

UNIVERSIDAD DE HOLGUÍN “Oscar Lucero Moya”
FACULTAD DE INFORMÁTICA-MATEMÁTICA



SISTEMA PARA LA GESTIÓN DE LA INFORMACIÓN DEL TRANSPORTE EN ETECSA
Trabajo para optar por el título de Ingeniero Informático

Autor:

Walter Jesús González Pérez

Tutores:

Ing. Oscar Gabriel Reyes Pupo

Lic. Reynier Pérez Salas

Holguín
Junio de 2009

Agradecimientos

A nuestro Dios, a quien le debemos todo.

A mis padres, por apoyarme desde mi infancia hasta este preciso momento, sin ellos imposible haber llegado.

A mis tutores Oscar y Reynier, por su dedicación.

Al departamento de TI y Facturación de ETECSA, gracias por su atención y preocupación.

A Ernest, por su ayuda incondicional..

A mis amigos.

Dedicatoria

A mi Dios por estar conmigo en todo momento, aunque a muchos le parezca incierto.

A mis padres queridos, seres especiales que me enseñaron a esforzarme y amar en la vida, y a quienes también les debo mucho de lo que soy.

A mis amigos.

Resumen

Las Tecnologías de la Información y las Comunicaciones se han convertido en la principal causa de un conjunto de transformaciones económicas y sociales que producen una transición en la base material de la sociedad, en la mayoría de sus expresiones, para mejorar las condiciones de vida.

La Empresa de Telecomunicaciones de Cuba S.A., cuyo objetivo es el de proporcionar a toda la población cubana servicios que garanticen la satisfacción de sus necesidades en materia de telecomunicaciones, respaldando los planes de desarrollo social y económico que lleva a cabo el país, para garantizar una alta calidad en la prestación de sus servicios, no es indiferente a las amplias posibilidades que brindan estas tecnologías de la información. Por ello hacen un buen uso de las mismas en todas las esferas en que se puedan implementar.

Por otra parte la empresa se encuentra en medio de un proceso de migración hacia tecnologías de código abierto, como estrategia para lograr una mayor fiabilidad, estabilidad y seguridad de las redes, los sistemas, tecnologías de la información y de las telecomunicaciones, así como la soberanía tecnológica como factores de respaldo de la defensa nacional.

Surge la concepción del sistema SAGTRANS a petición de los directivos del Departamento de Tecnologías de la Información, encargado del soporte informático de la empresa, ante la necesidad de contar con una herramienta informática desarrollada con tecnologías que se encuentren dentro de la categoría de código abierto o software libre, que gestione de una manera segura y eficiente la información de la actividad de transporte en la empresa.

La presente investigación propone una solución a estas necesidades, a través de la creación de un sistema informático que gestione eficientemente el flujo de información de la actividad de transporte, que brinde a sus datos confiabilidad, integridad y rapidez de cálculo.

Abstract

The information and the communications technologies have become the main cause of a group of economic and social transformations that produce a transition in our society's material base, in most of its expressions, to improve the life conditions.

The Company of Telecommunications of Cuba S.A. whose objective is to provide to the whole Cuban population with services that guarantee the satisfaction of its needs as regards telecommunications, supporting the plans of social and economic development that the country carries out, to guarantee a high quality in its services, is not indifferent to the wide possibilities that offer these information technologies. Hence they make a good use of these technologies in all the spheres in that they can be implemented.

On the other hand the enterprise is amid a migration process toward open code technologies, as strategy to achieve a bigger reliability, stability and security of the nets, of systems, technologies of the information and of the telecommunications, as well as the technological sovereignty as factors of backup of the national defense.

It arises SAGTRANS system at the request of the directive of the Department of the Information Technologies, in charge of the computer support of the company, in the face of the necessity of having a computer tool developed with technologies that are inside the category of open code or free software, that manage in a secure and efficient way the information of the transport activity in the company.

The present investigation proposes a solution to these necessities, through the creation of a computer system that manage the information flow of the transport activity that toasts to its data dependability, integrity and calculation speed efficiently.

Índice

Introducción	8
Capítulo 1: Fundamentos Teóricos	16
1.1 Objeto de Estudio	16
1.1.1 La Empresa de Telecomunicaciones de Cuba ETECSA	16
1.1.2 Sistemas de gestión basados en procesos	17
1.1.3 El Proceso de Servicios Generales de ETECSA	18
1.1.4 La Prestación de los Servicios de Transporte en ETECSA	18
1.1.5 Reglas a considerar que debe cumplir la actividad de transporte	20
1.2 Contexto teórico del desarrollo de aplicaciones Web	21
1.2.1 ¿Por qué Aplicación Web?	21
1.2.2 ¿Qué es una Aplicación Web?	22
1.2.3 Arquitectura Cliente-Servidor	23
1.2.4 Software libre y licencias	24
1.2.5 El servidor HTTP Apache	25
1.3 Tecnologías para el desarrollo de Aplicaciones Web	26
1.3.1 El lenguaje de programación Web PHP	26
1.3.2 Los frameworks	27
1.3.3 Framework Symfony	28
1.3.4 Sistema Gestor de Base de Datos	30
1.4 Metodologías para el desarrollo de Software	32
1.4.1 Proceso unificado de desarrollo (RUP)	32
1.4.2 Metodologías ágiles	33
1.4.3 Programación extrema (XP)	34
1.4.4 Metodologías Crystal	34
1.4.5 Metodología Scrum	34
1.4.6 Lenguaje Unificado de Modelación (UML)	35
1.5 Conclusiones	37
Capítulo 2: Diseño y construcción de la propuesta.	38

2.1 Flujo de trabajo de Scrum	38
2.1.1 Construcción de la Pila de Producto	39
2.1.2 Planificación de Sprint	40
2.1.3 Gráfico Burn-Down	41
2.2 Especificación detallada del comportamiento del sistema	43
2.2.1 Requerimientos Funcionales	43
2.2.2 Requerimientos no Funcionales	46
2.2.3 Actores del sistema	48
2.2.4 Diagramas de Casos de uso del sistema	50
2.2.6 Paquetes y sus relaciones	53
2.2.7 Diagrama de Clases del Diseño de los casos de uso fundamentales	53
2.2.8 Diagrama de Clases Persistentes del Diseño	54
2.2.9 Diagrama de Despliegue	55
2.2.10 Valoración de sostenibilidad de la propuesta	55
2.2.3 Valoración de los resultados obtenidos	60
2.4 Conclusiones	62
Conclusiones Generales	64
Recomendaciones	66
Referencia Bibliográfica	67
Bibliografía	70
Glosario de Términos	72
Anexos	74
Anexo No 1. Mapa General de Procesos de ETECSA	74
Anexo No 2. Pila de Producto	75
Anexo No 3. Planificación de Sprint.	80
Anexo No 4. Diagramas de Clases del Diseño	83
Anexo No 5. COCOMO 2	88
Anexo No 6. Encuesta de opinión de los usuarios del sistema SAGTRANS, fase de Prueba.	92
Anexo No 7. Procesamiento de la encuesta de opinión de los usuarios aplicando el método Delphy.	94

Introducción

El vertiginoso y revolucionario avance de las nuevas tecnologías de la información ha creado un nuevo tipo de sociedad, la sociedad en red o de redes, en la que muchos de sus aspectos más determinantes del bienestar humano se desenvuelven a escala global o planetaria. Las redes de comunicación e información, y las distintas posibilidades tecnológicas que ellas conllevan, permiten un crecimiento estable para mejorar la competitividad, ampliar las posibilidades de trabajo, y en definitiva mejorar la calidad de vida.

Desde hace más de veinte años se discute en todo el mundo el libre empleo de programas de computación denominados, por esa razón, "software libre". Hasta hace poco tiempo era imposible usar una computadora moderna sin la instalación de un sistema operativo propietario, provisto por el fabricante mediante licencias con altas restricciones para su uso. Nadie poseía permiso para compartir programas (software) libremente con otros usuarios de computadora y difícilmente alguien podría modificar los programas para adecuarlos a sus necesidades.

“El proyecto GNU, de la Fundación para el Software Libre (Free Software

Foundation), creada en 1984, constituye el inicio del Movimiento de Software Libre, y fue instituido para cambiar esa situación. Este proyecto involucró a centenas de programadores en diferentes partes del mundo, en un significativo esfuerzo cooperativo”. [1].

“La aparición del software libre es bien complicada y atrevida, pretende cambiar un modelo tradicional y se proclama como una alternativa de calidad superior provocando la ira de un sector empresarial muy consolidado económicamente y con un gran poder político”. [3].

En el sector empresarial, grandes empresas de servicios informáticos como IBM, Sun Microsystems, Apple, Intel y AOL-Time Warner apoyan esta iniciativa, buscando en primer lugar contrarrestar la competencia de empresas productoras de software propietario, encabezadas por Microsoft, en un mercado que tiene tendencias monopólicas. Otras empresas, usuarias de tecnologías de la información han adoptado soluciones de software libre movidos fundamentalmente por una razón económica, al no tener que contraer la obligación de pago de licencias, además de contar con la ventaja de poseer parte de esos programas abiertos distribuidos en forma gratuita. Esto permite a estas empresas tener libertad para crear soluciones propias, seguridad y estabilidad funcional de sus sistemas de información.

Con respecto a Cuba, la situación se torna más compleja. Las grandes empresas desarrolladoras de software propietario tienen su domicilio legal en Estados Unidos de América, por lo que se atienen a la legislación sobre exportaciones de su país, y si a esto se le suma que en las licencias de sus programas exigen a los licenciatarios cumplir esas mismas regulaciones, por lo que el país se encuentra ante la imposibilidad de adquirir un programa de esta clase.

Por lo que la oficina de Informatización de la Sociedad está haciendo esfuerzos, y trazando estrategias para la futura transición de los servicios, aplicaciones y sistemas operativos hacia plataformas amparadas por licencias de

libre distribución. [2]

Para Cuba es también un modelo muy ventajoso cuando se analizan las características de este país, que no tiene industrias como para dedicarse a la fabricación de computadoras pero sí cuenta con una infraestructura educacional que le permite formar programadores y técnicos que mediante la creación de software y la prestación de servicios informáticos pueden generar cuantiosos dividendos al país. Es entonces el software libre la mejor alternativa.

Hoy en día varias entidades en Cuba como La Empresa de Telecomunicaciones de Cuba S.A. (ETECSA), el Ministerio de la Informática y las Comunicaciones (MIC), la Empresa de Perforación y Extracción de Petróleo del Centro (EPEP-Centro), la Empresa de Perforación y Reparación Capital de Pozos de Petróleo y Gas (EMPERCAP), la Universidad de La Habana, la Universidad de Ciencias Informáticas (UCI), entre otras, se han adherido a los principios de la Fundación de Software Libre (FSF), con el objetivo de lograr una mayor independencia del software propietario.

ETECSA se encuentra inmersa en un proceso de migración hacia sistemas de código abierto como estrategia para lograr en el futuro mayor soberanía tecnológica y seguridad en las aplicaciones que se emplean.

Actualmente, los especialistas de transporte de la Dirección Territorial de ETECSA en Holguín (DTHO), utilizan un sistema implantado nacionalmente el cual les permite mantener cierto control de la información de la actividad de transporte en la Empresa, pero que no cumple todas las expectativas de los especialistas presentando las siguientes dificultades:

- ✓ El sistema depende de software propietario (fue implementado utilizando la tecnología .NET de Microsoft), lo cual se aleja de la actual política de la Empresa.
- ✓ No existe una forma de almacenar y gestionar los accidentes del transporte, garantizando que se almacenen todos los datos que resulten relevantes.

- ✓ No almacena los datos que permiten llevar un control de las piezas de los transportes ni los datos del parqueo.
- ✓ La información que representa el nivel de explotación del transporte no la almacena el sistema, realizándose en la actualidad de forma manual y está contenida en hojas de cálculo de Microsoft Excel.
- ✓ No existe total correspondencia entre las informaciones de los distintos sistemas que se emplean en la empresa, por ejemplo, la información de los trabajadores no siempre se corresponde con la generada por el Departamento de Capital Humano.
- ✓ No se garantiza de manera técnica la complejidad de la contraseña definida en la política de ETECSA.

Las dificultades antes mencionadas atentan contra la eficiencia de la gestión de la información relacionada con la actividad de transporte, ya que el sistema no se complementa con el principio de lograr una mayor soberanía tecnológica, definido en la resolución No. 148 del 2007 del Ministerio de la Informática y las Comunicaciones, como factor de respaldo de la Defensa Nacional, por otro lado los datos que no están contemplados y se controlan de forma manual, provocan que el proceso sea más lento y complicado para los especialistas de transporte, también el hecho de que no exista total correspondencia de los datos con los otros sistemas genera ruido en la información, y por último la falta de complejidad en las contraseñas puede constituir la base de una deficiencia de seguridad.

Por lo antes mencionado, en la Dirección Territorial de ETECSA en Holguín (DTHO) estaba siendo desarrollado un proyecto que tiene como propósito final la creación de un sistema informático que mejore la gestión de la información de la actividad de transporte. El proyecto fue codificado para su desarrollo con el acrónimo SAGTRANS, y se propuso como sustituto para el sistema actualmente en explotación.

El desarrollo del proyecto SAGTRANS se encontraba en el estado que a

continuación se describe:

- ✓ Se tienen identificados los requisitos funcionales, de seguridad, de implementación y explotación, y otros que debe cumplir el SAGTRANS.
- ✓ Están identificados los actores primarios que interactuarán con el Sistema final.
- ✓ Se tiene diseñado el modelo de clases de una base de datos necesario para contener la información con la que operará el SAGTRANS. También se tiene una implementación de este diseño utilizando el gestor de bases de datos MySQL.

El proyecto se encuentra en un estado primario, no siendo aún funcional para los usuarios.

A partir de esta problemática se identificó el siguiente **problema científico**: ¿Cómo favorecer la gestión de la información referente a la actividad de transporte en la Dirección Territorial de ETECSA en Holguín, cumpliendo con las políticas de la empresa para la prestación de los servicios de transporte?

El problema anteriormente definido se enmarca en el **objeto de estudio**: Proceso de Servicios Generales de ETECSA relacionado con la información de la actividad de transporte.

Por lo que se plantea el siguiente **objetivo**: Desarrollar un sistema informático que permita la gestión de la información de la actividad de transporte en la Dirección Territorial de ETECSA en Holguín, garantizando una mayor integridad en el control interno de la actividad de transporte e independencia del marco tecnológico elegido.

Este objetivo delimita el **campo de acción**: Informatización de la información correspondiente a la actividad de transporte contemplada dentro del Proceso de Servicios Generales de ETECSA.

Para guiar la investigación de este trabajo se planteó la siguiente **hipótesis**: El desarrollo de un sistema informático, que se caracterice por la independencia

del marco tecnológico elegido, ser eficiente, rápido, que garantice la confiabilidad, usabilidad, disponibilidad y seguridad de los datos, favorecerá el proceso de gestión de la información de la actividad de transporte en la Dirección Territorial de ETECSA en Holguín.

Para la realización del sistema, se utilizó el lenguaje de programación Web PHP, haciendo uso del framework¹ Symfony, para la creación de páginas dinámicas, y como sistema gestor de base de datos se empleó el PostgreSQL, además se escogió la metodología ágil SCRUM, utilizando esquemas del Lenguaje Unificado de Modelación (UML, por sus siglas en inglés). La Herramienta para Asistir por Computadora a la Ingeniería de Software, Visual Paradigm sirvió de apoyo en la confección de los documentos y gráficos requeridos por la metodología SCRUM.

Para cumplimentar el objetivo de la investigación se realizaron las siguientes **tareas**:

1. Determinar los fundamentos teóricos necesarios sobre la gestión de la información de la actividad de transporte en ETECSA.
2. Diagnóstico de la situación existente de la gestión de la información de la actividad de transporte en la empresa de ETECSA.
3. Realizar la fundamentación teórica de las nuevas tendencias basada en el uso de software libre para el desarrollo de sistemas informáticos.
4. Realizar la valoración de sostenibilidad del sistema informático.
5. Desarrollar el sistema informático.
6. Probar el sistema informático.
7. Evaluación de la satisfacción de los usuarios con respecto al sistema informático.

¹ framework: en el desarrollo de software, es una estructura de soporte definida mediante la cual otro proyecto de software puede ser organizado y desarrollado. [16].

Para desempeñar estas tareas se emplearon **métodos de investigación científica** teóricos y empíricos.

Los **métodos empíricos** que se usaron son:

La observación se empleó para tener una mayor precisión y seguridad en las decisiones a tomar y los resultados obtenidos acerca del comportamiento del objeto de investigación, tal y como este se comporta en la realidad, además de obtener la información directa e inmediata sobre el proceso de desarrollo que está siendo investigado.

La entrevista ofreció la posibilidad de conocer cómo se realiza la gestión de la información referente a la actividad de transporte en la DTHO, cómo la procesan y lo que necesitan.

La revisión de documentos permitió conocer los datos que son de interés para la prestación de los servicios de transporte, para así poderlos procesar de una forma correcta.

Encuestas: Este método se utilizó fundamentalmente para evaluar la satisfacción de los usuarios con respecto al sistema. Para el tratamiento de las encuestas se utilizó el método Delphi, este permitió el procesamiento estadístico de las encuestas realizadas a los usuarios del sistema con el objetivo de buscar el consenso en las opiniones de los encuestados.

Entre los **métodos teóricos** empleados se encuentran:

El análisis y la síntesis: El análisis posibilitó descomponer el procesamiento de la información en sus partes y cualidades y las múltiples relaciones entre ellas, mientras que la síntesis permitió la unidad entre las partes previamente analizadas y la simplificación de la información a procesar.

El hipotético deductivo permitió plantear una hipótesis que guió el desarrollo de la presente investigación cuya veracidad quedó demostrada al cumplirse el objetivo trazado.

Método de modelación: este método se utilizó durante la etapa de elaboración del sistema al hacer uso del UML como lenguaje de modelado, guiado por el proceso ágil Scrum, desde su análisis, hasta su implantación y

validación; mediante este proceso se crearon modelos con la perspectiva de investigar la realidad logrando que describieran todas las perspectivas posibles del proceso de desarrollo en sentido general.

El documento está estructurado en introducción, dos capítulos, conclusiones, recomendaciones, referencias bibliográficas, bibliografía, glosario de términos y anexos.

Capítulo 1. Fundamentos teóricos. Está orientado a mostrar los principales fundamentos que sostienen la investigación, además se describen las principales herramientas que se utilizan para la elaboración de la propuesta y la metodología de desarrollo de software utilizada en la investigación.

Capítulo 2. Diseño y construcción de la propuesta. Muestra los flujos de trabajo principales de la ingeniería de software realizada al sistema propuesto, incluyendo un estudio de factibilidad y una valoración de sostenibilidad del sistema como producto informático, además de realizar una evaluación de los resultados obtenidos.

Capítulo 1: Fundamentos Teóricos

En este capítulo se definen los principales conceptos relacionados con la actividad de transporte dentro del proceso de Servicios Generales de ETECSA, y de cómo se desarrolla esta. Además se relacionan las definiciones básicas relacionadas con las aplicaciones Web, así como las tecnologías para su desarrollo y los sistemas de gestión de base de datos más utilizados, todos estos enmarcados en la categoría de software libre. Por último, se aborda qué son las metodologías de desarrollo de software, haciéndose énfasis en aquella que se seleccionó para guiar el desarrollo de software.

1.1 Objeto de Estudio

1.1.1 La Empresa de Telecomunicaciones de Cuba ETECSA

La empresa ETECSA, es una organización cubana de capital mixto y tiene como objeto social prestar los servicios públicos de telecomunicaciones, mediante la operación, instalación, explotación, comercialización y mantenimiento de redes públicas de telecomunicaciones en todo el territorio de la república de Cuba.

