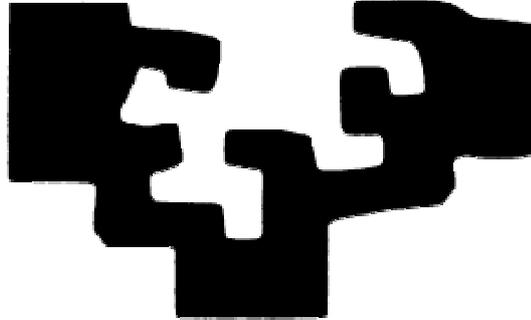


eman ta zabal zazu



universidad
del país vasco

euskal herriko
unibertsitatea

Facultad de Informática

Informatika Fakultatea

TITULACIÓN: Ingeniería Técnica Informática Sistemas

GESTOR DE TRANSPORTES

Autor: Miguel Ángel Rábano Astorga

Directora: Montserrat Hermo Huguet

GESTOR DE TRANSPORTES	1
1. INTRODUCCIÓN	6
2. ANÁLISIS DE ANTECEDENTES	7
3. DOCUMENTO DE OBJETIVOS DEL PROYECTO Y PLANIFICACIÓN	8
3.1 OBJETIVOS DEL PROYECTO	8
3.2 PROCESO DE DESARROLLO	9
3.2.1 Ciclo de vida.....	9
3.2.2 Fases del proyecto y su división en actividades	10
3.2.2.1 <i>Formación</i>	10
3.2.2.2 <i>Gestión del Proyecto</i>	10
3.2.2.3 <i>Definición de Objetivos del Proyecto</i>	11
3.2.2.4 <i>Requisitos</i>	11
3.2.2.5 <i>Diseño</i>	11
3.2.2.6 <i>Implementación</i>	12
3.2.2.7 <i>Pruebas</i>	12
3.2.2.8 <i>Implantación</i>	12
3.2.3 Estimaciones de tiempo	14
3.2.3.1 <i>Diagramas de Gantt</i>	17
3.2.3.1.1 Fases	17
3.2.3.1.2 Formación	18
3.2.3.1.3 Definición de Objetivos del Proyecto	18
3.2.3.1.4 Requisitos	18
3.2.3.1.5 Diseño	19
3.2.3.1.6 Implementación	19
3.2.3.1.7 Pruebas e Integración	19
3.2.3.1.8 Implantación	20
3.2.3.1.9 Gestión del Proyecto	20
3.2.4 Criterios de terminación.....	21
3.2.5 Valoraciones de riesgos (Plan de contingencia)	22
3.2.6 Supuestos claves del proyecto	23
3.3 DESCRIPCIÓN DEL MARCO DE TRABAJO.....	23
3.3.1 Flujo de Información.....	23
3.3.2 Recursos.....	24
3.3.2.1 <i>Humanos</i>	24
3.3.2.2 <i>Técnicos</i>	24
3.4 FACTIBILIDAD	25

