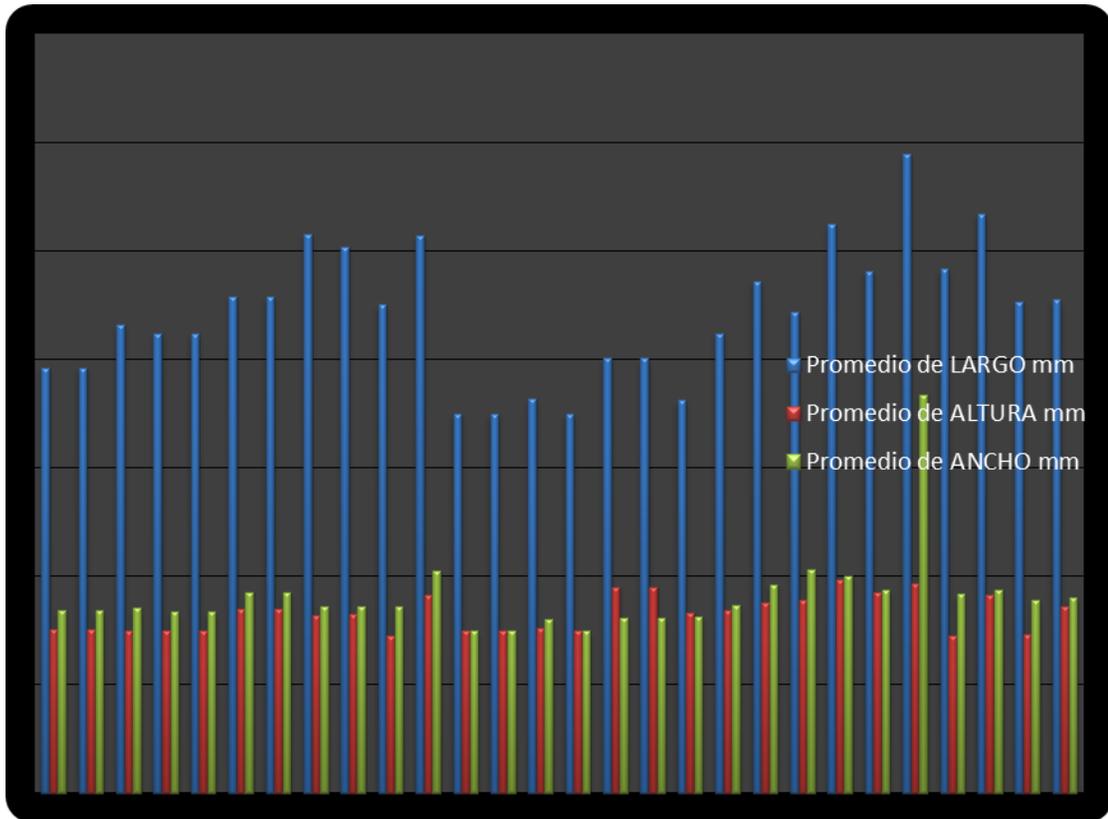


Este grafico muestra la relación de unidades movilizadas entre los años 2009 y 2010 al igual que la aceptación de daños en los años en relación.



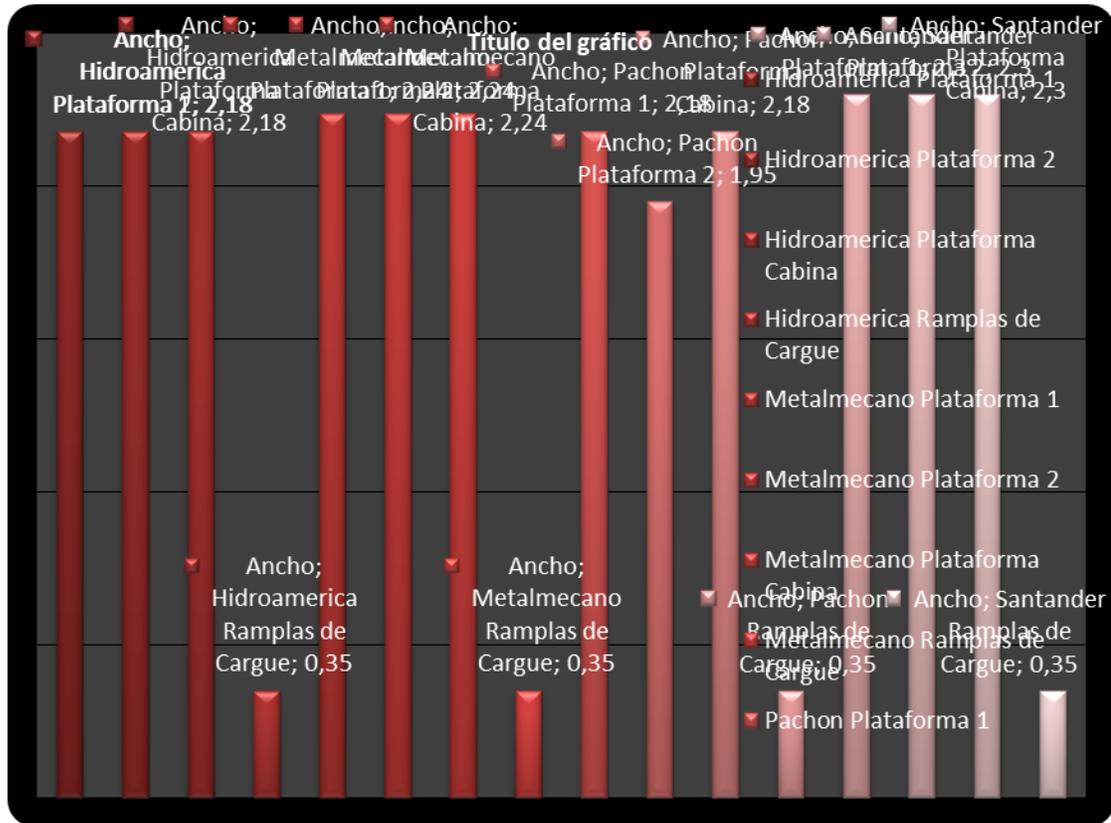
Este grafico muestra la facturación realizada mes a mes durante los años 2009 y 2010 al igual que las unidades movilizadas.