Esta empresa tiene una alta responsabilidad en el desarrollo socio-económico del país y en especial, en la informatización de la sociedad, garantizando una efectiva conectividad.

Como organización que presta servicios, centra su actividad en la calidad de los Recursos

Humanos sobre la base de la gestión integral de estos y se organiza en una estructura conformada por Unidades de Negocios que facilita la orientación de sus colaboradores a la satisfacción de las necesidades del mercado, dando soluciones creativas a los problemas mediante la utilización de una tecnología de avanzada y el compromiso con la calidad total, acorde a los valores y principios de la sociedad, que se ponen de manifiesto en su visión y misión empresarial.

ETECSA se ha caracterizado por llevar la prestación de modernos servicios de beneficio popular, tanto a las ciudades como a las zonas rurales de difícil acceso, respetando la naturaleza y la ecología en sus procesos inversionistas y tiene como misión proporcionar a los usuarios y a toda la población, servicios que garanticen la satisfacción de sus necesidades en materia de telecomunicaciones, respaldando los planes de desarrollo social y económico que lleva a cabo el país, las tareas de la defensa y garantizando los resultados económicos planeados. [25].

1.1.2 Sistemas de gestión basados en procesos

“Un proceso puede definirse como un conjunto de actividades de trabajo interrelacionadas, las cuales transforman elementos de entrada en un resultado de valor agregado para un cliente interno o externo”. [11]. En un proceso, los elementos de entrada y los resultados previstos pueden ser tangibles (equipos, materiales, componentes) o intangibles (energía, información).

Algunos requisitos básicos a considerar para identificar procesos a automatizar son [11]:

- ✓ Cualquier tarea monótona o repetitiva es un buen candidato para la automatización.
- ✓ Si la captura de datos puede hacerla una máquina en vez de una persona, se ahorra tiempo, independientemente de lograr una mayor exactitud.
- ✓ Transferencia de datos de un formato a otro, de una persona a otra, de un sistema a otro, es otro candidato de alta prioridad para automatizar. En algunos casos los estándares de computación han convertido esta tarea en algo innecesariamente complicado y, aún así, evitan la necesidad de capturar datos a un sistema donde ya se capturaron de otro. Esto no solo ahorra tiempo de captura sino todo un conjunto de problemas, cuando estos datos no coinciden.

- ✓ Análisis de datos que podrían llevarse a cabo sobre datos recopilados.

La adopción de un enfoque basado en procesos como sistema de gestión es una de las estrategias más importantes que presenta cualquier empresa, determinado por la necesidad de orientar sus resultados hacia la satisfacción de las necesidades y expectativas de los clientes o para mejorar la eficiencia y la eficacia de la organización. Tal enfoque consiste en la identificación y gestión sistemática de los procesos desarrollados en la organización y en particular las interacciones entre los mismos, haciendo énfasis en los resultados que se desean obtener.

1.1.3 El Proceso de Servicios Generales de ETECSA

La empresa ETECSA ha adoptado como sistema de gestión para el logro de sus objetivos, un enfoque basado en procesos.

La Dirección de Organización y Procesos de ETECSA, unidad organizativa rectora de la actividad, puso en vigor el Mapa General de Procesos de ETECSA (ver Anexo 1), en el cual se modelan los procesos de nivel 0, entre los que se encuentra el Proceso de Servicios Generales, como uno de los procesos de soporte de la Empresa.

El Proceso de Servicios Generales comprende, entre otros, la emisión de la definición de políticas, la planificación, *la gestión de la prestación de los servicios del transporte en la Empresa.*

1.1.4 La Prestación de los Servicios de Transporte en ETECSA

La Prestación de los Servicios de Transporte en la Empresa de Telecomunicaciones de Cuba S.A., permite la correcta planificación, administración y prestación de los servicios generales que requiere la empresa para su funcionamiento, ya sea ofrecida por los propios recursos humanos de la organización o con el empleo de un proveedor externo.

La Resolución No 06/06 de la Dirección de Servicios Generales, que pone en vigor las Políticas para la prestación de los servicios de transporte en la empresa, enfatiza en su cuarto por cuanto la importancia de los equipos de transporte como “uno de los recursos materiales más valiosos y de uso generalizado en todas las Unidades Organizativas (en lo adelante UO) de la empresa, que en gran medida contribuyen a elevar la productividad del trabajo”. [14].

En esa misma Resolución se resuelve que corresponde, entre otras, a los Departamentos de

Logística y Servicios de las Direcciones Territoriales “la ejecución operativa de la prestación de los servicios de transporte, según sus funciones y responsabilidades”.

Las políticas puestas en efecto en el mencionado documento establecen, entre otras cosas, que:

- ✓ Se debe mantener actualizada la plantilla de equipos de transporte por cada Unidad Organizativa, en correspondencia con las necesidades que demande el aseguramiento de la producción, la prestación de los servicios y otras actividades previstas, teniendo en cuenta potenciar el uso de equipos de alta eficiencia económica, de acuerdo a la explotación a que serán sometidos y las actividades que realizarán.
- ✓ Cada UO debe garantizar la correcta explotación del transporte, teniendo en cuenta la preparación de todos los conductores, la entrega del transporte mediante el acta correspondiente, el control de los insumos de explotación de los mismos (combustible, neumáticos, etc.), el control individual del transporte y el registro de accidentes, entre otros aspectos; siendo los máximos responsables del cumplimiento de estas medidas los jefes a todos los niveles que tengan bajo su dirección transportes y los especialistas de transporte de la Dirección Territorial.

Para una eficiente prestación de los servicios de transporte es fundamental gestionar el estado de disponibilidad técnica de los vehículos a partir del registro de los mismos y de sus incidencias en el taller, ya que los medios de transporte de la empresa están obligados a transitar presentando un estado técnico adecuado y contar con una correcta actualización de sus plantillas.

La actividad fundamental que se realiza en el Departamento de Logística y Servicio para dar soporte a la Prestación de los Servicios de Transporte en ETECSA es, la determinación de la disponibilidad del transporte, requiriendo una secuencia de acciones relacionadas que se llevan a cabo durante este proceso como, controlar de forma directa el cumplimiento de los indicadores técnicos económicos de explotación del parque automotriz en su territorio, controlar que los mantenimientos técnicos se realicen en los plazos y volúmenes que garanticen un alto coeficiente de disponibilidad técnica, organizar las reparaciones del transporte y de los equipos

para restablecer los parámetros de explotación, organizar y controlar la conservación de la técnica y equipos, garantizar la evacuación de la técnica y equipos que hayan sufrido desperfectos, organizar la recuperación y fabricación de piezas agregados y procesos tecnológicos, exigir el riguroso control de los medios técnicos, tramitaciones de alta, bajas, cambios de motor, etc.. Trabajar por mantener los índices propuestos y la prevención de accidentes del tránsito.

Para el caso de la DTHO, la responsabilidad mayor en la prestación de estos servicios recae en los especialistas de la Unidad de Servicio al Transporte, perteneciente al departamento de Logística y Servicio, que se encargan de garantizar una adecuada explotación y mantenimiento del transporte del territorio rindiendo cuenta a la Dirección Territorial.

1.1.5 Reglas a considerar que debe cumplir la actividad de transporte

- ✓ El control de los transportes de la empresa, realizado por los especialistas del transporte en el Departamento de Logística y Servicio, consta del registro de las características de cada transporte, el registro de sus piezas y accesorios, además de los accidentes y eventualidades que puede presentar el transporte.
- ✓ Las órdenes de trabajo contienen la información del transporte a la entrada y salida del taller de transporte, donde se encargan de realizar los trabajos para solucionar las interrupciones que le ocurren a los transportes de la empresa.
- ✓ Cada informe de un transporte contiene los siguientes campos: matrícula, modelo, color, tipo de combustible que usa, centro de costo al que pertenece, estado técnico, parqueo, trabajador responsable, chofer, actividad, municipio, VIN, DAM, código, número consecutivo, acreditación, patrón de consumo, FE, RMT, número de inventario, número de circulación, número de serie, garantía, fecha de recepción, fecha de explotación y la fecha de fabricación.
- ✓ El número de caracteres que tiene la matrícula de un transporte es de seis.
- ✓ Una eventualidad en el transporte representa cualquier interrupción ocurrida, ya sea por mantenimiento, reparación, trámite de baja, etc.
- ✓ La confección de los informes que representan el inventario del transporte es realizado por los especialistas del transporte.
- ✓ La información referente a los trabajadores debe corresponderse con la que mantiene

el Departamento de Capital Humano.

- ✓ El especialista emite informes del control del transporte a la Dirección Territorial, tales como, el inventario general de vehículos, inventario resumido de vehículos, listado de vehículos paralizados, informe de la disponibilidad de los vehículos, informe de los accidentes de los vehículos y el Modelo I de los vehículos.
- ✓ La confección de cada informe contiene, nombre del informe y fecha de emisión.
- ✓ El estado técnico de un transporte solo se clasifica en las categorías de, Bien, Regular o Mal.
- ✓ La clasificación de un accidente es de, Leve, Menos Grave o Grave.

1.2 Contexto teórico del desarrollo de aplicaciones Web

Para realizar el sistema SAGTRANS, se hizo un análisis del desarrollo Web de aplicaciones en la actualidad, y de las principales tecnologías usadas en el mismo, a continuación se enuncian los conceptos y características fundamentales que sirven de fundamento teórico para esta investigación.

1.2.1 ¿Por qué Aplicación Web?

Hoy en día el auge de las redes locales (empresariales o institucionales) y la popularidad de la Internet ha ofrecido la oportunidad de acceso a través de computadoras y otros dispositivos móviles, lo que era algo de uso privilegiado hace algunos años, ahora es común. La Internet ha elevado y extendido aun más el concepto de aplicación Web para servir no sólo a usuarios de una pequeña red sino ubicados en cualquier sitio donde tenga acceso a la Internet.

La esencia del concepto es: no dejar que el cliente realice demasiadas tareas, solo lo necesario para que lleve a cabo su trabajo y dejar que en el lado del servidor se realicen las operación importantes: almacenamiento de datos, transacciones, reglas del negocio y la lógica del programa.

Se decidió implementar el sistema informático, como Aplicación Web, debido a la necesidad de que la información pueda ser accedida desde cualquier punto en la red de la empresa y depender solo de un navegador Web como cliente para hacer uso de ella, sin necesidad de contar con una máquina computadora con elevados recursos de procesamiento o una gran capacidad de memoria física, posibilidad que se hace más difícil y costoso en el caso de una

aplicación de escritorio, pues esta debería estar desplegada en cada equipo individualmente de donde se quisiera acceder a la información, sumándole a esto que el mantenimiento y las actualizaciones que se le hagan a la aplicación resultan también más fácil ya que estas se realizarían solo en el servidor que se encuentra la aplicación, a diferencia de las aplicaciones de escritorio.

Se pudieran mencionar otros factores que no son tan medulares, pero que en cierta medida favorece a la hora de decidirse por Aplicación Web o de Escritorio, como puede ser el hecho que en estos últimos años las Aplicaciones Web han alcanzado gran popularidad en el contexto del desarrollo de sistemas informáticos, pues amplias son las posibilidades y funcionalidades que han proporcionado los distintos lenguajes de programación Web, para facilitar mucho más el trabajo de los programadores y enriquecer los sistemas en todas las partes que lo componen.

Por todo lo mencionado y debido a que resultó ser una petición del cliente, el sistema SAGTRANS fue implementado como una Aplicación Web.

1.2.2 ¿Qué es una Aplicación Web?

Una Aplicación Web, es aquella 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 un software que se codifica en un lenguaje soportado por los navegadores Web (HTML, JavaScript, Java, etc.) en la que se confía la ejecución al navegador. [15].

Las aplicaciones Web contienen 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. Además generan dinámicamente una serie de páginas en un formato estándar, como HTML o XHTML, soportadas por los navegadores Web comunes. Se utilizan lenguajes interpretados en el lado del cliente, tales como JavaScript, para añadir elementos dinámicos a la interfaz de usuario. Generalmente cada página Web en particular se envía al cliente como un documento estático, pero la secuencia de páginas ofrece al usuario una experiencia interactiva. Durante la sesión, el navegador Web interpreta y muestra en pantalla las páginas, actuando como cliente para cualquier Aplicación Web. [15].

Aunque existen muchas variaciones posibles, una Aplicación Web está normalmente

estructurada como una aplicación de tres-capas. En su forma más común, el navegador Web ofrece la primera capa y un motor capaz de usar alguna tecnología Web dinámica (ejemplo: PHP, Java Servlets o ASP, ASP.NET, CGI, ColdFusion, embPerl, Python (programming language) o Ruby on Rails) que constituye la capa intermedia. Por último, una base de datos constituye la tercera y última capa. El navegador Web manda peticiones a la capa intermedia que ofrece servicios valiéndose de consultas y actualizaciones a la base de datos y a su vez proporciona una interfaz de usuario. [15].

1.2.3 Arquitectura Cliente-Servidor

La arquitectura Cliente-servidor consiste básicamente en un programa cliente que realiza peticiones a otro programa que se encuentra en el servidor este le proporciona la respuesta. Aunque esta idea se puede aplicar a programas que se ejecutan sobre una sola computadora es más ventajosa en un sistema operativo multiusuario distribuido a través de una red de computadoras. [17].

En esta arquitectura la capacidad de proceso está repartida entre los clientes y los servidores, aunque son más importantes las ventajas de tipo organizativo debidas a la centralización de la gestión de la información y la separación de responsabilidades, lo que facilita y clarifica el diseño del sistema.

La separación entre cliente y servidor es una separación de tipo lógico, donde el servidor no se ejecuta necesariamente sobre una sola máquina ni es necesariamente un sólo programa. Los tipos específicos de servidores incluyen los servidores Web, los servidores de archivo, los servidores del correo, etc. Mientras que sus propósitos varían de unos servicios a otros, la arquitectura básica seguirá siendo la misma. Una disposición muy común son los sistemas multicapa en los que el servidor se descompone en diferentes programas que pueden ser ejecutados por diferentes computadoras aumentando así el grado de distribución del sistema. [17].

Algunas ventajas que brinda esta arquitectura:

- ✓ Centralización del control: los accesos, recursos y la integridad de los datos son controlados por el servidor de forma que un programa cliente defectuoso o no autorizado no pueda dañar el sistema. Esta centralización también facilita la tarea de

poner al día datos u otros recursos (mejor que en las redes P2P).

- ✓ Escalabilidad: se puede aumentar la capacidad de clientes y servidores por separado. Cualquier elemento puede ser aumentado (o mejorado) en cualquier momento, o se pueden añadir nuevos nodos a la red (clientes y/o servidores).
- ✓ Fácil mantenimiento: al estar distribuidas las funciones y responsabilidades entre varios ordenadores independientes, es posible reemplazar, reparar, actualizar, o incluso trasladar un servidor, mientras que sus clientes no se verán afectados por ese cambio (o se afectarán mínimamente).
- ✓ Existen tecnologías, suficientemente desarrolladas, diseñadas para el paradigma de C/S que aseguran la seguridad en las transacciones, la amigabilidad del interfaz, y la facilidad de empleo.

1.2.4 Software libre y licencias

Cuando se hace referencia al software libre se está indicando la libertad del usuario para ejecutar, copiar, distribuir, estudiar, cambiar y mejorar el software.

Esto se resume de manera más precisa en cuatro libertades [19]:

- ✓ *Libertad 0*: La libertad de usar el programa, con cualquier propósito.
- ✓ *Libertad 1*: La libertad de estudiar cómo funciona el programa, y adaptarlo a tus necesidades. El acceso al código fuente es una condición previa para esto.
- ✓ *Libertad 2*: La libertad de distribuir copias, con lo que puedes ayudar a tu vecino.
- ✓ *Libertad 3*: La libertad de mejorar el programa y hacer públicas las mejoras a los demás, de modo que toda la comunidad se beneficie. El acceso al código fuente es un requisito previo para esto.

“Siempre que se habla de software libre se suele especificar que para entender el concepto, se debe pensar en libre como en libertad de expresión, no como en cerveza gratis” [19], haciendo énfasis en que se trata de un asunto de libertad y no de precio. Esto se debe fundamentalmente a la confusión que puede existir con el término en inglés en el que una misma palabra (*free*) significa tanto libre como gratis, lo que ha dado lugar a cierta confusión.

También hay quien defiende el uso del término de Código Abierto (*Open Source*, en inglés),

pero este último se basa únicamente en los aspectos técnicos, mientras que el Software Libre tiene en cuenta los aspectos éticos y filosóficos de la libertad , por lo que se prefiere este. [20].

Una licencia es una autorización formal con carácter contractual que un autor de un software da a un interesado para que este explote el sistema legalmente. En el caso del software libre, para que las libertades antes mencionadas sean reales, deben ser irrevocables mientras no se haga nada incorrecto; por tanto, el desarrollador del software no debe poder revocar la licencia si no se le han dado motivos justificados para ello. (Las posibles normas de control de exportación de un gobierno y las sanciones mercantiles no constituyen condiciones de uso para ningún software que clasifique como libre, no siendo estas, por ende, motivos justificables para impedir la explotación del software.) [20].

En definitiva, se pueden definir las licencias de software libre como aquellas que, mediante la puesta a disposición del código fuente del programa de ordenador, permiten y aseguran a los usuarios el ejercicio de las libertades de ejecutar, copiar, distribuir, estudiar, cambiar y mejorar el software. [13].

1.2.5 El servidor HTTP Apache

Se ha escogido el servidor HTTP Apache para la aplicación, ya que este entra en la categoría de software (libre), es un servidor HTTP de código abierto para plataformas Unix (BSD, GNU/Linux, etc.), Windows, Macintosh y otras, que implementa el protocolo HTTP/1.1 y la noción de sitio virtual.

El servidor Apache se desarrolla dentro del proyecto HTTP Server de la Apache Software Foundation. Presenta entre otras características mensajes de error altamente configurables, bases de datos de autenticación y negociado de contenido, aunque fue criticado por la falta de una interfaz gráfica que ayude en su configuración. [21].

Apache tiene amplia aceptación en la red: desde 1996, Apache, es el servidor HTTP más usado. Alcanzó su máxima cuota de mercado en 2005 siendo el servidor empleado en el 70% de los sitios Web en el mundo, sin embargo ha sufrido un descenso en su cuota de mercado en los últimos años. (Estadísticas históricas y de uso diario proporcionadas por Netcraft). [21].

Entre sus ventajas se puede mencionar, que es:

- ✓ Modular

- ✓ Open source
- ✓ Multi-plataforma
- ✓ Extensible
- ✓ Popular (fácil de conseguir ayuda/suporte)

La arquitectura del servidor Apache es muy modular. El servidor consta de una sección core y diversos módulos que aportan mucha de la funcionalidad que podría considerarse básica para un servidor Web. Este es usado principalmente para enviar páginas Web estáticas y dinámicas en la World Wide Web. Muchas aplicaciones Web están diseñadas asumiendo como ambiente de implantación a Apache, o que utilizarán características propias de este servidor Web.

1.3 Tecnologías para el desarrollo de Aplicaciones Web

1.3.1 El lenguaje de programación Web PHP

PHP es un lenguaje de programación interpretado, diseñado originalmente para la creación de páginas Web dinámicas. Es usado principalmente en 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 las bibliotecas Qt o GTK. [22].

PHP es un acrónimo recursivo que significa PHP Hypertext Pre-processor (inicialmente PHP Tools, o, Personal Home Page Tools). Fue creado originalmente por Rasmus Lerdorf en 1994; sin embargo la implementación principal de PHP es producida ahora por The PHP Group y sirve como el estándar de facto para PHP al no haber una especificación formal. Publicado bajo la PHP License, la Free Software Foundation considera esta licencia como software libre.