3.5	DESVIACIÓN EN LA PLANIFICACIÓN	25
4.	CAPTURA DE REQUISITOS.....	26
4.1	USUARIOS DEL SISTEMA.....	26
4.2	MODELO DE CASOS DE USO DE LA APLICACIÓN	26
4.2.1	Módulo Actividad.....	28
4.2.1.1	Caso de uso <i>Transportistas</i>	28
4.2.1.1.1	Diagrama	28
4.2.1.1.2	Acciones	29
4.2.1.1.2.1	<i>Editar</i>	29
4.2.1.1.2.2	<i>Eliminar</i>	29
4.2.1.1.2.3	<i>Filtrar</i>	29
4.2.1.1.2.4	<i>Buscar</i>	30
4.2.1.1.2.5	<i>Consultas</i>	30
4.2.1.2	Caso de uso <i>Trayectos</i>	31
4.2.1.2.1	Diagrama	32
4.2.1.2.2	Acciones	32
4.2.1.2.2.1	<i>Informes Genéricos</i>	32
4.2.1.2.2.2	<i>Editar</i>	33
4.2.1.2.2.3	<i>Seleccionar Cliente</i>	33
4.2.1.2.2.4	<i>Informe morosidad cliente</i>	33
4.2.1.2.2.5	<i>Histórico de trayecto - Editar</i>	34
4.2.1.2.2.6	<i>Histórico de trayecto - Eliminar</i>	34
4.2.1.2.2.7	<i>Orden de carga – Seleccionar Lugar</i>	34
4.2.1.2.2.8	<i>Orden de carga – Generar Documento</i>	35
4.2.1.2.2.9	<i>Orden de carga - Editar</i>	35
4.2.1.2.2.10	<i>Orden de carga - Eliminar</i>	35
4.2.1.2.2.11	<i>Eliminar</i>	35
4.2.1.2.2.12	<i>Filtrar</i>	36
4.2.1.2.2.13	<i>Buscar</i>	36
4.2.1.2.2.14	<i>Consultas</i>	36
4.2.2	Módulo Financiero.....	37
4.2.2.1	Caso de uso <i>Facturas Emitidas</i>	37
4.2.2.1.1	Diagrama	38
4.2.2.1.2	Acciones	38
4.2.2.1.2.1	<i>Imprimir</i>	38
4.2.2.1.2.2	<i>Enviar Mail/Fax</i>	39
4.2.2.1.2.3	<i>Informes Genéricos - Facturas</i>	39
4.2.2.1.2.4	<i>Informes Genéricos – Facturas agrupadas por cliente</i>	39
4.2.2.1.2.5	<i>Editar</i>	40
4.2.2.1.2.6	<i>Editar - Seleccionar Cliente</i>	40
4.2.2.1.2.7	<i>Editar – Certificado Cliente</i>	40
4.2.2.1.2.8	<i>Editar – Línea Detalle - Editar</i>	40
4.2.2.1.2.9	<i>Editar – Línea Detalle – Editar – Trayectos - Navegar a Trayecto</i>	41
4.2.2.1.2.10	<i>Editar – Línea Detalle – Editar – Trayectos - Editar</i>	41
4.2.2.1.2.11	<i>Editar – Línea Detalle – Editar- Trayectos- Importar Trayectos</i>	41
4.2.2.1.2.12	<i>Editar – Línea Detalle – Eliminar</i>	41
4.2.2.1.2.13	<i>Eliminar</i>	42
4.2.2.1.2.14	<i>Filtrar</i>	42
4.2.2.1.2.15	<i>Buscar</i>	42
4.2.2.1.2.16	<i>Consultas</i>	42
4.2.2.2	Caso de uso <i>Facturas Recibidas</i>	43
4.2.2.2.1	Diagrama	44
4.2.2.2.2	Acciones	44
4.2.2.2.2.1	<i>Editar</i>	44
4.2.2.2.2.2	<i>Editar - Seleccionar Proveedor</i>	45
4.2.2.2.2.3	<i>Editar – Certificado Proveedor</i>	45
4.2.2.2.2.4	<i>Editar – Importar Trayectos del Proveedor</i>	45
4.2.2.2.2.5	<i>Editar – Navegar a Trayecto en modo Consulta</i>	46
4.2.2.2.2.6	<i>Editar – Desvincular Trayecto</i>	46
4.2.2.2.2.7	<i>Eliminar</i>	46
4.2.2.2.2.8	<i>Filtrar</i>	46
4.2.2.2.2.9	<i>Buscar</i>	46

4.2.2.2.10	Consultas	46
4.2.2.3	Caso de uso Pagarés a Proveedores	47
4.2.2.3.1	Diagrama	48
4.2.2.3.2	Acciones	48
4.2.2.3.2.1	Imprimir	48
4.2.2.3.2.2	Generar Documento Word	48
4.2.2.3.2.3	Editar	49
4.2.2.3.2.4	Editar - Seleccionar Proveedor	49
4.2.2.3.2.5	Editar – Editar Certificado Proveedor	49
4.2.2.3.2.6	Editar – Importar Facturas Pendientes de Proveedor	49
4.2.2.3.2.7	Editar – Navegar a Factura Recibida de Proveedor	49
4.2.2.3.2.8	Editar – Desvincular Factura	50
4.2.2.3.2.9	Eliminar	50
4.2.2.3.2.10	Filtrar	50
4.2.2.3.2.11	Buscar	50
4.2.2.3.2.12	Consultas	50
5.	MODELO DE DATOS	51
6.	DECISIONES TÉCNICAS	52
7.	DISEÑO	54
7.1	DISEÑO DE LA BASE DE DATOS	55
7.2	DISEÑO DE LAS CLASES MÁS SIGNIFICATIVAS	56
7.2.1	Módulo Actividad	56
7.2.1.1	Clase <i>Transportista</i>	56
7.2.1.1.1	Propiedades	56
7.2.1.1.2	Métodos	56
7.2.1.2	Clase <i>Trayecto</i>	58
7.2.1.2.1	Propiedades	58
7.2.1.2.2	Métodos	59
7.2.2	Módulo Financiero	61
7.2.2.1	Clase <i>Factura Emitida</i>	61
7.2.2.1.1	Propiedades	61
7.2.2.1.2	Métodos	61
7.2.2.2	Clase <i>Factura Recibida</i>	64
7.2.2.2.1	Propiedades	64
7.2.2.2.2	Métodos	64
7.2.2.3	Clase <i>Pagaré a Proveedor</i>	68
7.2.2.3.1	Propiedades	68
7.2.2.3.2	Métodos	68
8.	PRUEBAS DEL SISTEMA	72
8.1	PLAN DE PRUEBAS	72
8.1.1	Casos de Prueba del Caso de Uso Transportistas	73
8.1.2	Casos de Prueba del Caso de Uso Trayectos	75
8.1.3	Casos de Prueba del Caso de Uso Facturas Emitidas	79
8.1.4	Casos de Prueba del Caso de Uso Facturas Recibidas	83
8.1.5	Casos de Prueba del Caso de Uso Pagarés a Proveedores	86