## 1.1 GRAFICA COMPARATIVA UNIDADES.

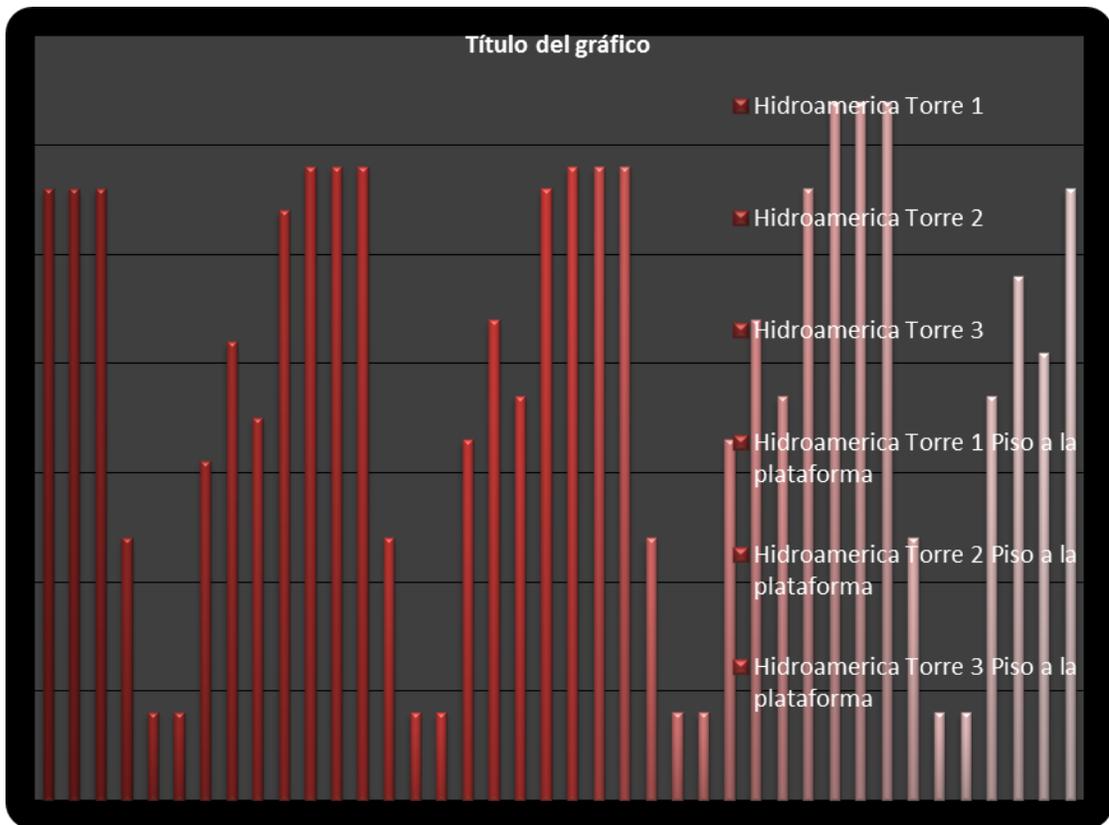


Esta gráfica muestra las diferencias entre unidades de diferentes casas matriz en relación con sus pesos y dimensiones.

## GRAFICAS COMPARATIVA DE TRAILERS.



Este gráfico muestra las diferencias entre las estructuras seleccionadas.



**GRAFICO DEL TRÁILER.**