PHP es un lenguaje interpretado de propósito general ampliamente usado y que está diseñado especialmente para desarrollo Web y puede ser incrustado dentro de código HTML. Generalmente se ejecuta en un servidor Web, tomando el código en PHP como su entrada y creando páginas Web como salida. Puede ser desplegado en la mayoría de los servidores Web y en casi todos los sistemas operativos y plataformas sin costo alguno. PHP se encuentra instalado en más de 20 millones de sitios Web y en un millón de servidores. [22].

Entre sus ventajas se destacan:

Es un lenguaje multiplataforma.

- ✓ Capacidad de conexión con la mayoría de los manejadores de base de datos que se utilizan en la actualidad, destaca su conectividad con MySQL.
- ✓ Capacidad de expandir su potencial utilizando la enorme cantidad de módulos (llamados ext's o extensiones).
- ✓ Posee una amplia documentación en su página oficial, entre la cual se destaca que todas las funciones del sistema están explicadas y ejemplificadas en un único archivo de ayuda.
- ✓ Es libre, por lo que se presenta como una alternativa de fácil acceso para todos.
- ✓ Permite las técnicas de Programación Orientada a Objetos.
- ✓ Biblioteca nativa de funciones sumamente amplia e incluida.
- ✓ No requiere definición de tipos de variables.
- ✓ Tiene manejo de excepciones (desde PHP5).

1.3.2 Los frameworks

“Un Framework, en el desarrollo de software, es una estructura de soporte definida mediante la cual otro proyecto de software puede ser organizado y desarrollado. Típicamente, puede incluir soporte de programas, bibliotecas y un lenguaje interpretado entre otros software para ayudar a desarrollar y unir los diferentes componentes de un proyecto”. [16].

Estos representan una arquitectura de software que modela las relaciones generales de las entidades del dominio, además de proveer una estructura y una metodología de trabajo la cual extiende o utiliza las aplicaciones del dominio. Son diseñados con el intento de facilitar el desarrollo de software, permitiendo a los diseñadores y programadores pasar más tiempo identificando requerimientos de software que tratando con los tediosos detalles de bajo nivel para proveer un sistema funcional. Fuera del contexto de los términos informáticos, pueden ser considerados como el conjunto de procesos y tecnologías usados para resolver un problema complejo. Es el esqueleto sobre el cual varios objetos son integrados para una solución dada. A esto se le suma la capacidad de extenderse sin prejuicios para diversificar la expresión del programa mismo. [16].

En la actualidad existen varios frameworks para PHP, algunos de los más populares son, el

Zend Frameworks, que no necesita de una instalación especial, requiere PHP 5 e incorpora el patrón MVC. El framework Prado, que está basado en componentes eventos con el objetivo de acelerar el desarrollo de aplicaciones Web usando PHP 5, el concepto del desarrollo de aplicaciones en Prado es diferente, se utilizan componentes, eventos y propiedades en vez de procedimientos, URL y parámetros. Este Framework combina especificaciones en un archivo XML, plantillas HTML y una clase PHP. Prado, cuenta con soporte para AJAX², validación, autenticación, plantillas y múltiples bases de datos. Otro de los más usados es el CodeIgniter, es utilizado por una gran comunidad de usuarios, construido para codificadores PHP que necesitan una herramienta de desarrollo fácil para crear aplicaciones Web simples y elegantes, entre sus características se encuentran, su compatibilidad con PHP 4 y PHP 5, incorpora el modelo MVC, soporte para múltiples bases de datos, plantillas, validaciones, no requiere instalación y se pueden encontrar una librería con un gran número de clases. Por último el framework Symfony, que también brinda las mismas posibilidades de los ya mencionados, este fue seleccionado para el desarrollo de la aplicación por las características que a continuación se detallan y por ser una petición especial de parte del cliente. [16].

1.3.3 Framework Symfony

Symfony es un framework diseñado para optimizar el desarrollo de las aplicaciones Web. Para empezar, separa la lógica de negocio, la lógica de servidor y la presentación de la Aplicación Web. Proporciona varias herramientas y clases encaminadas a reducir el tiempo de desarrollo de una Aplicación Web compleja. Además, automatiza las tareas más comunes, permitiendo al desarrollador dedicarse por completo a los aspectos específicos de cada aplicación. El resultado de todas estas ventajas es que no se debe reinventar la rueda cada vez que se crea una nueva Aplicación Web. Symfony está desarrollado completamente con PHP 5. Ha sido probado en numerosos proyectos reales y se utiliza en sitios Web de comercio electrónico de primer nivel. Symfony es compatible con la mayoría de gestores de bases de datos, como MySQL, PostgreSQL, Oracle y SQL Server de Microsoft. Se puede ejecutar tanto en plataformas *nix (Unix, Linux, etc.) como en plataformas Windows. [10].

Symfony se diseñó para que se ajustara a los siguientes requisitos:

² AJAX: tecnología que permite realizar peticiones en segundo plano a un sitio Web sin recargar el documento.

- ✓ Fácil de instalar y configurar en la mayoría de plataformas (y con la garantía de que funciona correctamente en los sistemas Windows y *nix estándares).
- ✓ Independiente del sistema gestor de bases de datos.
- ✓ Sencillo de usar en la mayoría de casos, pero lo suficientemente flexible como para adaptarse a los casos más complejos.
- ✓ Basado en la premisa de "convenir en vez de configurar", en la que el desarrollador solo debe configurar aquello que no es convencional.
- ✓ Sigue la mayoría de mejores prácticas y patrones de diseño para la Web.
- ✓ Preparado para aplicaciones empresariales y adaptables a las políticas y arquitecturas propias de cada empresa, además de ser lo suficientemente estable como para desarrollar aplicaciones a largo plazo.
- ✓ Código fácil de leer que incluye comentarios de phpDocumentor y que permite un mantenimiento muy sencillo.
- ✓ Fácil de extender, lo que permite su integración con librerías desarrolladas por terceros.

Symfony automatiza la mayoría de elementos comunes de los proyectos Web, como por ejemplo:

- ✓ La capa de internacionalización que incluye Symfony permite la traducción de los datos y de la interfaz, así como la adaptación local de los contenidos.
- ✓ La capa de presentación utiliza plantillas y layouts que pueden ser creados por diseñadores HTML sin ningún tipo de conocimiento del framework. Los helpers incluidos permiten minimizar el código utilizado en la presentación, ya que encapsulan grandes bloques de código en llamadas simples a funciones.
- ✓ Los formularios incluyen validación automatizada y relleno automático de datos, lo que asegura la obtención de datos correctos y mejora la experiencia de usuario.
- ✓ Los datos incluyen mecanismos de escape que permiten una mejor protección contra los ataques producidos por datos corruptos.
- ✓ La gestión de la caché reduce el ancho de banda utilizado y la carga del servidor.
- ✓ La autenticación y la gestión de credenciales simplifican la creación de secciones

restringidas y la gestión de la seguridad de usuario.

- ✓ El sistema de enrutamiento y las URL limpias permiten considerar a las direcciones de las páginas como parte de la interfaz, además de estar optimizadas para los buscadores.
- ✓ El soporte de e-mail incluido y la gestión de APIs permiten a las aplicaciones Web interactuar más allá de los navegadores.
- ✓ Los listados son más fáciles de utilizar debido a la paginación automatizada, el filtrado y la ordenación de datos.
- ✓ Los plugins, las factorías (patrón de diseño "Factory") y los "mixin" permiten realizar extensiones a medida de Symfony.

Las interacciones con Ajax son muy fáciles de implementar mediante los helpers que permiten encapsular los efectos JavaScript compatibles con todos los navegadores en una única línea de código.

1.3.4 Sistema Gestor de Base de Datos

Los sistemas de gestión de base de datos (SGBD); (en inglés: *DataBase Management System*, abreviado DBMS) son un tipo de software muy específico, dedicado a servir de interfaz entre la base de datos, el usuario y las aplicaciones que la utilizan. El propósito general de los sistemas de gestión de base de datos es el de manejar de manera clara, sencilla y ordenada un conjunto de datos que posteriormente se convertirán en información relevante, para un buen manejo de los datos. [18].

En la actualidad existen una gran cantidad de Sistemas Gestores de Bases de Datos en el mercado. Entre los gestores libres, PostgreSQL es considerado uno de los sistemas gestores de bases de datos más usados en el mundo. A continuación se describen las ventajas que conlleva utilizar el PostgreSQL como sistema de gestión de los datos en el desarrollo del sistema informático SAGTRANS.

PostgreSQL

Es un sistema de base de datos objeto-relacional que tiene las características de sistemas de bases de datos comerciales tradicionales con mejoras que serán encontradas en sistemas

DBMS de la próxima generación. PostgreSQL es libre y está disponible todo su código fuente, característica que beneficia su preferencia. Su desarrollo es realizado por un equipo de programadores (voluntarios en su mayoría) dispersos alrededor del mundo y comunicados vía Internet. Este es un proyecto de la comunidad y no es controlado por ninguna compañía. [5].

El modelo relacional sustituyó modelos previos, en parte por su simplicidad. Sin embargo, esta simplicidad hace muy difícil la implementación de ciertas aplicaciones. PostgreSQL ofrece una potencia adicional sustancial al incorporar los siguientes cuatro conceptos adicionales básicos en una vía en la que los usuarios pueden extender fácilmente el sistema [9]:

- ✓ Clases: Esta es una de las características fundamentales de PostgreSQL y es la colección de instancias de un objeto. Cada instancia tiene la misma colección de atributos y cada atributo es de un tipo específico, además cada instancia tiene un identificador de objeto (OID).
- ✓ Herencia: Una clase puede heredar de ninguna o varias otras clases y una consulta puede hacer referencia a todas las instancias de una clase.
- ✓ Tipos: Pueden ser definidos nuevos tipos de datos además de los convencionales.
- ✓ Funciones: Las funciones de conjunto calculan un único resultado a partir de múltiples filas de entrada.

Entre sus ventajas se encuentran:

- ✓ Cero problemas de licencias.
- ✓ Costo de adquisición bajo o nulo.
- ✓ En las comunidades se brinda soporte gratuito.
- ✓ En las comunidades participan programadores del mismo motor de la base de datos.
- ✓ Se puede modificar según las necesidades (acceso a código fuente).
- ✓ Si se usa en Linux, BSD o *nix es inmune a los virus.
- ✓ Soporta la creación de procedimientos almacenados y disparadores en diferentes lenguajes como pl/pgsql; java; c; phyton; tcl entre otros.

1.4 Metodologías para el desarrollo de Software

El desarrollo de software no es una tarea fácil. Prueba de ello es que existen numerosas propuestas metodológicas que inciden en distintas dimensiones del proceso de desarrollo. Por una parte se tienen aquellas propuestas más tradicionales que se centran especialmente en el control del proceso, estableciendo rigurosamente las actividades involucradas, los artefactos que se deben producir, y las herramientas y notaciones que se usarán. Otra aproximación es centrarse en otras dimensiones, como por ejemplo el factor humano o el producto software. Esta es la filosofía de las metodologías ágiles, las cuales dan mayor valor al individuo, a la colaboración con el cliente y al desarrollo incremental del software con iteraciones muy cortas.

Un Proceso de Desarrollo de Software es la definición del conjunto de actividades que guían los esfuerzos de las personas implicadas en el proyecto, a modo de plantilla que explica los pasos necesarios para terminar el proyecto. [7].

Para escoger la metodología a usar es necesario un estudio previo de las existentes, y ver cual de ellas es la más conveniente en dependencia del contexto de desarrollo y de las características específicas del proyecto.

A continuación se describen las metodologías analizadas en este trabajo, siendo Scrum la seleccionada para el desarrollo de la presente investigación.

1.4.1 Proceso unificado de desarrollo (RUP)

El proceso unificado de desarrollo (RUP) llamada así por sus siglas en inglés Rational Unified Process, es una metodología para la ingeniería de software, que proporciona una familia de técnicas que soportan el ciclo completo de desarrollo de software. El resultado es un proceso basado en componentes, dirigido por los casos de uso, centrado en la arquitectura, iterativo e incremental que utiliza el Lenguaje Unificado de Modelado (UML) para describir un sistema. [7].

RUP divide el proceso en cuatro fases, dentro de las cuales se realizan varias iteraciones en número variable según el proyecto y en las que se hace un mayor o menor hincapié en las distintas actividades.

1. Inicio, el objetivo en esta etapa es determinar la visión del proyecto.
2. Elaboración, en esta etapa el objetivo es determinar la arquitectura óptima.
3. Construcción, en esta etapa el objetivo es llevar a obtener la capacidad operacional inicial.

4. Transmisión, el objetivo es llegar a obtener el realce del proyecto.

Cada una de estas etapas es desarrollada mediante el ciclo de iteraciones, la cual consiste en reproducir el ciclo de vida en cascada a menor escala. Los Objetivos de una iteración se establecen en función de la evaluación de las iteraciones precedentes.

Esta es una propuesta tradicional que se centra especialmente en el control del proceso, estableciendo rigurosamente las actividades involucradas, los artefactos que se deben producir, y las herramientas y notaciones que se usarán. Esta ha demostrado ser efectiva y necesaria en un gran número de proyectos, pero también en algunos no ha sido funcional.

1.4.2 Metodologías ágiles

En febrero de 2001, tras una reunión celebrada en Utah-EEUU, nace el término “ágil” aplicado al desarrollo de software. En esta reunión participan un grupo de 17 expertos de la industria del software, incluyendo algunos de los creadores o impulsores de metodologías de software. Su objetivo fue esbozar los valores y principios que deberían permitir a los equipos desarrollar software rápidamente y respondiendo a los cambios que puedan surgir a lo largo del proyecto.

Se pretendía ofrecer una alternativa a los procesos de desarrollo de software tradicionales, caracterizados por ser rígidos y dirigidos por la documentación que se genera en cada una de las actividades desarrolladas. Tras esta reunión se creó The Agile Alliance 3, una organización, sin ánimo de lucro, dedicada a promover los conceptos relacionados con el desarrollo ágil de software y ayudar a las organizaciones para que adopten dichos conceptos. [6].

Hasta hace poco el proceso de desarrollo llevaba asociada un marcado énfasis en el control del proceso mediante una rigurosa definición de roles, actividades y artefactos, incluyendo modelado y documentación detallada. Este esquema "tradicional" para abordar el desarrollo de software ha demostrado ser efectivo y necesario en proyectos de gran tamaño (respecto a tiempo y recursos), donde por lo general se exige un alto grado de ceremonia en el proceso. Sin embargo, este enfoque no resulta ser el más adecuado para muchos de los proyectos actuales donde el entorno del sistema es muy cambiante, y en donde se exige reducir drásticamente los tiempos de desarrollo pero manteniendo una alta calidad. [6].

Ante las dificultades para utilizar metodologías tradicionales con estas restricciones de tiempo y flexibilidad, las metodologías ágiles emergen como una posible respuesta para llenar

ese vacío metodológico, garantizando la calidad del producto.

Estas están basadas en heurísticas provenientes de prácticas de producción de código, especialmente preparadas para cambios durante el proyecto, es un proceso menos controlado, el cliente es parte del equipo de desarrollo, con grupos pequeños (<10 integrantes) trabajando en el mismo sitio, con pocos artefactos, pocos roles y menos énfasis en la arquitectura de software. [6].

1.4.3 Programación extrema (XP)

XP es una metodología ágil centrada en potenciar las relaciones interpersonales como clave para el éxito en desarrollo de software, promoviendo el trabajo en equipo, preocupándose por el aprendizaje de los desarrolladores, y propiciando un buen clima de trabajo. XP se basa en realimentación continua entre el cliente y el equipo de desarrollo, comunicación fluida entre todos los participantes, simplicidad en las soluciones implementadas y coraje para enfrentar los cambios. XP se define como especialmente adecuada para proyectos con requisitos imprecisos y muy cambiantes, y donde existe un alto riesgo técnico. [6].

1.4.4 Metodologías Crystal

Se trata de un conjunto de metodologías para el desarrollo de software caracterizadas por estar centradas en las personas que componen el equipo y la reducción al máximo del número de artefactos producidos. Han sido desarrolladas por Alistair Cockburn. El desarrollo de software se considera un juego cooperativo de invención y comunicación, limitado por los recursos a utilizar. El equipo de desarrollo es un factor clave, por lo que se deben invertir esfuerzos en mejorar sus habilidades y destrezas, así como tener políticas de trabajo en equipo definidas. [6].

1.4.5 Metodología Scrum

Scrum fue Desarrollada por Ken Schwaber, Jeff Sutherland y Mike Beedle. Define un marco para la gestión de proyectos, que se ha utilizado con éxito durante los últimos 10 años. Está especialmente indicada para proyectos con un rápido cambio de requisitos. Sus principales características se pueden resumir en dos. El desarrollo de software se realiza mediante iteraciones, denominadas Sprints (una iteración es un ciclo corto de construcción repetitivo), estas iteraciones están diseñadas para ser cortas y de duración fija, cada iteración termina con

una pieza de software ejecutable que incorpora nueva funcionalidad y se muestra al cliente. Los Sprints en general tienen una duración entre 2 y 4 semanas. La segunda característica importante son las reuniones a lo largo del proyecto, entre ellas destaca la reunión diaria de 15 minutos del equipo de desarrollo para coordinación e integración. [6].

Este enfoque en entregar código funcional cada poco tiempo significa que los equipos Scrum no disponen de mucho tiempo para teorías. No persiguen dibujar el modelo UML perfecto en una herramienta CASE, o escribir código que se adapte a todos los cambios futuros imaginables. Los equipos Scrum se enfocan y ponen su empeño en que las cosas se implementen. Estos equipos admiten que pueden existir equivocaciones por el camino, pero también son conscientes de que la mejor manera de encontrar dichos errores es dejar de pensar en el software a un nivel teórico de análisis y diseño y sumergirse en él y comenzar a construir el producto. [12].

La metodología Scrum fue de mayor preferencia que las otras metodologías analizadas, ya que como su categoría lo sugiere, esta agiliza el proceso de desarrollo de software, siendo el tiempo un factor determinante en esta investigación, sumándole a esto que la mayor parte de la tecnología a usar referente al contexto del desarrollo de aplicaciones Web, era desconocida por parte del programador (máximo encargado de la implementación del software), y además de ser la metodología seleccionada por el cliente, ya que entre las políticas de la empresa está el uso de la metodología Scrum para el desarrollo de software.

1.4.6 Lenguaje Unificado de Modelación (UML)

Durante el desarrollo del sistema se hace uso de este Lenguaje Unificado de Modelado (UML, por sus siglas en inglés, Unified Modeling Language) que es el lenguaje de modelado de sistemas de software más conocido y utilizado en la actualidad; está respaldado por el OMG (Object Management Group). Es un lenguaje gráfico para visualizar, especificar, construir y documentar un sistema. UML ofrece un estándar para describir un "plano" del sistema (modelo), incluyendo aspectos conceptuales tales como procesos de negocio y funciones del sistema, y aspectos concretos como expresiones de lenguajes de programación, esquemas de bases de datos y componentes reutilizables. [23].

Es importante resaltar que UML es un "lenguaje de modelado" para especificar o para describir métodos o procesos. Se utiliza para definir un sistema, para detallar los artefactos en

el sistema y para documentar y construir. En otras palabras, es el lenguaje en el que está descrito el modelo.

Se puede aplicar en el desarrollo de software entregando gran variedad de formas para dar soporte a una metodología de desarrollo de software, pero no especifica en sí mismo qué metodología o proceso usar.

UML cuenta con varios tipos de diagramas, los cuales muestran diferentes aspectos de las entidades representadas. [23].

En UML 2.0 hay 13 tipos diferentes de diagramas.

Los Diagramas de Estructura enfatizan en los elementos que deben existir en el sistema modelado:

- ✓ Diagrama de clases.
- ✓ Diagrama de componentes.
- ✓ Diagrama de objetos.
- ✓ Diagrama de estructura compuesta (UML 2.0).
- ✓ Diagrama de despliegue.
- ✓ Diagrama de paquetes.