8.2	CONCLUSIONES.....	90
9.	CONCLUSIONES GENERALES	91
10.	LÍNEAS FUTURAS.....	92
11.	BIBLIOGRAFÍA	93

1. INTRODUCCIÓN

El proyecto final de carrera (PFC) es un aspecto clave en la inserción al mercado laboral de un estudiante. Mi intención era realizar un proyecto en el que aplicar los conocimientos adquiridos durante mis años de carrera, así como dar una solución real a las necesidades de la empresa de transportes FERCAEZ.

El proyecto denominado “Gestión de Transportes”, en adelante nombrado como “GeTrans”, engloba una aplicación Cliente-Servidor, que permite a varios usuarios gestionar la operativa así como diversos aspectos económicos de la empresa FERCAEZ.

GeTrans es una aplicación Cliente-Servidor desarrollada para Windows en el que los datos radican en una base de datos centralizada.

De modo general, GeTrans gestiona información agrupada en secciones. La sección General permite gestionar la información relativa a clientes, proveedores y tablas maestras como por ejemplo bancos, formas de pago, tractores, remolques y estados de trayectos. La sección Actividad gestiona información relacionada con transportistas y trayectos. La sección financiera gestiona información relacionada con facturas emitidas y recibidas así como pagarés de proveedores. La sección de Utilidades contempla aspectos relacionados con la aplicación como la configuración así como la gestión de backups de los datos. La sección Ventana contiene aspectos relacionados con la forma y posición en que los datos se presentan al usuario.

En resumen, GeTrans ofrece a los usuarios una gestión completa de la actividad de transporte para pequeñas empresas, en la que si bien se da cabida a las soluciones particulares de la empresa FERCAEZ, es prioritaria la estandarización de las diferentes funcionalidades para tratar dicha actividad.

2. ANÁLISIS DE ANTECEDENTES

Antes de la implantación de la aplicación GeTrans, la empresa Fercaez almacenaba algunos datos en libros Excel o en papel, editaba documentos que enviaban a clientes y proveedores de forma manual y enviaba, mediante correo electrónico o fax, estos documentos manualmente.

Para los clientes y proveedores, se obtenían sus datos básicos mediante contacto telefónico y se almacenaban en un libro Excel, pudiendo producirse la repetición de datos almacenados y una información incongruente (un cliente repetido dos veces y con diferente información).

Se realizaba la reclamación de un documento que prueba la entrega de mercancía mediante una plantilla denominada ReclamacionProveedores.doc. No se realizaba ningún tipo de almacenamiento de dicha información.

Para enviar la orden de carga a los proveedores utilizaban la plantilla OrdenCarga.doc. En este caso tampoco se realizaba ningún tipo de almacenamiento de dicha información.

Cuando se enviaba el pago de sus facturas mediante el documento PagareProveedores.doc. No se realizaba ningún tipo de almacenamiento de dicha información.

A los clientes, se les reclamaba las facturas vencidas y no cobradas, en el idioma en el que se establecía contacto (español, francés o inglés), mediante la plantilla ReclamacionFacturas.doc. Al igual que en los casos anteriores, no se realizaba ningún tipo de almacenamiento de dicha información.

La factura se generaba mediante un libro Excel llamada Factura.xls. Estas facturas se imprimían y se enviaban mediante correo a los clientes. El almacenamiento de las facturas era solamente la de las hojas impresas.

Finalmente no había ningún tipo de gestión sobre los trayectos y transportistas.

3. DOCUMENTO DE OBJETIVOS DEL PROYECTO Y PLANIFICACIÓN

En este capítulo se muestran los objetivos que se desean alcanzar junto con los requisitos del proyecto para ser llevado a cabo. Después se describen brevemente los sujetos que pueden interactuar con la aplicación junto con las opciones de las que disponen.

A continuación, se presentan los temas relacionados con la gestión y ejecución del proyecto que son importantes para conseguir su finalización correcta. Se incluyen las fases del proyecto, los riesgos que se dan durante su realización y su plan de contingencia y, finalmente, se aportará la planificación del proyecto y las desviaciones que se han producido junto con el presupuesto de la aplicación.