Los Diagramas de Comportamiento enfatizan en lo que debe suceder en el sistema modelado:

- ✓ Diagrama de actividades.
- ✓ Diagrama de casos de uso.
- ✓ Diagrama de estados.

Los Diagramas de Interacción son un subtipo de diagramas de comportamiento, que enfatiza sobre el flujo de control y de datos entre los elementos del sistema modelado:

- ✓ Diagrama de secuencia.
- ✓ Diagrama de comunicación, que es una versión simplificada del Diagrama de

- colaboración (UML 1.x).
- ✓ Diagrama de tiempos (UML 2.0).
- ✓ Diagrama global de interacciones o Diagrama de vista de interacción (UML 2.0).

De los diagramas mencionados, se usaron solo los que eran convenientes para el desarrollo de Software, bajo el criterio de la metodología usada.

1.5 Conclusiones

En este capítulo se realizó un estudio acerca de la Prestación de los Servicios de Transporte en ETECSA dentro del proceso de Servicios Generales de la empresa y que políticas debe cumplir este. Permitiendo comprender mejor su funcionamiento, favoreciendo el posterior desarrollo de la investigación.

Se corroboró que desarrollar el sistema informático como Aplicación Web es lo más conveniente para la empresa, y que el uso del Framework Symfony teniendo como base el lenguaje de programación Web PHP 5 y el PostgreSQL como sistema gestor de base de datos, constituyen una excelente opción para el desarrollo de sistemas Web Cliente-Servidor.

Luego de un análisis de otras metodologías se escogió la metodología ágil para el desarrollo de software Scrum. Considerando que a pesar de ser mucho más joven que las metodologías tradicionales, permite que la gestión de proyectos se realice con una alta calidad.

Capítulo 2: Diseño y construcción de la propuesta.

En este capítulo se presenta la solución propuesta para el desarrollo del sistema SAGTRANS a través de los flujos de trabajo que propone la metodología para desarrollo de software Scrum.

Además, se realiza un estudio de la sostenibilidad del sistema informático propuesto a través de las dimensiones: económica, socio cultural, ambiental y administrativa. Además, se hace una exposición de los resultados que se alcanzaron con el desarrollo del sistema, a través de encuestas que midieron la satisfacción de los usuarios del sistema con respecto al mismo.

2.1 Flujo de trabajo de Scrum

Según el flujo de trabajo que plantea la metodología Scrum, el equipo de trabajo, debe contar con un Dueño del Producto, que es el que representa a todos los interesados en el producto final y fija las tareas a completar y las prioridades. El Dueño del Producto debe formular un plan de modo que se incluya en la Pila de Producto con estimaciones creadas por el equipo. El equipo planifica cada Sprint³ y qué funcionalidades se incluirán a desarrollar, además de fabricar un gráfico Burn-Down⁴ por cada Sprint. Por último el equipo debe tener una reunión

³ Sprint: una iteración de aproximadamente 30 días de calendario consecutivos de trabajo.

⁴ Burn-Down: gráfico que permite tener una visión del proyecto.

diaria aproximadamente de 15 minutos con el Scrum Master⁵ (en la presente este rol lo representa también el Dueño del Producto), donde cada miembro dice qué ha hecho, qué va a hacer y que observaciones tiene sobre el cumplimiento del plan.

2.1.1 Construcción de la Pila de Producto

La Pila de Producto es el corazón de Scrum, es donde empieza todo, básicamente una lista priorizada de requisitos, o historias, o funcionalidades. Cosas que el cliente quiere, descritas usando su terminología. Se le llama a esto historias, o a veces simplemente elementos de la pila, y que estos elementos son dinámicos, manejan constantemente los cambios para identificar que necesita el producto para ser, apropiado, competitivo, y útil. [12].

Las historias incluyen los siguientes campos:

- ✓ *ID* – Es un identificador único, simplemente un número auto-incremental. Que permite no perder la pista a las historias.
- ✓ *Nombre* – Es una descripción corta de la historia. Suficientemente clara como para que el Dueño del Producto comprenda aproximadamente de qué trata y para distinguirla de las otras historias.
- ✓ *Importancia* – El ratio de importancia que el Dueño del Producto asigna a esta historia.
- ✓ *Estimación inicial* – La valoración inicial del Equipo acerca de cuanto trabajo es necesario para implementar la historia, comparada con otras historias. La unidad son “puntos de historia” y usualmente corresponde a “días-persona ideales”. Lo importante no es que las estimaciones absolutas sean correctas (es decir, que una historia de 2 puntos deba durar 2 días), lo importante es que las estimaciones relativas sean correctas (es decir, que una historia de 2 puntos debería durar la mitad que una historia de 4 puntos).
- ✓ *Como probarlo* – Una descripción a alto nivel de como se demostrará esta historia en la demostración al final del Sprint. Se trata, esencialmente, de una simple especificación de un test.
- ✓ *Notas* – cualquier otra información, clarificación, referencia a otras fuentes de

⁵ Scrum Master: Es el responsable del proceso Scrum.

información, etc que sean necesarias. Normalmente muy breve.

En la Pila de producto construida se incluyen 22 historias, cada una con los campos que anteriormente se describieron para ser consultada ver el anexo 2.

La metodología de desarrollo de software Scrum propone que se escojan aquellos artefactos o diagramas UML que el equipo de trabajo considere necesarios e indispensables para el desarrollo del software. Con vista a mejorar la comprensión de las particularidades del sistema SAGTRANS se requiere realizar una especificación del comportamiento del sistema, que incluya:

1. Una especificación de los requerimientos del sistema, con un modelo de casos de uso del sistema por paquetes, haciendo uso del lenguaje UML.
2. Diagrama de los actores del sistemas con sus relaciones.
3. Un diagrama de paquetes que muestre las agrupaciones lógicas del sistema.
4. Diagrama de Clases del Diseño para los casos de usos más importantes.
5. Un diagrama de las clases persistentes del diseño.
6. Un diagrama de despliegue que modele la distribución física de los componentes del software en los nodos de cómputo.

2.1.2 Planificación de Sprint

Un Sprint es una iteración de duración fija aproximadamente de 30 días de calendario consecutivos de trabajo, donde se definen las historias que el equipo desarrollará para poder convertir la Pila de Producto en un incremento potencialmente funcional del producto, y al final de cada Sprint se muestran al cliente las nuevas funcionalidades implementadas. [12].

Una planificación de Sprint produce:

- ✓ Una lista de miembros (y su nivel de dedicación, si no es del 100%).
- ✓ Una Pila de Sprint (lista de historias incluidas en el Sprint).
- ✓ Una fecha concreta para la demostración del Sprint.

En el desarrollo del SAGTRANS el equipo Scrum es de un integrante, con un nivel de dedicación del 90%, debido a que en el mismo tiempo de producción se realizó la documentación de la investigación.

En el primer Sprint planificado se incluyeron 10 historias, seleccionadas por el valor de importancia que tenía cada una, donde suman 22 puntos de historia (días-persona ideales), y estos puntos de historias pueden verse como la velocidad estimada para el Sprint, considerando que una velocidad alta o baja dependerá de varios factores, ya que pueden haber estimaciones iniciales incorrectas, cambios en el alcance, distracciones imprevistas durante el Sprint, etc. El tiempo de duración planificado fue de 30 días calendario, desde el lunes 23 de febrero hasta el lunes 23 marzo, donde al final del Sprint se hace la demostración de las historias implementadas.

Para el segundo Sprint planificado se incluyeron 12 historias que suman 31 puntos de historia (días-persona ideales). El tiempo de duración planificado fue de 30 días calendario, desde el lunes 23 de marzo hasta el martes 21 abril, donde al final del Sprint se hace la demostración de las historias implementadas.

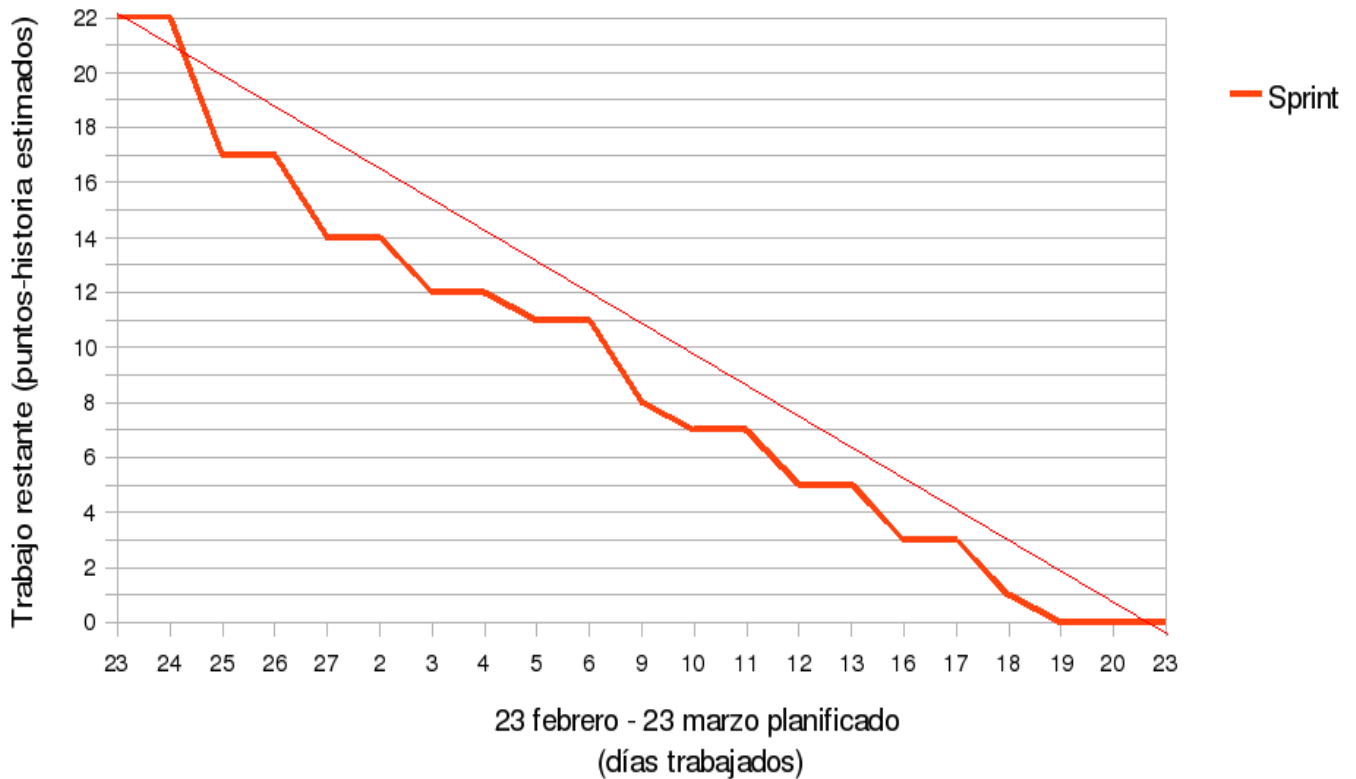
Para ver la planificación de cada Sprint consultar el anexo 3.

2.1.3 Gráfico Burn-Down

Un gráfico Burn-Down demuestra la cantidad de trabajo restante a través del tiempo. Esta es una manera excelente de visualizar la correlación entre la cantidad de trabajo restante en cualquier punto y el progreso del equipo de proyecto en la reducción de este trabajo. La intersección de una línea de la tendencia para el trabajo restante y el eje horizontal indica el punto más probable en el que terminen las actividades. La carta del Burn- Down es la colisión de la realidad (trabajo hecho y cuán rápidamente se está haciendo) con lo planeado, o lo que se espera. [12].

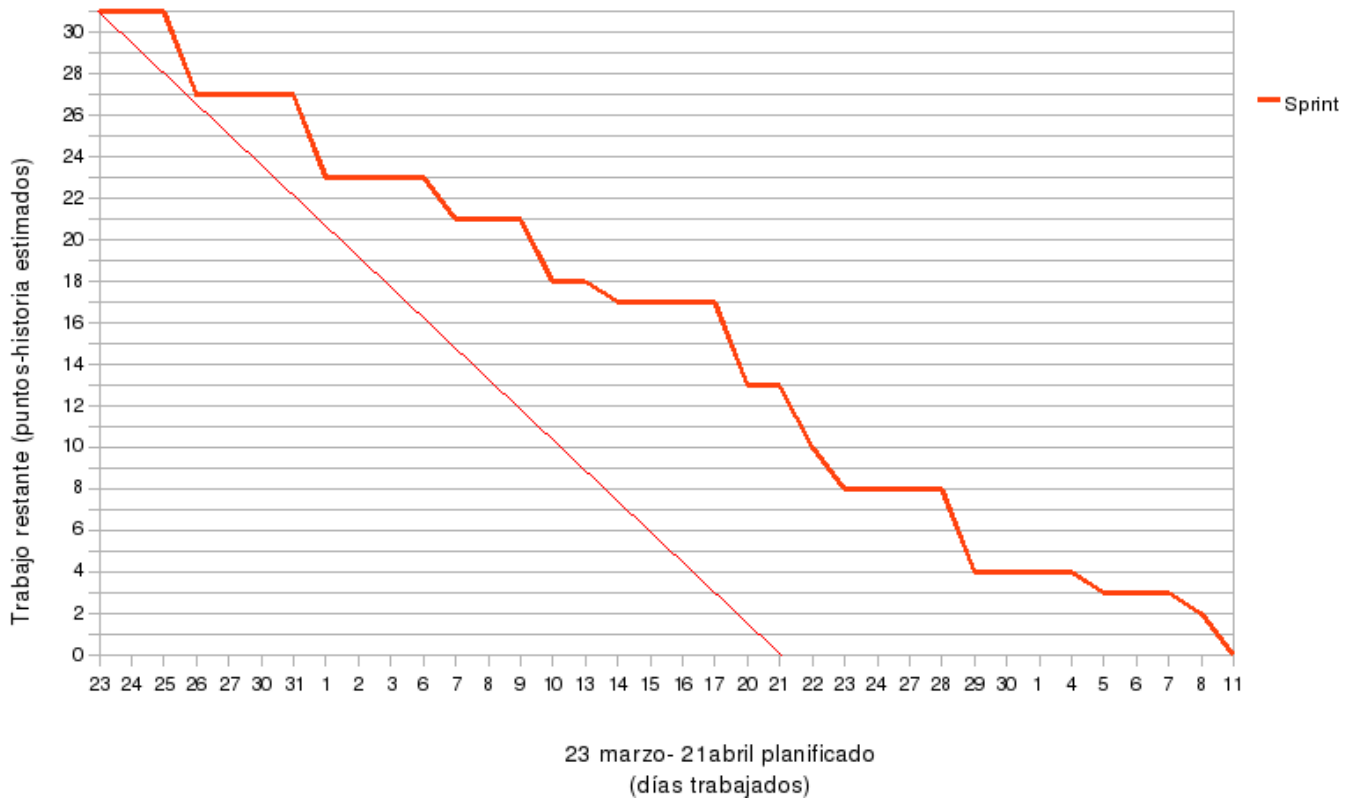
Como se puede apreciar en la figura 1, el gráfico Burn-Down del primer Sprint muestra, en la medida de que se van realizando las historias, el trabajo restante y si va coordinado con lo planificado, además se evidencia que el trabajo estuvo la mayoría del tiempo por debajo de la línea de estimación, lo que es un indicador de que no hubo retrasos en la producción y de que se terminó el Sprint en la fecha planificada.

Figura 1. Burn-Down Primer Sprint



En la figura 2, se muestra el gráfico Burn-Down del segundo Sprint, aquí no ocurre como en la primera iteración, el proceso de producción se retrasó por 11 días de trabajo, en la primera parte el trabajo estuvo aproximadamente coordinado con la línea de estimación, pero a partir de los 18 puntos de historia restantes comenzó a retrasarse, el Sprint se terminó el 11 de mayo. Es importante aclarar que durante el proceso de producción hubo algunos imprevistos, lo que evidentemente contribuyó a este resultado.

Figura 2. Grafico Burn-Down Segundo Sprint



2.2 Especificación detallada del comportamiento del sistema

Requerimientos del sistema

La determinación de requerimientos del software es una descripción completa del comportamiento del sistema a desarrollar. Se dividen en requerimientos funcionales, que son los que el usuario necesita que efectúe el software, y no funcionales (o suplementarios), estos requerimientos imponen restricciones al diseño o funcionamiento del sistema (tal como requerimientos de funcionamiento, estándares de calidad, o requerimientos del diseño) [24].

2.2.1 Requerimientos Funcionales

Los requerimientos que a continuación se enumeran están separados en tres paquetes que agrupan las funcionalidades del sistema SAGTRANS y han facilitado su desarrollo y utilización. Estos no son independientes de los que contiene la Pila de Producto, sino que están

especificados para obtener una mayor claridad y favorecer la implementación del software.

Seguridad

1. Autenticar usuario.
2. Actualizar usuario.
3. Visualizar listado usuario.
4. Actualizar permiso usuario.
5. Visualizar listado permiso usuario.
6. Actualizar grupo usuario.
7. Visualizar listado grupo usuario.
8. Visualizar traza.
9. Insertar traza.
10. Iniciar sesión.
11. Salir del sistema.

Codificadores

12. Visualizar listado modelo de transporte.
13. Actualizar modelo de transporte.
14. Visualizar listado marca de transporte.
15. Actualizar marca de transporte.
16. Visualizar listado tipo de transporte.
17. Actualizar tipo de transporte.
18. *Visualizar listado clase de tipo de transporte.*
19. Actualizar clase de tipo de transporte.
20. *Visualizar listado tipos de pieza.*
21. Actualizar tipo de pieza.
22. *Visualizar listado marcas de pieza.*
23. Actualizar marca de pieza.

Principal

24. Visualizar listado transporte.
25. Actualizar transporte.

26. Crear reporte Inventario General.
27. Crear reporte Inventario Resumido.
28. Crear reporte Vehículos Paralizados.
29. Crear reporte Disponibilidad de Vehículos.
30. Crear reporte Modelo No. 1.
31. Visualizar listado accidente.
32. Actualizar accidente.
33. Crear reporte Accidentes Ocurridos.
34. Visualizar listado pieza.
35. Actualizar pieza.
36. Visualizar listado eventualidad.
37. Actualizar eventualidad.
38. Visualizar listado parqueo.
39. Actualizar parqueo.
40. Visualizar listado informe de explotación.
41. Actualizar informe de explotación.
42. Visualizar listado Ficav.
43. Actualizar Ficav.
44. Visualizar listado Lot.
45. Actualizar Lot.
46. Visualizar listado centro de costo.
47. Actualizar centro de costo.
48. Visualizar listado actividad.
49. Actualizar actividad.
50. Visualizar listado tipos de combustible.
51. Actualizar tipo de combustible.
52. Visualizar listado unidad organizativa.
53. Actualizar unidad organizativa.
54. Visualizar listado trabajador.
55. Actualizar trabajador.

El requerimiento actualizar en todos los casos incluye las acciones, insertar, eliminar y modificar.

2.2.2 Requerimientos no Funcionales

Se determinaron como requerimientos no funcionales para el sistema SAGTRANS los que a continuación se describen:

Apariencia o interfaz externa.

- ✓ No utilizar logotipos, colores o símbolos que contravengan lo establecido en el Manual de Identidad Visual Corporativo de ETECSA.
- ✓ El sistema debe ser asequible para los usuarios pues estos no son expertos en computación.
- ✓ La navegación en el sistema debe ser simple e intuitiva, facilitándole al usuario el acceso a la información.

Usabilidad.

- ✓ Debe ser accesible desde cualquier terminal con acceso a la intranet corporativa de ETECSA.
- ✓ Debe permitir el acceso simultáneo de varios usuarios.
- ✓ El sistema debe de estar funcionando durante el horario laboral.
- ✓ Tiene que contar con una documentación o ayuda para el usuario.