3.1 Objetivos del proyecto

En el presente proyecto, el objetivo general es la creación y desarrollo de una aplicación Cliente-Servidor que permita gestionar tanto la actividad propia del negocio (como son los trayectos que realizan los camiones así como las cargas subcontratadas) como la gestión económico-contable de la actividad (facturas, pagarés).

El proyecto nace con el objetivo de crear un sistema de información que realice toda la gestión de información de forma centralizada, organizada e interrelacionada entre sí. Esto permitirá que los datos no se pierdan y acceder rápidamente a ellos. Aumentar la eficacia y eficiencia en el trabajo, así como minimizar la pérdida de control y pérdida de tiempo.

Debido a los antecedentes de la operativa habitual de la empresa, anteriormente detallados, es necesaria una tarea pedagógica orientada a los usuarios tanto en el aspecto de adquirir un hábito en la nueva forma de operar como en el aspecto de comprender de forma global el nuevo sistema.

3.2 Proceso de Desarrollo

3.2.1 Ciclo de vida

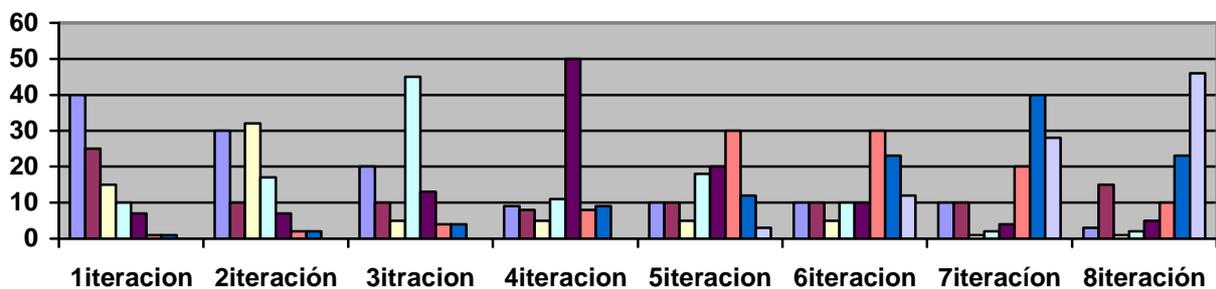
Entre los modelos del proceso de desarrollo de ingeniería del software existentes se decide utilizar el modelo denominado ITERATIVO (prototipado). Este modelo se basa en la ampliación y refinamiento sucesivos del sistema, mediante múltiples iteraciones con retroalimentación cíclica y adaptación como elementos principales, que dirigen la convergencia hacia un sistema adecuado.

El sistema crece incrementalmente a lo largo del tiempo, iteración tras iteración, y por ello también este ciclo de vida se puede denominar iterativo e incremental. Cada iteración conlleva la elección de un pequeño conjunto de requisitos y rápidamente, diseñar, implementar y probar.

En las primeras iteraciones, la elección de los requisitos y el diseño podrían no ser exactamente lo que se desea al final. Pero el acto de dar un pequeño paso con rapidez, antes de capturar todos los requisitos y que el diseño completo se haya definido de forma especulativa, lleva a una rápida retroalimentación lo cual es de gran importancia, incluso más que las especulaciones sobre los requisitos y diseños correctos, ya que la retroalimentación a partir de la construcción y prueba realista de algo, da la posibilidad de modificar o adaptar los requisitos o el diseño. Pero además de clarificar los requisitos, actividades como las pruebas de carga probarán si el diseño y la implementación parcial están en el camino correcto o si por el contrario en el siguiente paso se necesita un cambio en la arquitectura básica. Por tanto cuanto antes se resuelvan y prueben las decisiones de diseño críticas y arriesgadas mejor.

En consecuencia el trabajo se desarrolla a lo largo de una serie de ciclos estructurados de construir-retroalimentar-adaptar. A continuación se muestra un gráfico con los distintos porcentajes de las tareas en cada iteración definidas para el proyecto que se explican más adelante junto con sus actividades.

En las primeras iteraciones tienen más carga de tiempo las fases de formación y gestión del proyecto porque es necesaria una buena organización y conocimiento del campo en el cual se trabaja. Veamos por tanto el gráfico de un proceso iterativo normal:



■ Formación

■ Definición de objetivos del proyecto

■ diseño

■ pruebas

■ Gestion del proyecto

■ requisitos

■ implementación

■ implantación

3.2.2 Fases del proyecto y su división en actividades

Para un buen desarrollo del proyecto es conveniente plantearse los distintos trabajos a desarrollar junto con sus actividades. Las fases que se han estimado necesarias son las siguientes:

3.2.2.1 Formación

La formación es el primer paso que hay que dar junto con la gestión para comenzar un proyecto. La formación comienza en la evaluación de las distintas herramientas que se pueden utilizar para desarrollar nuestro proyecto y eligiendo la más conveniente. Posteriormente se produce la familiarización y aprendizaje del funcionamiento de las mismas. Finalmente se trabajará con herramientas estándares y con un nivel de integración óptimo en Windows XP.

Las tareas que componen esta fase son:

- Entrevista con el cliente (sugiriéndole y asesorándole el tipo de sistema de información que desarrollemos).
- Formación de la actividad de la empresa.
- Documentación UML y Ms Project.

3.2.2.2 Gestión del Proyecto

Esta fase no debe enmarcarse antes ni después de otra, ya que se debe llevar a cabo a lo largo de todo el proyecto para el buen desarrollo de este. Podríamos clasificarla como una de las más importantes del proyecto.

La gestión de un proyecto conlleva las siguientes tareas:

- Planificación de Tareas (El proyecto se divide en tareas y éstas a su vez en actividades ya que es más fácil empezar haciendo cosas pequeñas y a partir de ahí construir otras más grandes).
 - División de tareas y actividades.
 - Estimación de tiempo (Diagramas de Gantt).
- Seguimiento del Proyecto (comprobar que los objetivos se van cumpliendo y si no aplicar el plan correspondiente).
- Reuniones con personal de la empresa Fercaez (Se llevarán a cabo a lo largo de todo el proyecto para confirmar que el trabajo que se está realizando es correcto).
- Plan control de cambios.
- Plan control de riesgos.
- Preparación memoria.
 - Consulta memorias años anteriores.

3.2.2.3 Definición de Objetivos del Proyecto

Esta fase intenta establecer las funcionalidades a incluir y a excluir del sistema.

Las tareas que componen esta fase son:

- Entrevista con el Usuario (Establecer y obtener acuerdo formal con la empresa FERCAEZ de las funcionalidades del sistema).
- Identificar objetivos del proyecto.
- Realizar Borrador DOP (Documento de Objetivos del Proyecto).
- Revisión Borrador DOP.
- Aprobación definitiva del DOP.

3.2.2.4 Requisitos

La captura de requisitos establece los tipos de datos con los que se quiere trabajar en cada módulo y define las especificaciones de manera clara y completa.

Sus tareas son:

- Identificación de requisitos.
- Borrador informe de requisitos.
- Informe de requisitos definitivo.
- Borrador Modelos caso de Uso
- Modelos caso de uso definitivo
- Borrador Modelo de dominio
- Modelo del dominio definitivo

3.2.2.5 Diseño

El diseño puede clasificarse desde dos puntos de vista distintos. El punto de vista del usuario final donde se realiza el prototipo de las interfaces o el punto de vista técnico (ordenador, sistema operativo, lenguaje de programación, base de datos).

Las tareas de esta fase son:

- Diseño de Interfaces (prototipo)
 - Clientes
 - Proveedores
 - Tablas editables
 - Transportistas
 - Trayectos
 - Facturas Emitidas y Recibidas
 - Pagares
 - Plantillas de documentos (Facturas, ReclamacionFacturas...)

- Diseño Base de datos.
 - Entidades
 - Relaciones

3.2.2.6 Implementación

En esta fase se produce el desarrollo del código, es decir, la programación. El objetivo es ir construyendo cada componente del sistema asegurando que finalmente conformen un sistema integrado.

Las tareas que ello conlleva son:

- Instalación de las herramientas necesarias para el desarrollo del software.
- Implementación de la aplicación GeTrans.

3.2.2.7 Pruebas

Las pruebas consisten en someter al sistema a una serie de situaciones y condiciones límites para verificar que el sistema cumple las especificaciones realizadas. Las pruebas se llevan a cabo por módulos y se realizan sucesivamente en bloques mayores de módulos.

Las tareas para esta fase son:

- Pruebas de los módulos (se realizan junto con la implementación de cada módulo).
- Pruebas del sistema de información al completo.