Portabilidad.

- ✓ Podrá tenerse acceso total al código de la aplicación.
- ✓ Debe distribuirse bajo alguna de las licencias de software libre, teniendo como referencia las publicadas en el sitio de la GNU.
- ✓ Soportará, al menos, las plataformas Windows y GNU/Linux.

Seguridad y privacidad.

- ✓ La autenticación de los usuarios en el Sistema se realizará mediante el uso de un usuario y una contraseña, siguiendo lo indicado en el Procedimiento de Administración de Accesos a los Sistemas.
- ✓ Las contraseñas utilizadas deberán cumplir con los requisitos de seguridad que se definen en la Política para la construcción, empleo y gestión de contraseñas en ETECSA.
- ✓ No existe información que se pueda obtener sin ser usuario del sistema por lo que la información manejada por el sistema está protegida de acceso no autorizado y divulgación.
- ✓ Según el rol de cada usuario, deberá estar claramente definido e implementado su entrada al sistema para garantizar que cada usuario tenga acceso únicamente, con los permisos mínimos requeridos, a aquellas funcionalidades que le sean necesarias para desempeñar su labor.
- ✓ El Sistema debe ser capaz de registrar todas las acciones que se realicen.

Software.

En el Cliente

- ✓ Las máquinas deben tener como explorador Web fundamentalmente, el Mozilla FireFox, otro puede ser el Internet Explorer a partir de su versión 6.0.
- ✓ Las máquinas, como una de las funcionalidades del sistema consiste en generar informes en formato PDF, necesitan tener instalado un visor de documentos portátiles (PDF).
- ✓ La resolución de pantalla se recomienda que sea de 1024x768.

En el Servidor

- ✓ La máquina computadora debe tener preferentemente el sistema operativo CentOS, o cualquier distribución de Linux, también puede usar Windows XP o superior.
- ✓ Un Servidor Web Apache, versión 2.2.8 o superior.

- ✓ Un sistema gestor de base de datos PostgreSQL con una versión mayor o igual a la 8.3.
- ✓ PHP versión 5.2.4 o superior.

Hardware.

En el Cliente

- ✓ La frecuencia de procesamiento debe ser de 500 MHz o superior.
- ✓ Debe tener 64 MB de RAM o superior.
- ✓ Las computadoras deben estar conectadas a la red.

En el Servidor

- ✓ La frecuencia de procesamiento debe ser de 1.6 GHz o superior.
- ✓ La máquina computadora debe tener 256 MB de RAM o superior.
- ✓ La máquina computadora servidor debe estar conectada a la red.

2.2.3 Actores del sistema

Los actores del sistema no son parte de este aunque pueden intercambiar información con él, además de tener la facultad de ser recipiente pasivo de información y representar el rol que juega una o varias personas, un equipo o un sistema automatizado. [7].

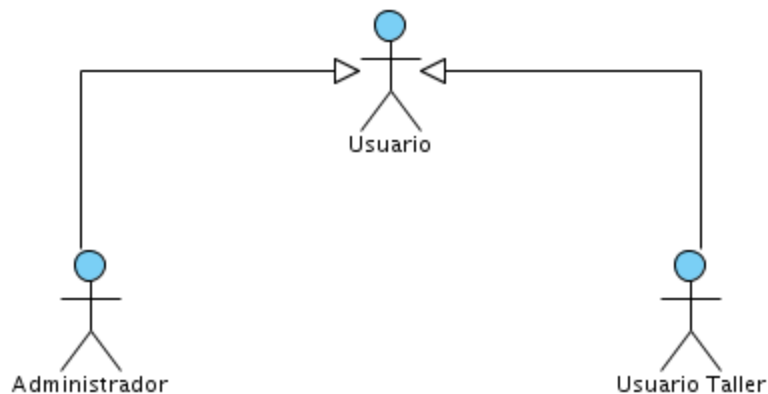
En el sistema SAGTRANS fueron identificados tres actores principales y los casos de uso con los que estos interactúan para dar respuesta a los objetivos propuestos.

Tabla 1. Definición de actores del sistema a automatizar

Actores del sistema	Justificación
Administrador	Representa a el que define y controla el acceso al sistema ubicando a los usuarios en el grupo correspondiente y dándole los

	permisos que le pertenecen. Además registra los codificadores ⁶ .
Usuario del taller	Representa al especialista de transporte que mantiene actualizado el inventario de transporte y genera todos los reportes.
Usuario	Es la generalización de los actores Administrador y Usuario del taller. Representa el rol de cualquier usuario para entrar, salir y visualizar la información del sistema.

Figura 3. Diagrama de Actores del Sistema



⁶ Codificadores: una información estandarizada que codifica otra información propia del sistema.

Figura 5. Diagrama de Casos de Uso del paquete codificadores

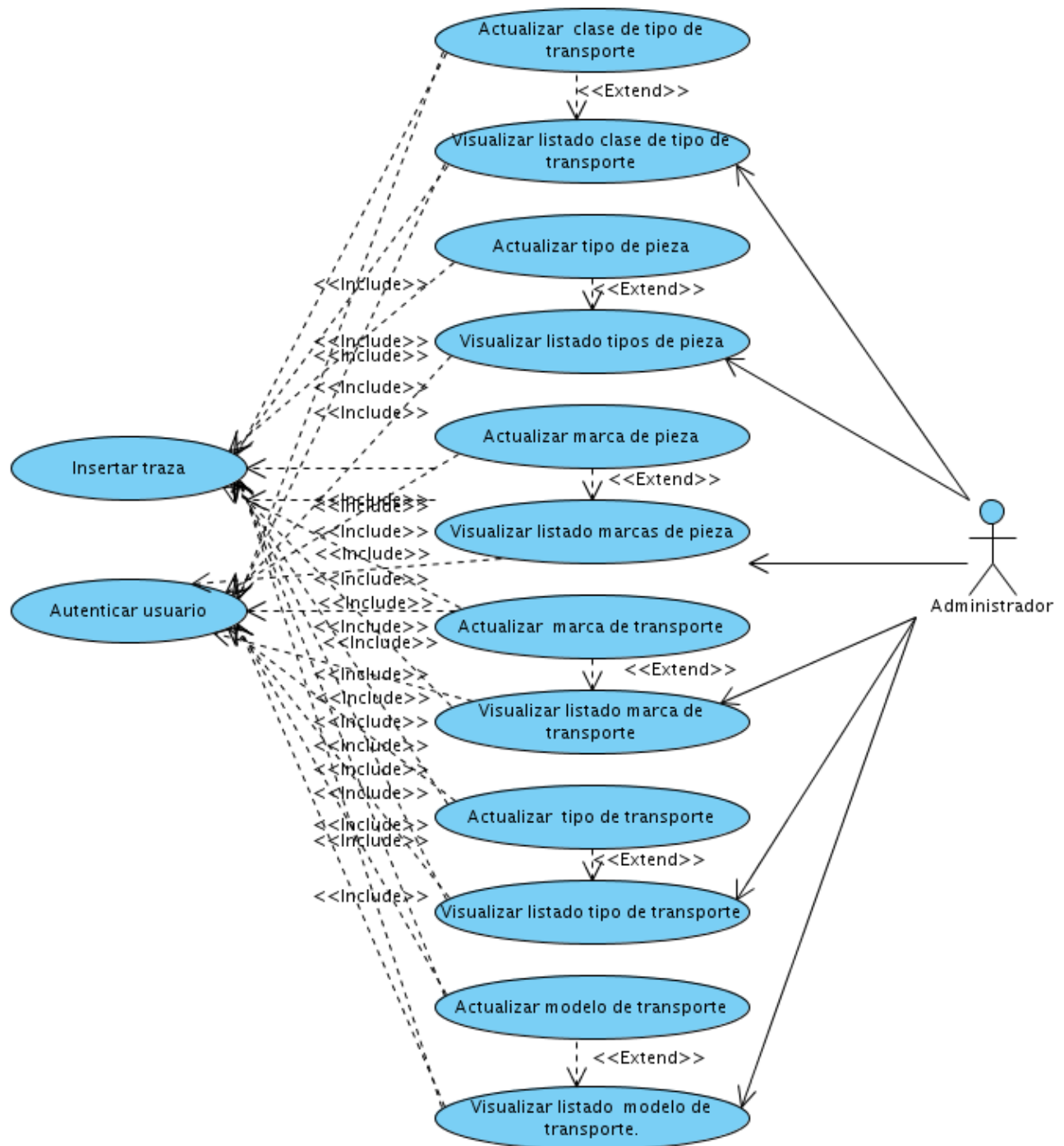
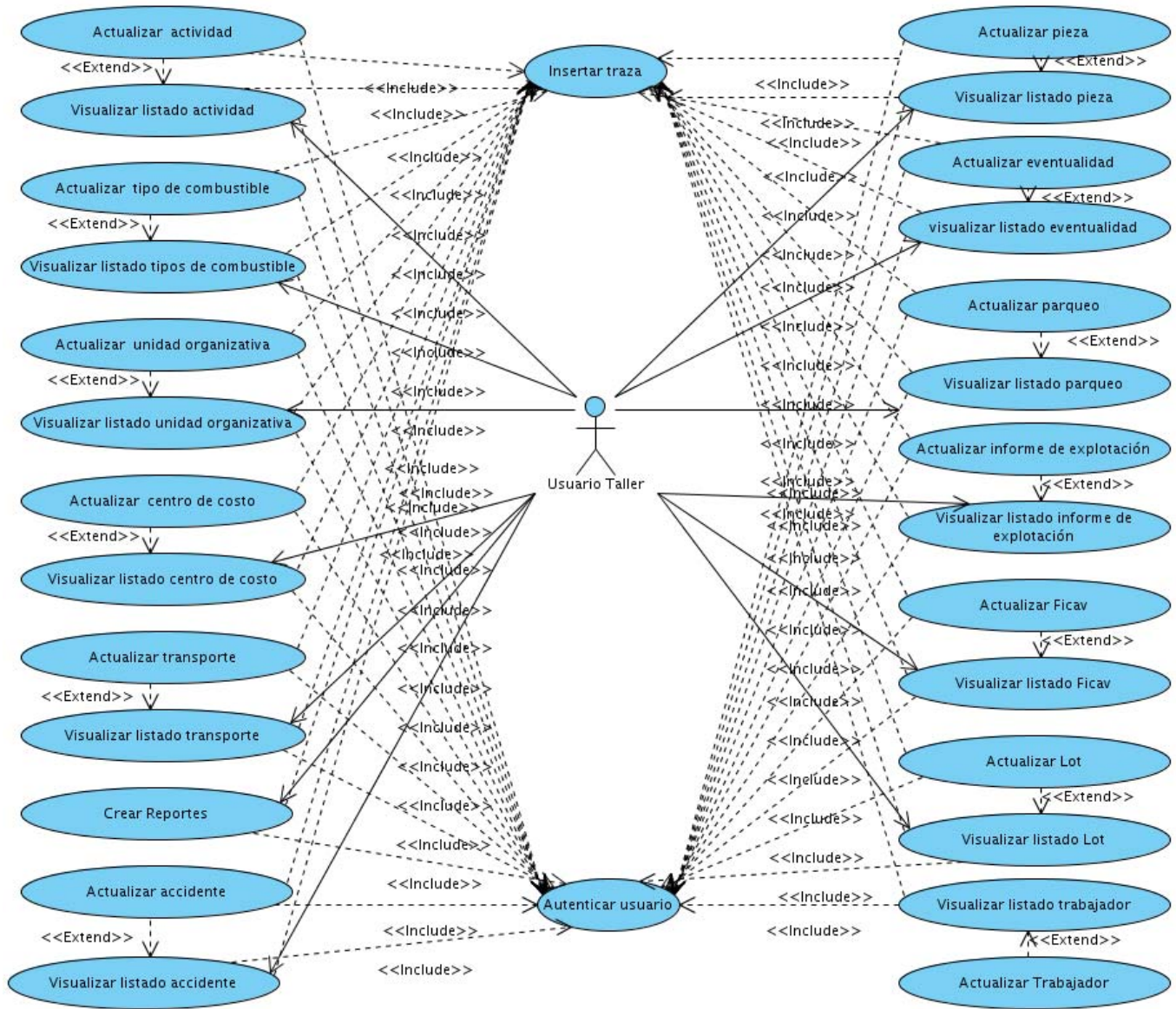


Figura 6. Diagrama de Casos de Uso del paquete principal

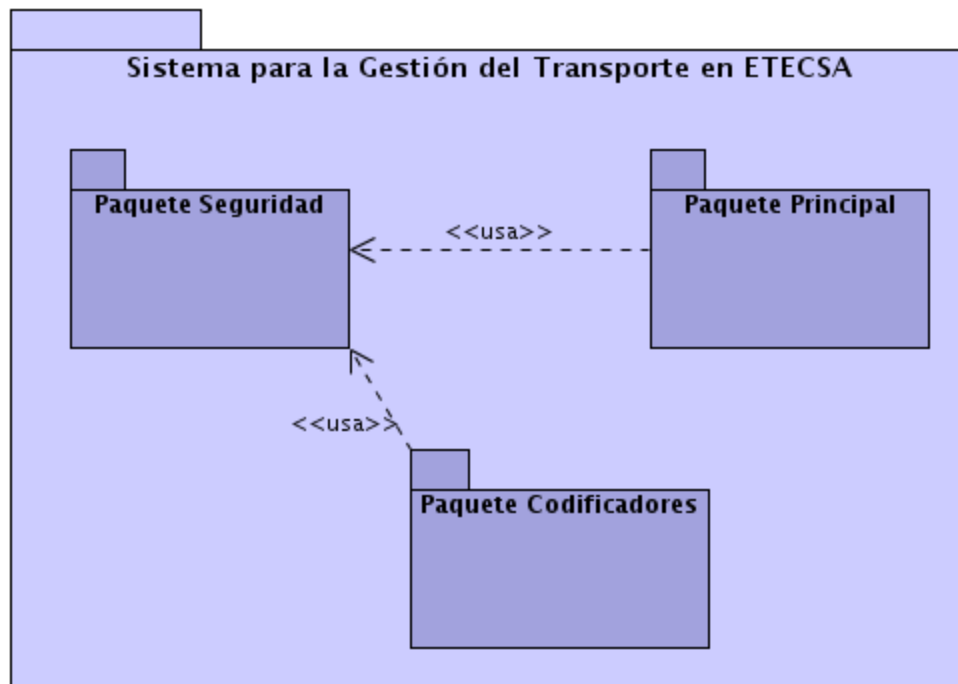


En el diagrama de casos de uso del paquete principal, el caso de uso crear reportes, engloba

todos los reportes enunciados en los requerimientos funcionales.

2.2.6 Paquetes y sus relaciones

El siguiente diagrama de paquetes muestra como el sistema está dividido en tres agrupaciones lógicas mostrando las dependencias entre esas agrupaciones, organizados para maximizar la coherencia interna dentro de cada paquete.



2.2.7 Diagrama de Clases del Diseño de los casos de uso fundamentales

Con el objetivo de profundizar en los aspectos relacionados con los requisitos funcionales y no funcionales, y restricciones relacionadas con lenguajes de programación, sistemas operativos, tecnologías de interfaz de usuario y componentes reutilizables, se realizaron los diagramas de clases del diseño para los casos de uso que se consideraron más importantes.

Estos diagramas de clases describen gráficamente las especificaciones de las clases de

software, de las interfaces, así como sus relaciones en la aplicación. Las clases incluyen definiciones para atributos y operaciones, son una construcción similar en la implementación del sistema, es decir, se pueden establecer comparaciones entre los componentes de las clases del diseño y los de la aplicación. [26].

A continuación se mencionan los estereotipos que UML usa para los elementos más comunes que conforman la tecnología Web, como las páginas y los formularios.

<<Server Page>> Representa la página Web que tiene código que se ejecuta en el servidor.

<<Client Page>> Una instancia de Página Cliente es una página Web, con formato HTML. Son interpretadas por el navegador.

<<Html Form>> Colección de elementos de entrada que son parte de un página cliente. Sus atributos son los elementos de entrada del formulario.

Los diagramas de los escenarios de los casos de uso más importantes pueden ser consultados en el anexo 4.

2.2.8 Diagrama de Clases Persistentes del Diseño

Se realizó con el propósito de proveer un diseño conceptual de la información que se manejará en el sistema, y sus relaciones internas que garantizan su funcionamiento, un diagrama de clases persistentes del diseño.

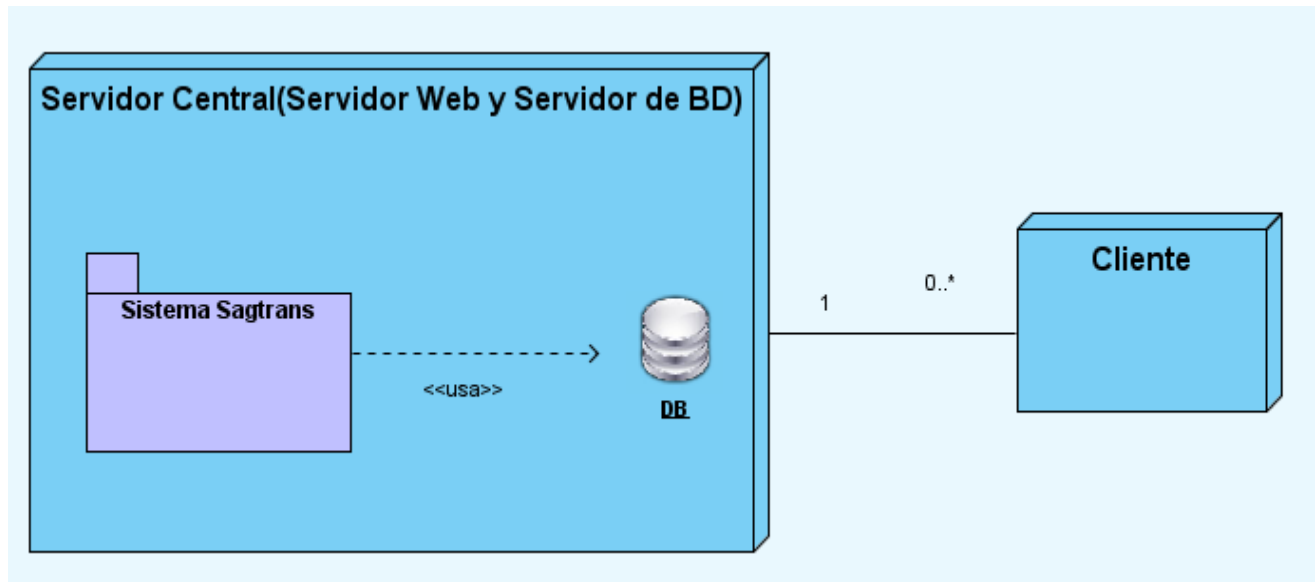
Las clases persistentes son aquellas capaces de mantener su valor en el espacio y el tiempo. Son de vital importancia en los diagramas de clases del diseño pues en su interacción con las clases servidoras definen en la aplicación todos los accesos físicos que se realizarán a la BD. [7].

Se definieron 29 clases persistentes para almacenar la información del sistema SAGTRANS en la base de datos, las relaciones y atributos del diagrama del modelo de datos y el diagrama

de clases persistentes, puede ser consultado digitalmente en los ficheros que se adjuntan al documento.

2.2.9 Diagrama de Despliegue

El Diagrama de Despliegue que se muestra a continuación describe la distribución física del sistema en dependencia de cómo se distribuye la funcionalidad entre los nodos de cómputo, además de contener los nodos que forman la topología hardware sobre la que se ejecuta el sistema y la distribución de las partes del sistema en ellos. Para el sistema SAGTRANS un nodo de cómputo sería el Servidor Central que en el mismo están contenidos el Servidor Web y el Servidor de Base de Datos, y otro nodo sería el Cliente, donde ninguno o varios clientes pueden estar conectados al mismo tiempo al Servidor.



2.2.10 Valoración de sostenibilidad de la propuesta

Debido a que el producto informático tiene un impacto sobre aquellos que hacen uso del mismo, se hace necesario el estudio de su efecto en las dimensiones administrativa, socio-humanista, ambiental y tecnológica. Teniendo en cuenta las características del sistema

propuesto se determinará si es sostenible en el tiempo.

Dimensión Administrativa

Todo producto informático tiene asociado un costo, para ello se tienen en cuenta una serie de parámetros que son de vital importancia para la cuantificación. Algunos de estos son: complejidad y envergadura del proyecto, el tiempo de desarrollo, el personal involucrado y los beneficios que el mismo reporta a la sociedad.

Para el análisis de la factibilidad económica del Sistema SAGTRANS en La Empresa de Telecomunicaciones de Cuba S.A., se utilizó el Modelo Constructivo de Costos (COCOMO, por sus siglas en inglés).

El COCOMO tiene como objetivo la estimación del costo asociado al desarrollo de un software. Utiliza los puntos de función para calcular el tamaño de un producto de software y el esfuerzo asociado a su desarrollo. El empleo de COCOMO permite realizar una estimación de la cantidad de personas que se requieren utilizar para el desarrollo del sistema, así como el tiempo de desarrollo aproximado.

Los puntos de función son ponderados mediante un factor de complejidad de tres niveles (simple, medio y complejo). A continuación se muestra el estudio realizado de la factibilidad económica para el sistema SAGTRANS.

Las tablas con las entradas externas, consultas externas y ficheros lógicos internos, asociadas a los requisitos funcionales del sistema SAGTRANS y utilizadas para el cálculo de los puntos de función, pueden ser consultadas en el Anexo 5.

Los elementos que se tienen en cuenta para calcular el total de puntos de función desajustados se encuentra en la Tabla 2.

Tabla 2. Puntos de función desajustados

Puntos de Función Desajustado (UFP)							
Características	Baja		Media		Alta		Subtotal
	N	Peso	N	Peso	N	Peso	
EI	22	3	1	4	1	6	76
ILF	31	7	0	10	0	15	217
EQ	21	3	0	4	2	6	75
EO	0	4	0	5	1	7	7
Total							375

El cálculo de las instrucciones fuentes, se basa en la cantidad de instrucciones fuentes por punto de función que genera el lenguaje de programación empleado.

Para desarrollar el sistema se utilizó el lenguaje de programación PHP. El uso del PHP es de un 100 % del código utilizado. Los valores de las instrucciones fuentes por los lenguajes utilizados se muestran en la Tabla 3.

Tabla 3. Instrucciones fuentes para el lenguaje utilizado

Cantidad de Instrucciones Fuentes	
Elementos	Valor
UFP	375
Ratio lenguaje PHP	35
Instrucciones fuentes	13125
Reducción 40%	7875
Size	7.875

$$\text{SLOC} = \text{UFP (PHP)} * \text{ratio (PHP)}$$

El tamaño estimado en miles de instrucciones fuentes se denomina Size y es de 7,88.

Teniendo en cuenta que el framework seleccionado para el desarrollo del sistema contribuye con técnicas visuales al desarrollo de la Aplicación Web y considerando que esto facilita el trabajo, se aplicó una reducción del 40 % de las instrucciones fuentes, obteniéndose:

El esfuerzo asociado al desarrollo del sistema se denomina PM. Para calcular PM se utilizan los multiplicadores de esfuerzo, los factores de escala y una serie de valores calibrados (A, B, C, D).

Tabla 4. Multiplicadores de esfuerzo

Multiplicadores de Esfuerzo	
Multiplicador	Valor
RCPX	1
RUSE	1
PDIF	1
PERS	1
PREX	1.33

FCIL	0.87
SCED	1
Productoria	1.1571

Tabla 5. Factores de escala

Factores de Escala	
Factor	Valor
PREC	6.2
FLEX	2.03
RESL	1.41
TEAM	1.1
PMAT	3.12
Total	13.86

Tabla 6. Valores calibrados

Valores Calibrados	
A	2.94
B	0.91
C	3.67
D	0.28

$$PM = A * Size^E \prod_{i=1}^5 EM_i$$

E depende de los factores de escala (SF):

$$E = B + 0.01 * \sum_{j=1}^5 SF_j$$

El esfuerzo asociado al sistema es de 30 hombres mes.

$$TDEV = C * PM^F$$

$$\text{Donde } F = D + 0.2 * (E - B)$$

$$TDEV = PM/CHR = 10.50$$

$$CH = PM / TDEV = 2,82$$

El desarrollo de la Aplicación Web SAGTRANS requiere aproximadamente 11 meses para realizarlo con 3 personas.

Como se cuenta con 1 desarrollador (CHR=1), la aplicación se termina en aproximadamente 30 meses.

El salario medio de un analista se considera que es \$225 y se denomina SP. Se denomina CHM costo por Hombre Mes. El costo total del desarrollo del sistema propuesto se determina por:

$$CHM = 1 * SP = \$225$$

$$\text{Costo} = CHM * TDEV = \$ 6 663.60$$

El costo total del proyecto asciende a \$ 6 663.60.

Este trabajo aporta beneficios, no incurriendo en ingresos directamente, sino en cuanto a que la entidad no tiene que mejorar el equipamiento dentro de los requerimientos. Además, permite ahorro de recursos humanos e insumos, tales como materiales de oficinas, papel, tinta, cintas de impresoras, etc.

El sistema mejora significativamente el proceso de gestión de las actividades de transporte pues la información puede ser consultada por el personal involucrado en el proceso y los directivos de la entidad en una fecha más temprana, contando con un tiempo mayor para discutir los datos en las entidades superiores.

Se ahorra tiempo de trabajo, porque la creación de informes es algo fácil y rápido. Al ahorrarse tiempo, también se ahorra electricidad, ya que es menor el tiempo que se necesita para realizar el trabajo, por lo que además es un ahorro de recursos para el país.

Dimensión Socio-Humanista

Para el ser humano las condiciones del ambiente de trabajo tienen una gran influencia en su

vida, ya que tienen una estrecha relación el rendimiento de las personas en las tareas que ellos realizan con las comodidades que se les proporcionan. El sistema no genera o disminuye empleos, pero se encarga de proveer mejoras a las condiciones de trabajo de las personas relacionadas directamente con el proceso de gestión del transporte en la DTHO, y presenta las funcionalidades necesarias para satisfacer las expectativas de los especialistas de transporte.

Dimensión Ambiental

La interfaz que el sistema brinda al usuario resulta agradable, cuidando el uso de los colores, los cuales están en la gama de los azules y no resultan agresivos a la vista. No son colores con demasiado brillo, ni son colores oscuros, sino más bien colores frescos, que no alteran al usuario, sino que le dan una sensación de relajamiento y tranquilidad. El mismo está diseñado para facilitar la actividad de los usuarios y disminuir el tiempo de estancia frente a la computadora evitando con ello los problemas de salud provocados por la mala postura y el desgaste visual, lo que contribuye a aumentar la calidad de vida de los trabajadores.

Dimensión Tecnológica

La Dirección Territorial de ETECSA en Holguín, cuenta con los medios y la infraestructura electrónica necesarios para la correcta implantación del sistema. Se hace necesaria en cierta medida la capacitación del personal que interactuará con la Aplicación Web, con el uso del manual de usuario y la Ayuda será suficiente.

Existe una estrecha comunicación entre el desarrollador de la aplicación y los interesados, logrando que el sistema desarrollado satisfaga las necesidades y expectativas de los clientes finales. Los códigos están debidamente comentados, lo que facilitará su entendimiento en caso de ser necesario un mantenimiento futuro. El sistema podrá ser vulnerable si el servidor de la aplicación sufre alguna avería, es por esto que se recomienda que se realicen copias tanto de la BD como de la aplicación en general en otra computadora, para evitar que se pierda el sistema o parte de este. Luego de este análisis se puede plantear que el sistema es sostenible en el tiempo.

22.3 Valoración de los resultados obtenidos

Una vez terminado el sistema informático SAGTRANS se realizó la valoración del mismo a

través de encuestas realizadas a los especialistas, en el procesamiento de las encuestas se usó el método Delphy, el cual se basa en la utilización sistemática del juicio intuitivo de un grupo de usuarios para obtener un consenso de opiniones informadas o también en la utilización sistemática del criterio práctico para conseguir un consenso de opiniones sobre la posible aplicabilidad de una propuesta determinada. [27]

En la encuesta aplicada (ver anexo 6) se ponen en consideración los aspectos fundamentales del sistema con el objetivo de comprobar su aporte a la entidad y a la gestión de la información de la actividad de transporte, la pregunta número uno de la encuesta fue procesada por el método Delphy (ver anexo 7), donde se tiene en cuenta:

1. Utilidad del sistema informático.
2. Cumplimiento de las expectativas del usuario con el sistema informático.
3. Qué tanto se mejoró la integridad de la información.
4. Uso de las imágenes y los colores.
5. Qué tan “amigable” resultó el diseño de las interfaces del sistema.
6. Valoración de la organización de la información.
7. Eficiencia de los procesos que se realizaban de forma manual anteriormente.
8. Qué tanto apoya el sistema la gestión de la información.
9. Qué tan útil es la ayuda suministrada.

Luego de obtener el resultado procesado por el método Delphy, se concluye que los encuestados concuerdan en que todos los aspectos son “muy adecuados”.

CONCLUSIONES GENERALES					
Criterios	MA	BA	A	PA	NO
¿Cómo valora la utilidad del sistema informático?	Si	-	-	-	-

¿Cómo cree que fueron cumplidas sus expectativas con la implementación?	Si	-	-	-	-
¿Qué tanto mejora la integridad de la información?	Si	-	-	-	-
¿Cómo fue el uso de las imágenes y los colores?	Si	-	-	-	-
¿Qué tan “amigable” le resultó el diseño de las interfaces del sistema?	Si	-	-	-	-
¿Cómo valora la organización de la información?	Si	-	-	-	-
¿Cómo considera que se mejoró la eficiencia de los procesos que se realizaban de forma manual?	Si	-	-	-	-
¿Qué tanto apoya el sistema la gestión de la información?	Si	-	-	-	-
¿Qué tan útil es la ayuda suministrada?	Si	-	-	-	-

En el caso de la pregunta dos, se emitieron opiniones favorables que elogian la integridad y organización de la información, el uso de los colores y la facilidad de uso que tiene el sistema. En cuanto a la pregunta tres, los encuestados hicieron algunas sugerencias de funcionalidades que se le podría añadir a otras versiones futuras del SAGTRANS con el objetivo mejorar la gestión de la información del sistema, algunas de las cuales se contemplan en las recomendaciones.

Los resultados obtenidos a partir de las encuestas realizadas, constituyen un argumento convincente del grado positivo en que el sistema informático apoya y favorece la calidad de la gestión referente a la información del transporte en ETECSA.

2.4 Conclusiones

En este capítulo se obtuvieron los requerimientos para el sistema, siguiendo el flujo de trabajo propuesto por la metodología Scrum y haciendo uso de los esquemas UML necesarios, lo que finalmente produjo la estructura del SAGTRANS y el funcionamiento de la aplicación.

El producto informático resultante, teniendo en cuenta sus características, queda clasificado como sostenible en el tiempo, además de considerarse factible la solución empleada. De igual forma los resultados positivos obtenidos en la aplicación de encuestas, demuestran la satisfacción de los usuarios y el cumplimiento de sus expectativas con respecto al sistema

informático, resolviendo así el problema definido en esta investigación.

Conclusiones Generales

Con el desarrollo del sistema propuesto, se ha dado cumplimiento al objetivo de esta investigación, pues como resultado se obtuvo un sistema informático, que garantiza una mayor integridad en el control interno de la actividad de transporte e independencia del marco tecnológico elegido, además de ser eficiente, rápido, y de proporcionar una mayor confiabilidad, usabilidad y disponibilidad de los datos.

Además se puede concluir que:

- ✓ La propuesta que se presenta es capaz de brindar al especialista de transporte, durante el cumplimiento y desarrollo de su trabajo, una mayor calidad y un mejor control de los datos.
- ✓ El empleo del framework Symfony teniendo como base el lenguaje de programación Web PHP favoreció en grado superlativo el desarrollo del sistema SAGTRANS, permitiendo un mayor énfasis en la implementación de los requerimientos del software y no en detalles de bajo nivel para proveer un sistema funcional, que ya vienen soportados por el framework.
- ✓ La metodología utilizada durante el desarrollo de software resultó ser eficiente, ya que debido a sus características, agilizó el proceso de construcción del sistema, proporcionando un mayor aprovechamiento del tiempo con que se contaba y

potenciando sus funcionalidades.

- ✓ El análisis realizado desde el impacto social, ambiental, tecnológico y administrativo muestra un producto sostenible que no solo eleva la eficiencia y satisfacción de los trabajadores sino que se adapta a nuevos cambios para evolucionar en el futuro.

La hipótesis enunciada al inicio de esta investigación queda demostrada a través de los argumentos expuestos en el presente documento, y se puede arribar a la conclusión de que el sistema propuesto logra favorecer el proceso de gestión de la información correspondiente a la actividad del transporte en la empresa de ETECSA.

Recomendaciones

Una vez concluida esta investigación se proponen las siguientes recomendaciones:

- ✓ Confeccionar un plan de capacitación para los usuarios que usarán la herramienta, posibilitando aprovechar al máximo las facilidades que brinda.
- ✓ Incorporar al sistema propuesto dos nuevos módulos, uno relacionado con las órdenes de trabajo en el taller de transporte para solucionar las interrupciones, y otro que gestione un registro histórico de los transportes pertenecientes a la empresa que se han dado de baja.

Referencia Bibliográfica

[1] “Estrategia para el uso del Software Libre en Cuba”. http://www.ispcmw.rimed.cu/sitios/digbiblio/cont/EI/SO_Linux/SoftLibre/EstSoftLibre.htm

Consultado el (10/01/2009).

[2] Gámez Torrecilla Leisy, Calderón Rodríguez Víctor. “Migración de Windows Active Directory hacia un servidor de directorio Open Source en Linux”. Consultado el (10/01/2009).

[3] Batista Soler Otto, Rodríguez Figueredo Héctor. “Análisis jurídico del software libre. sus particularidades en cuba”. (10/06/2006).

[4] Sæther Bakken Stig, Aulbach Alexander, “Manual de PHP”. (10/06/2006).

[5] “FAQ de PostgreSQL”. Actualizado domingo, noviembre 20 2005.

[6] Letelier Patricio, Penadés Carmen. “Metodologías ágiles para el desarrollo de software: eXtreme Programming (XP)”. 2009

[7] Jacobson, Ivar. “El Proceso Unificado de Desarrollo de Software”. 2000

[8] Kniberg Henrik, "Scrum y XP desde las trincheras". 2007

[9] Lockhart Thomas. "Tutorial de PostgreSQL". (01/10/1998).

[10] Potencier Fabien, Zaninotto Francois. "Symfony 1.2, la guía definitiva". (30/12/2008).

[11] "Metodología para el Estudio Organizativo de los Procesos Empresariales". Instrucción No 01 de 2006 de la Dirección de Organización y Procesos de ETECSA.

[12] Henrik Kniberg. "Scrum y XP desde las trincheras". (2007).

[13] Bain Malcom, Gallego Rodríguez Manuel. "Aspectos legales y de explotación del software libre. Parte I Primera edición: marzo (2004).

[14] "Políticas para la Prestación de los Servicios de Transporte en la Empresa de Telecomunicaciones de Cuba S.A.". Resolución No 06 de 2006 de la Dirección de Servicios Generales de ETECSA.

[15] "Aplicación Web". http://es.wikipedia.org/wiki/Aplicación_Web Consultado el (06/04/2009).

[16]. "Framework". <http://es.wikipedia.org/wiki/Framework> Consultado el (06/04/2009).

[17] "Cliente-servidor". http://es.wikipedia.org/wiki/Cliente_servidor Consultado el (06/04/2009).

[18] "Sistema de gestión de base de datos". http://es.wikipedia.org/wiki/Sistema_de_gestión_de_base_de_datos Consultado el (06/04/2009).

[19] "La definición de Software Libre – Proyecto GNU – Fundación para el Software Libre

(FSF)". <http://www.gnu.org/philosophy/free-sw.es.html> Consultado el (07/04/2009).

[20] "Software libre". [http://es.wikipedia.org/w/index.php?title=Software libre](http://es.wikipedia.org/w/index.php?title=Software+libre) Consultado el (07/04/2009).

[21] "Servidor HTTP Apache". [http://es.wikipedia.org/wiki/Apache http server](http://es.wikipedia.org/wiki/Apache_http_server) Consultado el (07/04/2009).

[22] "PHP". [http://es.wikipedia.org/wiki/Apache http server](http://es.wikipedia.org/wiki/Apache_http_server) Consultado el (07/04/2009).

[23] "Lenguaje Unificado de Modelado (UML)". <http://es.wikipedia.org/wiki/UML> Consultado el (14/04/2009).

[24] "Ingeniería de requerimientos". [http://es.wikipedia.org/wiki/Ingenieria de requerimientos](http://es.wikipedia.org/wiki/Ingenieria_de_requerimientos) Consultado el (04/05/2009).

[25] "Información Corporativa". <http://www.ETECSA.cu/infocorporativa.asp?codigo=19&padre=19> Consultado el (12/05/2009).

[26] G. Irlán. "Sistema de Estudiantes". Trabajo de Diploma para optar por el título de Ingeniero Informático, Universidad de Holguín "Oscar Lucero Moya", Ciudad de Holguín, junio 2006.

[27] Rodríguez, Félix. "El método Delphy para el procesamiento de los resultados de encuestas a expertos o usuarios en estudios de mercado y en la investigación educacional".

Bibliografía

Clemente Laboreo Daniel, “Cómo hacer buenas páginas Web”. (Septiembre 2004).

Eguíluz Pérez Javier, “Introducción a CSS “. (17/10/2008).

Eguíluz Pérez Javier, “Introducción a XHTML”. (17/12/2008).

Gutmans Andi, Sæther Bakken Stig. “PHP 5 Power Programming”. (Octubre 2004).

José Manuel Alarcón Medina, “Administración SGBD PostgreSQL” . (Octubre 2006).

M. Stallman Richard, “Software libre para una sociedad libre”. (2004).

Novoa Hernández Pavel. “Aplicación Web para el Presupuesto de Ventas de la empresa Comandante Ernesto Che Guevara”. Trabajo de Diploma para optar por el título de Ingeniero Informático, Universidad de Holguín “Oscar Lucero Moya”, Ciudad de Holguín, junio 2007.

Potencier Fabien, “Los formularios de Symfony 1.2”. (2009).

Potencier Fabien, "El tutorial Jobeet ". (01/01/2009).

Reyes Pupo Oscar. "Mantenimiento del Sistema Informático TI". Trabajo de Diploma para optar por el título de Ingeniero Informático, Universidad de Holguín "Oscar Lucero Moya", Ciudad de Holguín, junio 2008.

Glosario de Términos

SAGTRANS: Sistema para la Gestión del Transporte en ETECSA.

ETECSA: Empresa de Telecomunicaciones de Cuba S.A.

DTHO: Dirección Territorial de Holguín.

Soberanía Tecnológica: Dominio nacional del diseño y en su caso, de la fabricación de los equipos y componentes que sirven de soporte a las TIC, así como de los programas informáticos asociados, para alcanzar gradualmente la fiabilidad, estabilidad, seguridad e invulnerabilidad de las tecnologías de la información y las comunicaciones.

Framework: en el desarrollo de software, es una estructura de soporte definida mediante la cual otro proyecto de software puede ser organizado y desarrollado.