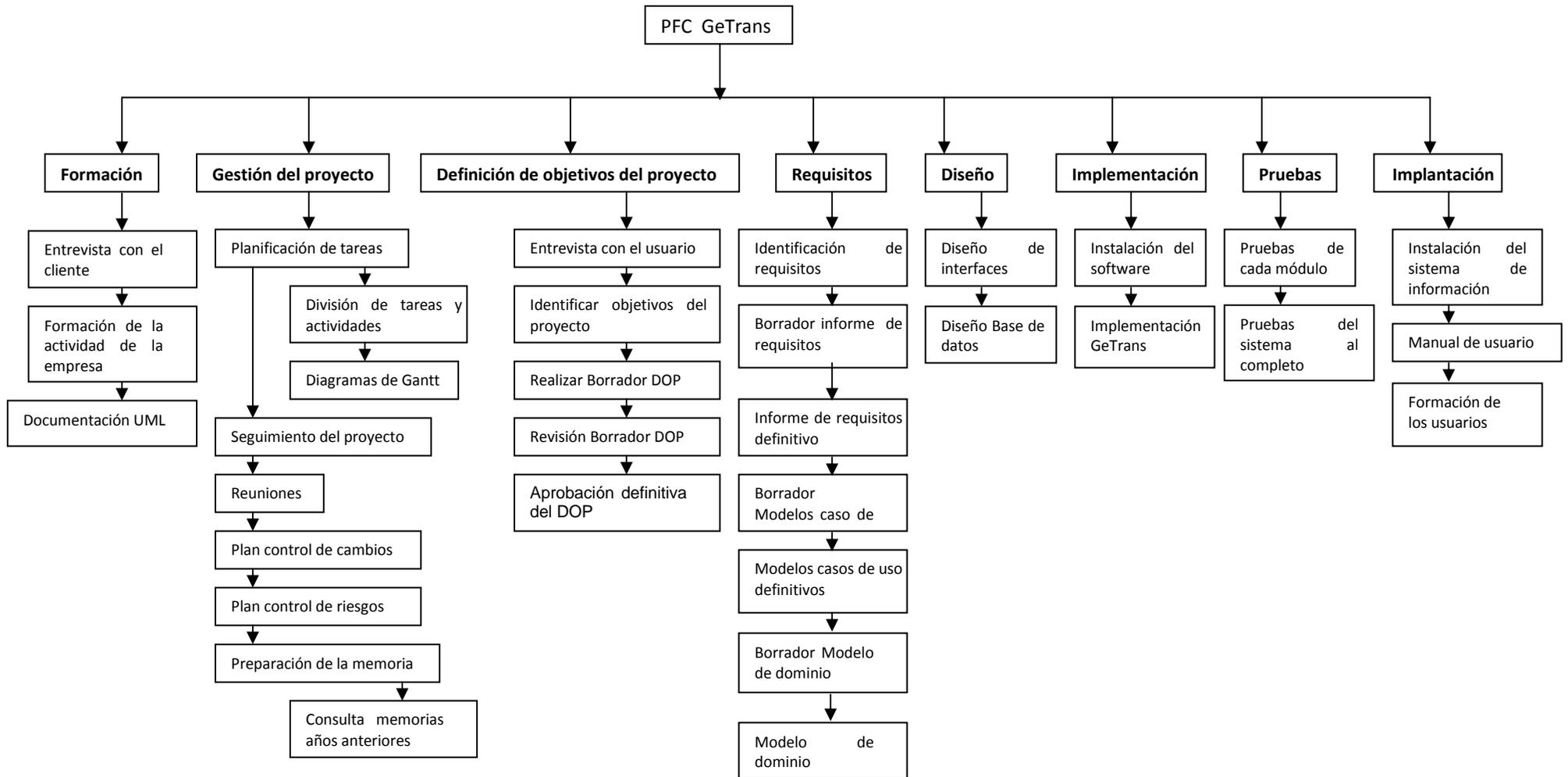
3.2.2.8 Implantación

Se procede a la instalación del sistema de información en la empresa FERCAEZ. Se crean los manuales que explican el funcionamiento de la aplicación y se procede a la formación de los futuros usuarios.

Por tanto las tareas para esta fase son:

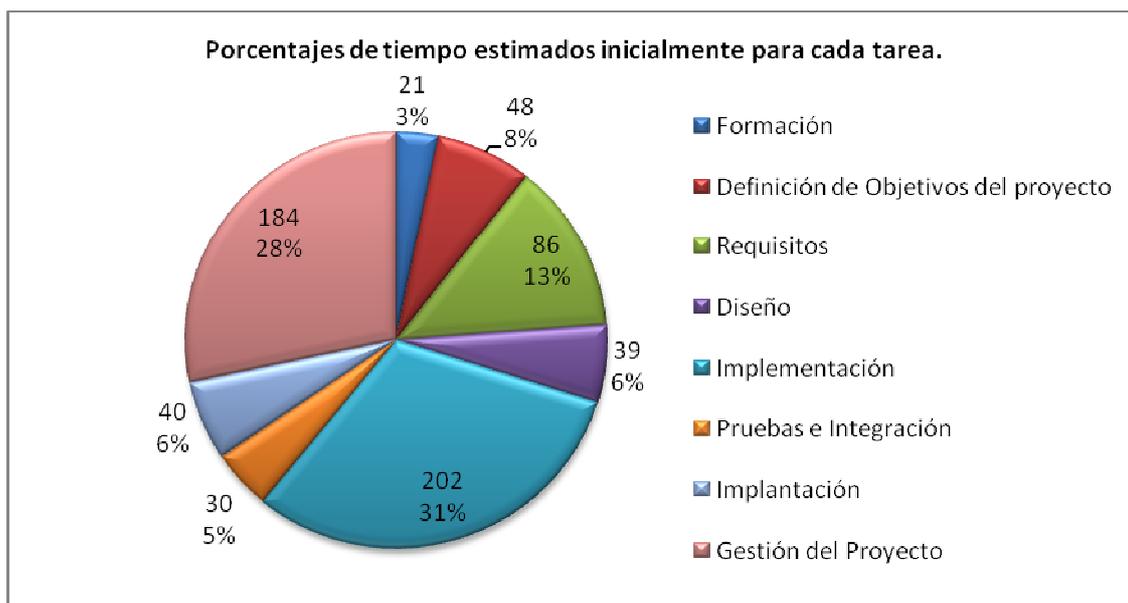
- Instalación del sistema de información.
- Manual de Usuario.
- Formación de los usuarios.

A continuación podemos ver en la EDT, las fases y tareas en las que se ha dividido el proyecto.



3.2.3 Estimaciones de tiempo

Las horas estimadas son aquellas que se calculan al inicio del proyecto. Inicialmente se planificaron 650 horas de trabajo para llevar a cabo el desarrollo del proyecto. A continuación se observa un gráfico con los distintos porcentajes de horas planificadas para cada una de las fases en las que se ha dividido el proyecto.



Como se observa en el gráfico, en la planificación inicial las fases que más tiempo conllevan son las de “Implementación” y “Gestión del proyecto”, ya que una buena gestión, seguimiento y toma de decisiones oportunas en cada momento nos acerca más a un proyecto con éxito.

Otras fases importantes en las que se asignó bastante carga de tiempo son la Captura de requisitos y el diseño, ya que una buena captura de requisitos puede ahorrarnos mucho tiempo tanto a la hora de diseñar como de implementar. Además la captura de requisitos es esencial porque es necesaria una buena especificación de las funcionalidades del sistema de información.

Para las “Pruebas” también se planificó tanta carga de horas como para el diseño y la captura de requisitos, ya que es necesaria una buena batería de pruebas para comprobar el correcto funcionamiento del sistema.

A continuación se muestra en detalle la siguiente información:

- **Fase.** El nombre de la fase o tarea. Una tarea es una actividad que se clasifica en una fase.
- **Duración.** N° de horas estimada para cada fase o actividad.
- **Fecha inicio.** Fecha de inicio planificada para cada tarea.
- **Fecha final.** Fecha de fin planificada para cada tarea.
- **Predecesor.** Tarea o tareas de la que depende la tarea.
- **Días (8h/día).** N° de días de duración estimado. El criterio empleado para cuantificar los días de duración, es estimar el n° de horas de cada actividad, y por cada actividad, dividir el n° de horas entre horas/día de trabajo.

$$N^{\circ} \text{ días} = \frac{N^{\circ} \text{ de Horas}}{\text{Horas Trabajo / día}}$$

Donde el n° de horas de trabajo por día estimadas son 8h.