Hardware: Conjunto de elementos materiales que componen un ordenador. En dicho conjunto se incluyen los dispositivos electrónicos y electromecánicos, circuitos, cables, tarjetas, armarios o cajas, periféricos de todo tipo y otros elementos físicos.

Software: Conjunto de programas que puede ejecutar el hardware para la realización de las tareas de computación a las que se destina. Es el conjunto de instrucciones que permite la utilización del equipo.

Internet: Red de redes a escala mundial de millones de computadoras interconectadas con el conjunto de protocolos TCP/IP.

Página Web: Es un documento escrito en formato HTML que es visible en un navegador.

HTML: Es el Lenguaje de Marcado de Hipertexto (HyperText Markup Language), que se utiliza para crear los documentos a los que se accede a través de navegadores WWW.

Protocolo: Es una serie de reglas que utilizan dos computadoras para comunicarse entre sí. Cualquier producto que utilice un protocolo dado puede funcionar con otros productos que utilicen el mismo protocolo.

HTTP: Es el Protocolo de Transferencia de Hipertexto, un protocolo Web que controla las peticiones y servicios de documentos HTML.

TCP/IP: Protocolo de Control de Transmisión / Protocolo de Internet (Transmission Control Protocol/Internet Protocol), es el conjunto de protocolos que rigen todas las comunicaciones entre todas las computadoras en Internet.

Autenticación: Proceso mediante el cual se comprueba la identidad de un usuario en la red.

LDAP: Servicio de directorio ordenado y distribuido para buscar diversa información en un entorno de red. LDAP puede considerarse una base de datos (aunque su sistema de almacenamiento puede ser otro diferente) al que pueden realizarse consultas.

Interfaz: Parte de un programa informático que permite a éste comunicarse con el usuario o con otras aplicaciones permitiendo el flujo de información.

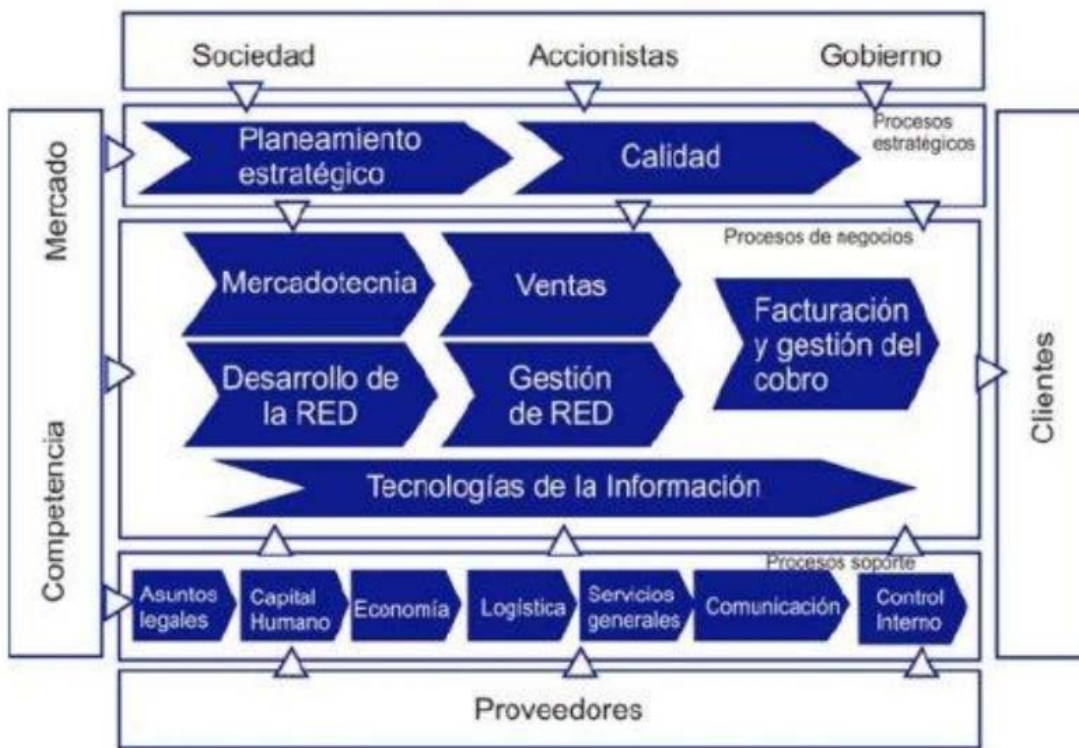
Sistema Operativo: Conjunto de software que controla los distintos recursos del ordenador.

BD: Base de Datos.

SQL: Lenguaje de Consultas Estructurado (Structured Query Language). Lenguaje de consultas y programación de Base de Datos. Se utiliza ampliamente para tener acceso a datos, consultar, actualizar y administrar sistemas de bases de datos relacionales.

Anexos

Anexo No 1. Mapa General de Procesos de ETECSA



Anexo No 2. Pila de Producto

Tabla 7. Definición de la Pila de Producto

ID	Nombre	Imp.	Est.	Como Probarlo	Notas
1	Como Jefe Técnico, mantener actualizada una base de datos con el inventario del transporte.	900	5	Insertar un nuevo transporte. Verificar si se insertó. Actualizar los datos de un transporte. Verificar su inserción y actualización.	El gestor de BD que se utilice debe ser el PostreSQL.
2	Como Administrador del Sistema, registrar los codificadores que se emplean.	850	3	Entrar. Insertar un nuevo codificador. Verificar su inserción y actualización.	
3	El sistema debe funcionar después de los cambios que se están haciendo en la Empresa.	800	2	Debe correr igual en Windows XP y en LINUX.	Especial atención al funcionamiento en GNU/Linux, particularmente en Ubuntu y CentOS.
4	Como Administrador del Sistema, controlar el acceso del resto de los usuarios a las funcionalidades del sistema.	780	1	Entrar. Crear un usuario con determinados privilegios. Salir. Entrar con el usuario creado. Verificar que el usuario accede a las funcionalidades definidas.	El Dueño del Producto debe definir los accesos a controlar por el Administrador del Sistema.
5	Como Jefe	750	3	Entrar. Insertar una nueva	Debe mantenerse

	Técnico, mantener un control de las piezas del transporte.			pieza en un transporte. Verificar su inserción y actualización.	un control de la fecha en que se puso o se cambió la pieza.
6	Acceder al sistema desde cualquier punto de la red a través de la Web.	700	1	Debe funcionar igual usando los navegadores MS, Internet Explorer y Mozilla Firefox.	Optimizar funcionamiento para el Mozilla Firefox, fundamentalmente.
7	Para trabajar en el sistema se debe iniciar sesión usando un usuario y una contraseña.	680	2	Verificar que se puede entrar al sistema con uno de los usuarios creados. Verificar que no se puede acceder con un usuario inexistente.	Autenticación usando protocolo seguro SSL, contra un servidor LDAP. La contraseña debe cumplir los parámetros de seguridad establecidos en la Empresa.
8	Como Usuario del Taller, controlar las interrupciones del transporte.	650	2	Entrar. Registrar una nueva interrupción sobre un transporte. Ir a la página de disponibilidad del transporte. Comprobar que el transporte aparece como no disponible.	
9	Como Usuario del Taller, controlar los mantenimientos del transporte.	645	2	Entrar. Registrar la entrada a mantenimiento de un transporte. Ir a la página de disponibilidad del transporte. Comprobar que el transporte aparece como no	

				disponible.	
10	Disponer de una estructura de filtrado por unidades organizativas, que permita la navegación por niveles.	643	1	Entrar. Ir a la página del inventario del transporte. Filtrar para una unidad organizativa. Comprobar que sólo se muestran los datos de esa unidad organizativa.	
11	Como Jefe Técnico, obtener reportes de la disponibilidad del transporte que incluyan, si no está disponible, la causa y la demora en la solución.	640	4	Entrar. Ir a la página de disponibilidad del transporte. Obtener un reporte y verificar que muestre los datos necesarios correctamente.	
12	Como Jefe Técnico, obtener reportes del nivel de explotación del transporte.	630	4	Entrar. Ir a la página de nivel de explotación del transporte. Obtener un reporte y verificar que muestre los datos necesarios correctamente.	
13	El sistema debe ser auditable.	550	2	Entrar. Modificar los datos de un transporte. Ir a página de auditoría y comprobar que se registró la fecha, la hora, la acción y quién la realizó	Cumplir con lo establecido por la Empresa
14	Como Jefe Técnico, registrar los	400	3	Entrar. Seleccionar un transporte y registrar un	

	accidentes del transporte.			accidente sobre este. Ir a la página de reportes de accidentes. Comprobar que aparece el accidente registrado.	
15	Como Jefe Técnico, obtener el listado de transportes por modelos, tipos, marcas, tipos de combustible.	390	1		
16	Como Jefe Técnico, obtener reportes de los accidentes ocurridos sobre el transporte en un período de tiempo	380	4	Entrar. Registrar un accidente. Ir a la página de reportes de accidentes. Verificar su inserción y actualización.	
17	Como Jefe Técnico, mantener actualizada la información de los parqueos autorizados	200	3	Entrar. Insertar un nuevo parqueo. Comprobar su inserción y actualización.	
18	Varios usuarios deben poder trabajar sobre el sistema simultáneamente	150	2		
19	Como usuario con privilegios de	100	4	Entrar. Comprobar que se puede visualizar información	

	visualización, leer información en el sistema			acerca de la flota de transporte.	
20	Como Jefe Técnico, obtener reportes de las piezas que se han puesto en un transporte a lo largo del tiempo	50	1	Entrar. Insertar una nueva pieza en un transporte. Comprobar su inserción y actualización.	
21	Los reportes que se obtengan deben tener los atributos: nombre del reporte, fecha de impresión, paginado, logo de la empresa.	20	1	Entrar. Obtener un reporte cualquiera y comprobar que muestra todos los atributos necesarios.	
22	Las interfaces del sistema no pueden utilizar logotipos o símbolos corporativos que no estén aprobados.	19	2	No pueden aparecer logotipos o símbolos corporativos que no estén en el Manual de Identidad Visual Corporativo.	

Anexo No 3. Planificación de Sprint.

Tabla 8. Definición del Sprint # 1

ID	Nombre	Imp.	Est.
1	Como Jefe Técnico, mantener actualizada una base de datos con el inventario del transporte.	900	5
2	Como Administrador del Sistema, registrar los codificadores que se emplean.	850	3
3	El sistema debe funcionar después de los cambios que se están haciendo en la Empresa.	800	2
4	Como Administrador del Sistema, controlar el acceso del resto de los usuarios a las funcionalidades del sistema.	780	1
5	Como Jefe Técnico, mantener un control de las piezas del transporte.	750	3
6	Acceder al sistema desde cualquier punto de la red a través de la Web.	700	1
7	Para trabajar en el sistema se debe iniciar sesión usando un usuario y una contraseña.	680	2
8	Como Usuario del Taller, controlar las interrupciones del transporte.	650	2
9	Como Usuario del Taller, controlar los mantenimientos del transporte.	645	2
10	Disponer de una estructura de filtrado por unidades organizativas, que permita la navegación por niveles.	643	1

Tabla 9. Definición del Sprint # 2

ID	Nombre	Imp.	Est.
11	Como Jefe Técnico, obtener reportes de la disponibilidad del transporte que incluyan, si no está disponible, la causa y la demora en la solución.	640	4
12	Como Jefe Técnico, obtener reportes del nivel de explotación del transporte.	630	4
13	El sistema debe ser auditable.	550	2
14	Como Jefe Técnico, registrar los accidentes del transporte.	400	3
15	Como Jefe Técnico, obtener el listado de transportes por modelos, tipos, marcas, tipos de combustible.	390	1
16	Como Jefe Técnico, obtener reportes de los accidentes ocurridos sobre el transporte en un período de tiempo	380	4
17	Como Jefe Técnico, mantener actualizada la información de los parqueos autorizados	200	3
18	Varios usuarios deben poder trabajar sobre el sistema simultáneamente	150	2
19	Como usuario con privilegios de visualización, leer información en el sistema	100	4
20	Como Jefe Técnico, obtener reportes de las piezas que se han puesto en un transporte a lo largo del tiempo	50	1
21	Los reportes que se obtengan deben tener los atributos: nombre del reporte, fecha de impresión, paginado, logo de la empresa.	20	1

22	Las interfaces del sistema no pueden utilizar logotipos o símbolos corporativos que no estén aprobados.	19	2
----	---	----	---

Anexo No 4. Diagramas de Clases del Diseño

Diagrama de clases del Diseño del caso de uso Actualizar Transporte

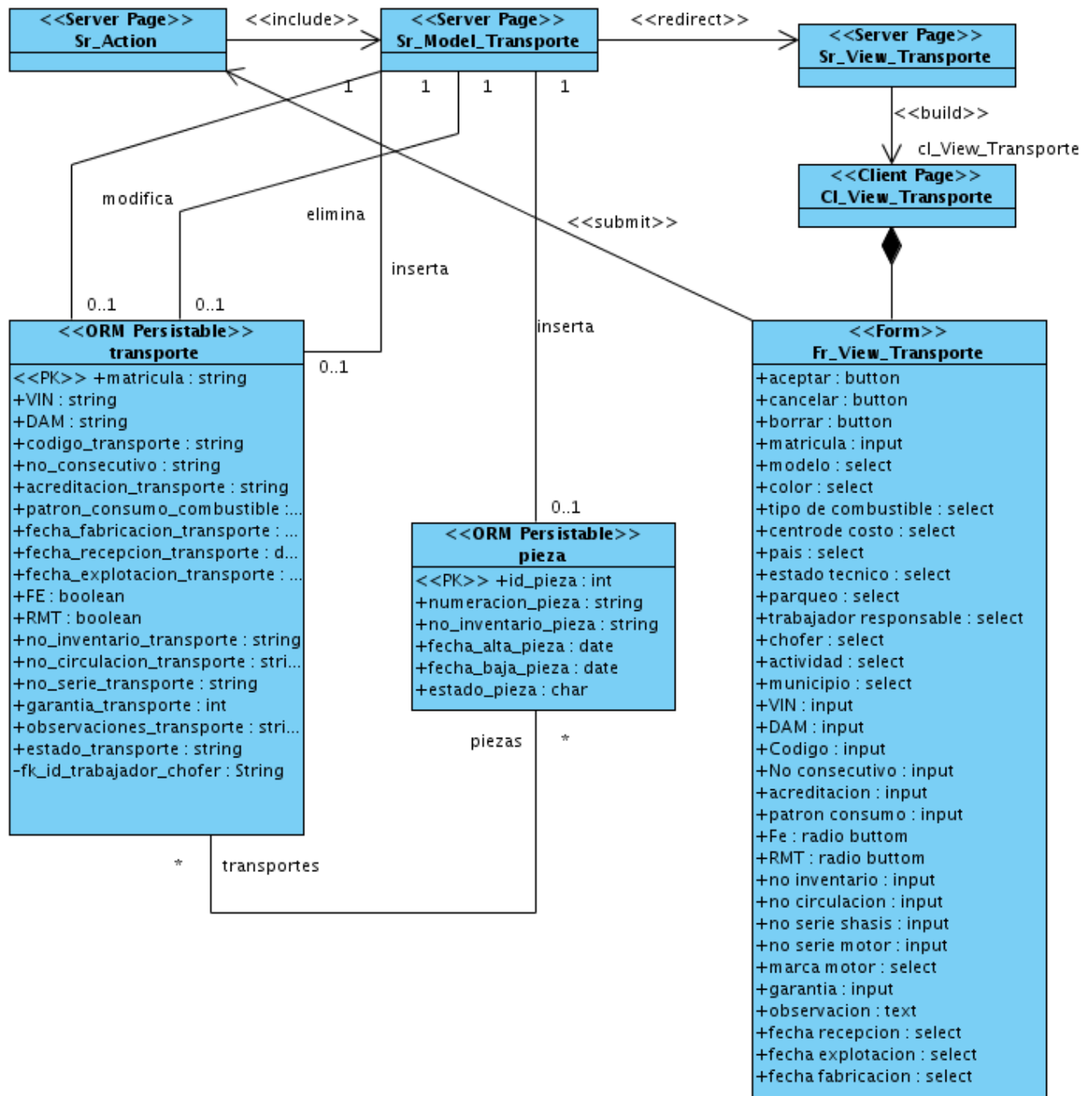


Diagrama de clases del Diseño del caso de uso Actualizar Accidente

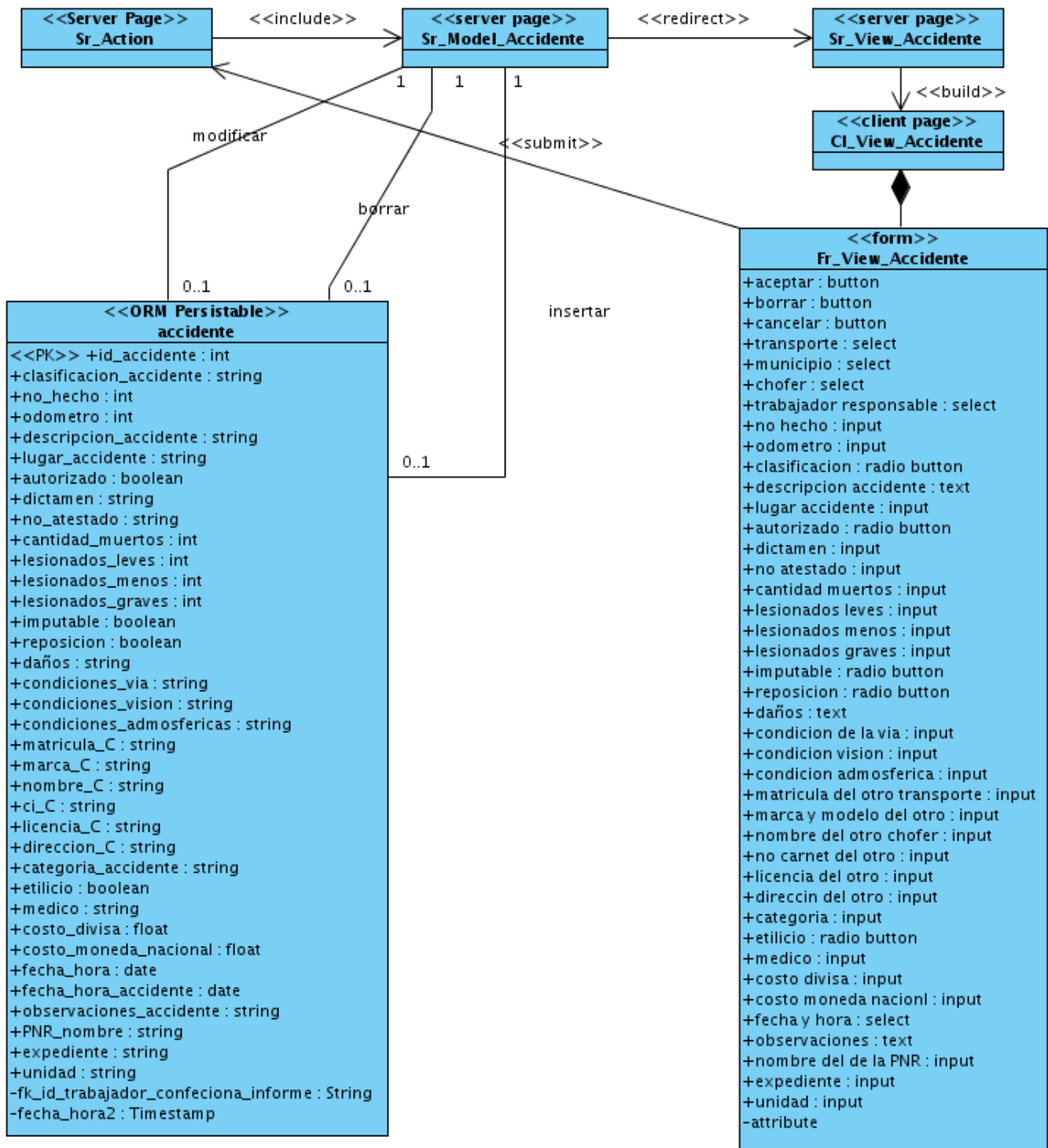


Diagrama de clases del Diseño del caso de uso Visualizar Accidente

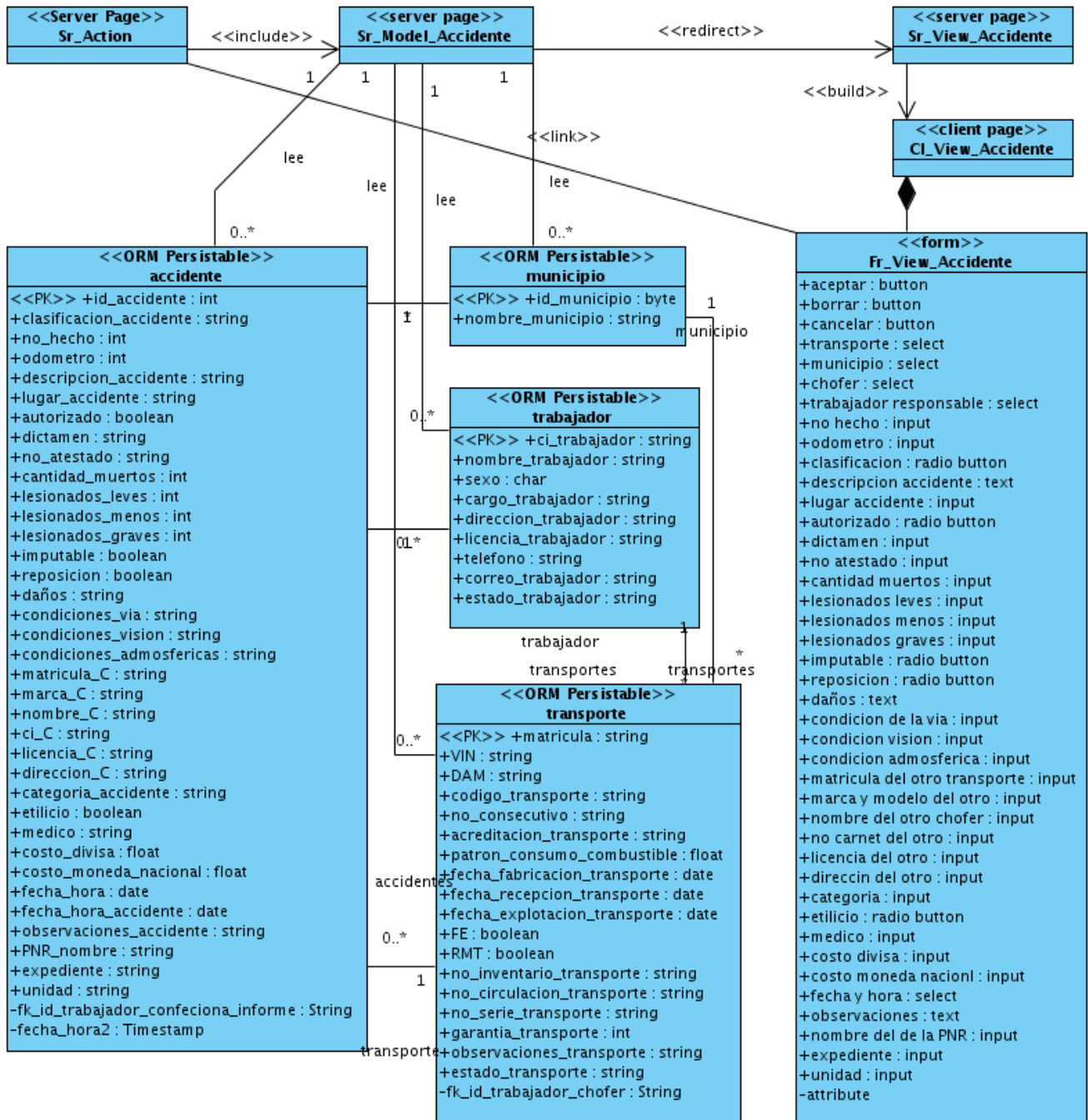


Diagrama de clases del Diseño del caso de uso Actualizar Parqueo

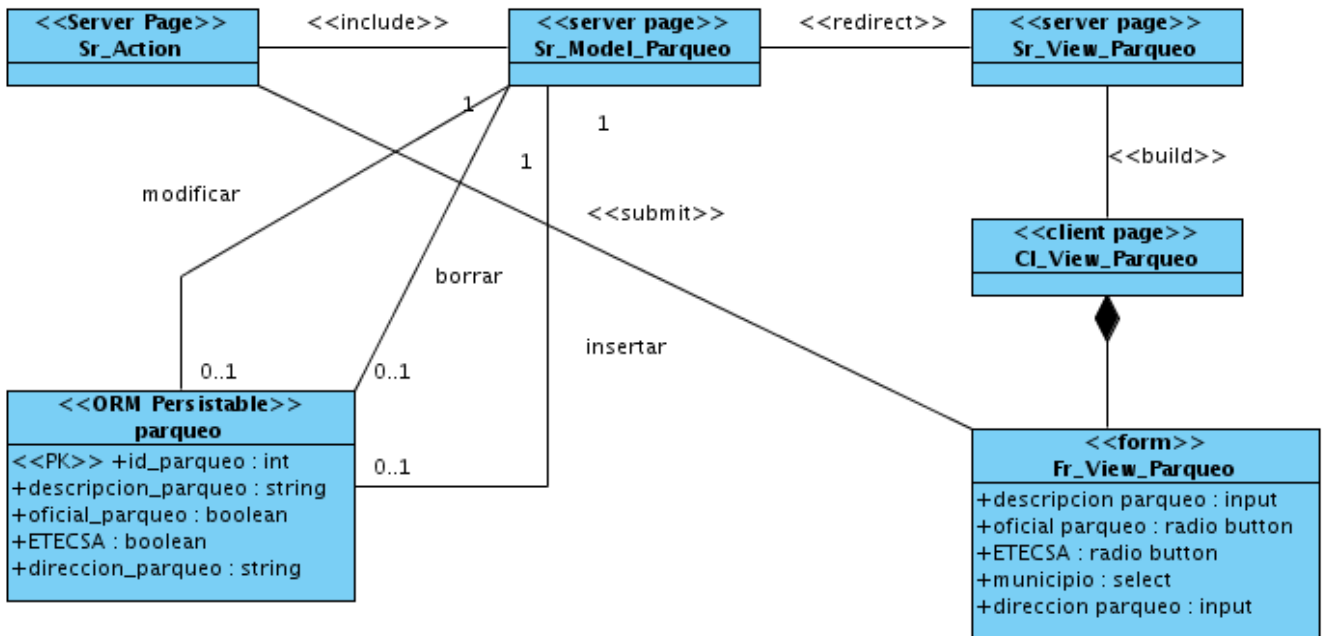
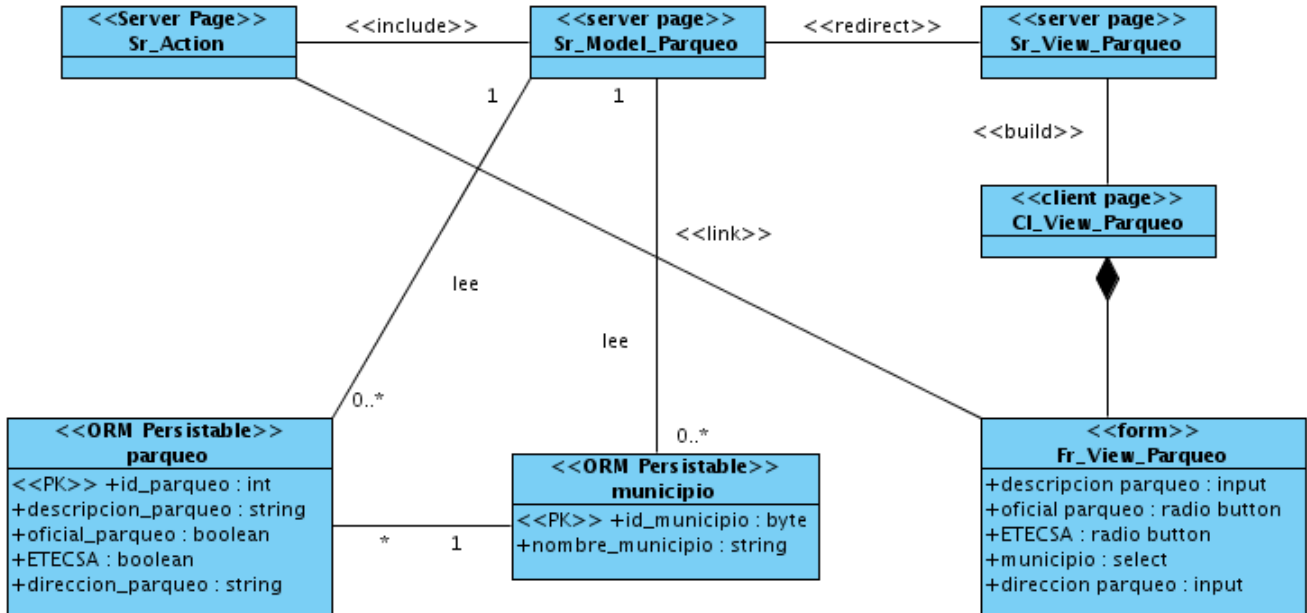


Diagrama de clases del Diseño del caso de uso Visualizar Parqueo



Anexo No 5. COCOMO 2

Entradas Externas (EI)	Cant Ficheros	Cant Elementos	Clasificació n
Actualizar transporte	2	30	alto
Actualizar modelo de transporte	1	6	bajo
Actualizar marca de transporte	1	2	bajo
Actualizar tipo de transporte	1	4	bajo
Actualizar clase de tipo de transporte	1	3	bajo
Actualizar Accidente	1	39	media
Actualizar Pieza	1	8	bajo
Actualizar tipo de Pieza	1	2	bajo
Actualizar marca de Pieza	1	2	bajo
Actualizar Pieza-Transporte	1	6	bajo
Actualizar Eventualidad	1	7	bajo
Actualizar Parqueo	1	6	bajo
Actualizar Informe de Explotación	1	15	bajo
Actualizar Ficav	1	6	bajo
Actualizar Lot	1	5	bajo
Actualizar Centro de Costo	1	4	bajo
Actualizar Actividad	1	3	bajo
Actualizar Tipo de Combustible	1	3	bajo
Actualizar Unidad Organizativa	1	5	bajo
Actualizar Trabajadores	1	10	bajo
Actualizar Usuarios	1	7	bajo
Actualizar Permiso	1	3	bajo
Actualizar Grupo	1	3	bajo
Actualizar Traza	1	7	bajo

Ficheros Lógicos Internos (ILF)	Cant Ficheros	Cant Elementos	Clasificación
accidente	1	39	bajo
actividad	1	3	bajo
centro_de_costo	1	4	bajo
clase_de_tipo_transporte	1	3	bajo
color	1	2	bajo
estado_técnico	1	3	bajo
eventualidad	1	7	bajo
ficav	1	6	bajo
grupo	1	3	bajo
informe_de_explotación	1	15	bajo
lot	1	5	bajo
marca_de_pieza	1	2	bajo
marca_de_transporte	1	2	bajo
modelo_de_transporte	1	7	bajo
municipio	1	3	bajo
país	1	2	bajo
parqueo	1	6	bajo
pieza	1	9	bajo
pieza_transporte	1	6	bajo
provincia	1	3	bajo
permiso	1	3	bajo
grupo_permiso	1	2	bajo
tipo_de_combustible	1	3	bajo
tipo_de_pieza	1	2	bajo
tipo_de_transporte	1	4	bajo
trabajador	1	10	bajo
transporte	1	30	bajo

traza_base_datos	1	7	bajo
unidad_organizativa	1	5	bajo
usuario	1	9	bajo
usuario_grupo	1	2	bajo
usuario_permiso	1	2	bajo

	Cant Ficheros	Cant Elementos	Clasificació n
Salidas externas (EO)			
Visualizar transporte	13	29	alto

	Cant Ficheros	Cant Elementos	Clasificació n
Consultas externas (EQ)			
Visualizar transporte	12	29	alto
Visualizar modelo de transporte	3	6	bajo
Visualizar marca de transporte	1	2	bajo
Visualizar tipo de transporte	2	4	bajo
Visualizar clase de tipo de transporte	1	3	bajo
Visualizar Accidente	4	39	alto
Visualizar Pieza	3	8	bajo
Visualizar tipo de Pieza	1	2	bajo
Visualizar marca de Pieza	1	2	bajo
Visualizar Pieza-Transporte	3	6	bajo
Visualizar Eventualidad	2	7	bajo
Visualizar Parqueo	2	6	bajo
Visualizar Informe de Explotación	2	15	bajo
Visualizar Ficav	2	6	bajo
Visualizar Lot	2	5	bajo
Visualizar Centro de Costo	2	4	bajo

Visualizar Actividad	1	3	bajo
Visualizar Tipo de Combustible	1	3	bajo
Visualizar Unidad Organizativa	3	5	bajo
Visualizar Trabajadores	2	10	bajo
Visualizar Usuarios	3	7	bajo
Visualizar Permiso	1	3	bajo
Visualizar Grupo	1	3	bajo
Visualizar Traza	2	7	bajo

Anexo No 6. Encuesta de opinión de los usuarios del sistema SAGTRANS, fase de Prueba.

Estimado compañero(a), el departamento de TI de la DTHO, con motivo de recopilar opiniones y sugerencias acerca de la reciente implementación del sistema SAGTRANS para la gestión de la información relacionada con la actividad de transporte en la empresa, aplica esta encuesta. Le pedimos por favor que lea cuidadosamente las preguntas siguientes, y nos exprese su criterio. Su opinión vale mucho para nosotros. Gracias anticipadas.

Es usted: Especialista de transporte: ____

- ✓ A continuación sometemos a su valoración una serie de elementos que se consideran importantes en la evaluación del sistema SAGTRANS implementado.

Para esta evaluación se le propone las siguientes categorías:

1. Muy Adecuado (**MA**)
2. Bastante Adecuado (**BA**)
3. Adecuado (**A**)
4. Poco Adecuado (**PA**)
5. No Adecuado (**NA**)

Marque con una **X** la categoría que considera adecuada para cada criterio:

	MA	BA	A	PA	NO
Criterios					
¿Cómo valora la utilidad del sistema informático?					
¿Cómo cree que fueron cumplidas sus expectativas con la implementación?					
¿Qué tanto mejora la integridad de la información?					
¿Cómo fue el uso de las imágenes y los colores?					
¿Qué tan “amigable” le resultó el diseño de las interfaces del sistema?					
¿Cómo valora la organización de la información?					
¿Cómo considera que se mejoró la eficiencia de los					

procesos que se realizaban de forma manual?					
¿Qué tanto apoya el sistema la gestión de la información?					
¿Qué tan útil es la ayuda suministrada?					

✓ ¿Qué otros elementos novedosos encuentra en la aplicación?

✓ Nos gustaría conocer alguna otra idea o sugerencia que no se haya abordado en esta encuesta.

Gracias por su cooperación.

Anexo No 7. Procesamiento de la encuesta de opinión de los usuarios aplicando el método Delphy.

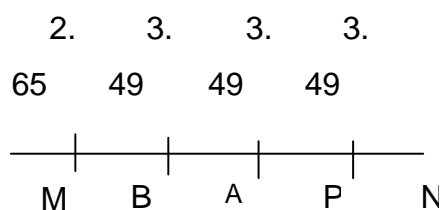
TABLA DE FRECUENCIA ABSOLUTA						
Criterios	M A	BA	A	F A	N A	TOTA L
¿Cómo valora la utilidad del sistema informático?	9					9
¿Cómo cree que fueron cumplidas sus expectativas con la implementación?	6	3				9
¿Qué tanto mejora la integridad de la información?	8	1				9
¿Cómo fue el uso de las imágenes y los colores?	9					9
¿Qué tan “amigable” le resultó el diseño de las interfaces del sistema?	8	1				9
¿Cómo valora la organización de la información?	9					9
¿Cómo considera que se mejoró la eficiencia de los procesos que se realizaban de forma manual?	9					9
¿Qué tanto apoya el sistema la gestión de la información?	9					9
¿Qué tan útil es la ayuda suministrada?	9					9

TABLA DE FRECUENCIA ABSOLUTA ACUMULADA					
Criterios	MA	BA	A	PA	NA
¿Cómo valora la utilidad del sistema informático?	9	9	9	9	9
¿Cómo cree que fueron cumplidas sus expectativas con la implementación?	6	9	9	9	9
¿Qué tanto mejora la integridad de la información?	8	9	9	9	9
¿Cómo fue el uso de las imágenes y los colores?	9	9	9	9	9
¿Qué tan “amigable” le resultó el diseño de las interfaces del sistema?	8	9	9	9	9

¿Cómo valora la organización de la información?	9	9	9	9	9
¿Cómo considera que se mejoró la eficiencia de los procesos que se realizaban de forma manual?	9	9	9	9	9
¿Qué tanto apoya el sistema la gestión de la información?	9	9	9	9	9
¿Qué tan útil es la ayuda suministrada?	9	9	9	9	9

TABLA DE DETERMINACIÓN DE LOS PUNTOS DE CORTES							
	MA	BA	A	P	Su	Prome	N -
					ma	dio	Prom.
¿Cómo valora la utilidad del sistema informático?	3.49	3.49	3.49	3.49	13.96	3.49	3.49
¿Cómo cree que fueron cumplidas sus expectativas con la implementación?	0.43	3.49	3.49	3.49	10.9	2.73	2.73
¿Qué tanto mejora la integridad de la información?	1.22	3.49	3.49	3.49	11.69	2.92	2.92
¿Cómo fue el uso de las imágenes y los colores?	3.49	3.49	3.49	3.49	13.96	3.49	3.49
¿Qué tan “amigable” le resultó el diseño de las interfaces del sistema?	1.22	3.49	3.49	3.49	11.69	2.92	2.92
¿Cómo valora la organización de la información?	3.49	3.49	3.49	3.49	13.96	3.49	3.49
¿Cómo considera que se mejoró la eficiencia de los procesos que se realizaban de forma manual?	3.49	3.49	3.49	3.49	13.96	3.49	3.49
¿Qué tanto apoya el sistema la gestión de la información?	3.49	3.49	3.49	3.49	13.96	3.49	3.49
¿Qué tan útil es la ayuda suministrada?	3.49	3.49	3.49	3.49	13.96	3.49	3.49

Suma	23.8 1	31. 41	3 1.41	3 1.41	11 8.04		
Punto de corte	2.65	3.4 9	3. 49	3. 49	13. 12	3.28	= N (Prom . Gen.)



CONCLUSIONES GENERALES					
Criterios	M A	BA	A	PA	NO
¿Cómo valora la utilidad del sistema informático?	Si	-	-	-	-
¿Cómo cree que fueron cumplidas sus expectativas con la implementación?	Si	-	-	-	-
¿Qué tanto mejora la integridad de la información?	Si	-	-	-	-
¿Cómo fue el uso de las imágenes y los colores?	Si	-	-	-	-
¿Qué tan "amigable" le resultó el diseño de las interfaces del sistema?	Si	-	-	-	-
¿Cómo valora la organización de la información?	Si	-	-	-	-
¿Cómo considera que se mejoró la eficiencia de los procesos que se realizaban de forma manual?	Si	-	-	-	-
¿Qué tanto apoya el sistema la gestión de la información?	Si	-	-	-	-

¿Qué tan útil es la ayuda suministrada?	Si	-	-	-	-
---	----	---	---	---	---