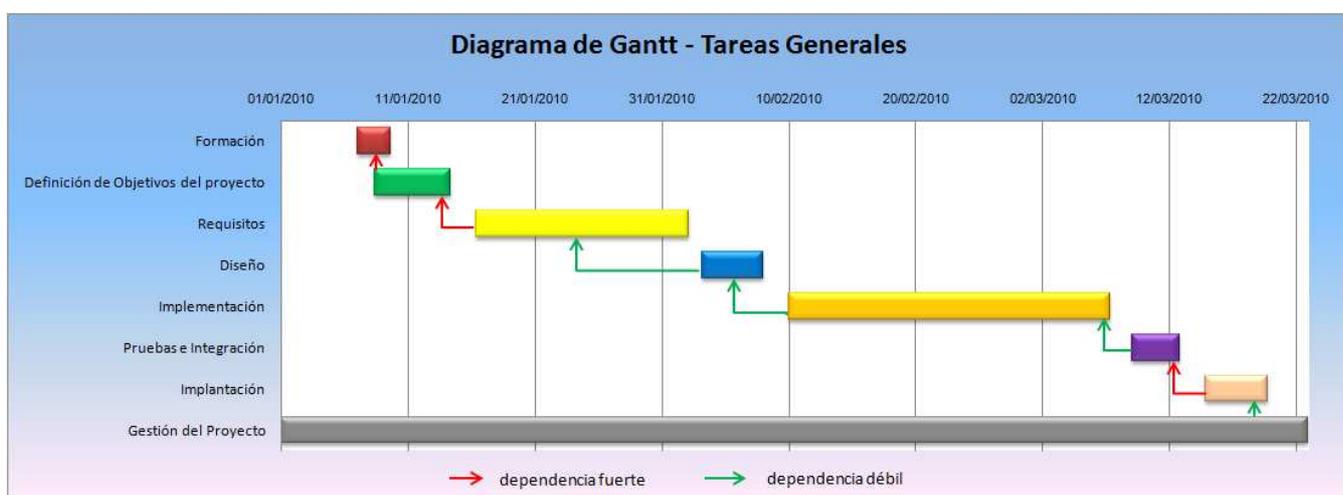
Total Horas PFC GeTrans	650				81,25
Fase	Duración	Comienzo	Fin	Predecesor	Días (8 h/día)
Formación	21	07/01/2010	09/01/2010		2,625
Entrevista con el cliente	3	07/01/2010	07/01/2010		0,375
Formación de la actividad de la empresa	10	07/01/2010	08/01/2010		1,25
Documentación UML	8	07/01/2010	08/01/2010		1
Definición de Objetivos del proyecto	48	08/01/2010	15/01/2010		6
Entrevista con el usuario	4	08/01/2010	08/01/2010	5	0,5
Identificar Objetivos del proyecto	21	09/01/2010	12/01/2010	9	2,625
Realizar Borrador DOP	14	12/01/2010	14/01/2010	10	1,75
Revisión Borrador DOP	7	14/01/2010	15/01/2010	11	0,875
Aprobación definitiva del DOP	2	15/01/2010	15/01/2010	12	0,25
Requisitos	86	16/01/2010	02/02/2010	8	10,75
Identificar requisitos	21	16/01/2010	19/01/2010	8	2,625
Borrador informe de requisitos	7	20/01/2010	20/01/2010	15	0,875
Informe de requisitos definitivo	7	21/01/2010	22/01/2010	16	0,875
Borrador Modelo casos de uso	25	23/01/2010	26/01/2010	17	3,125
Modelo casos de uso definitivo	8	27/01/2010	28/01/2010	18	1
Borrador modelo de dominio	14	29/01/2010	31/01/2010	17	1,75
Modelo de dominio definitivo	4	01/02/2010	02/02/2010	20	0,5
Diseño	39	03/02/2010	09/02/2010	14	4,875
Diseño de Base de datos	16	03/02/2010	05/02/2010	21	2
Diseño de interfaces (prototipos)	23	06/02/2010	09/02/2010	23	2,875
Implementación	202	10/02/2010	08/03/2010	26	25,25
Instalación de herramientas	2	10/02/2010	10/02/2010		0,25
Implementación GeTrans	200	11/02/2010	08/03/2010	26	25
Pruebas e Integración	30	09/03/2010	13/03/2010	25	3,75
Pruebas de cada módulo	15	09/03/2010	10/03/2010		1,875
Pruebas del SI completo	15	11/03/2010	13/03/2010	29	1,875
Implantación	40	14/03/2010	20/03/2010	28	5
Instalación del SI	7	14/03/2010	15/03/2010		0,875
Manual de usuario	21	15/03/2010	18/03/2010		2,625
Formación de los usuarios	12	19/03/2010	20/03/2010	32,33	1,5
Gestión del Proyecto	184	01/01/2010	22/03/2010		23
Planificación de las tareas	15	01/01/2010	02/01/2010		1,875
Seguimiento del proyecto	20	03/01/2010	06/01/2010		2,5
Reuniones cliente	21	03/01/2010	06/01/2010		2,625
Reuniones tutor universidad	8	03/01/2010	04/01/2010		1
Plan control de cambios	12	03/01/2010	05/01/2010		1,5
Plan control de riesgos	12	01/01/2010	02/01/2010		1,5
Preparación de la memoria	80	01/01/2010	11/01/2010		10
Preparación de la defensa	16	20/03/2010	22/03/2010		2

3.2.3.1 Diagramas de Gantt

A continuación se muestra una relación de los diagramas de Gantt tanto para el proyecto como para cada fase, en el que se pueden observar la descomposición de tareas estimadas, la fecha de inicio y la duración

3.2.3.1.1 Fases

El siguiente diagrama de Gantt ilustra **la planificación de las fases** que componen el proyecto. Además, se puede observar el grado de dependencia entre las diferentes fases.



Una dependencia fuerte (flecha roja) indica que una fase no puede comenzar hasta que finalice completamente la fase de la que depende.

Una dependencia débil (flecha verde) indica que una fase puede comenzar y realizarse en paralelo con la fase de la que depende.

Por ejemplo, como se muestra en el gráfico, la fase de implantación no puede realizarse hasta que no haya terminado la fase de pruebas.

3.2.3.1.2 Formación

En el siguiente diagrama de Gantt se puede observar que tanto la tarea “Formación de la actividad de la empresa” como “Documentación UML” son actividades no dependientes entre ellas, y que por tanto, se podrían llevar en paralelo.



3.2.3.1.3 Definición de Objetivos del Proyecto

Todas las actividades son dependientes entre ellas cronológicamente y no se pueden realizar en paralelo.



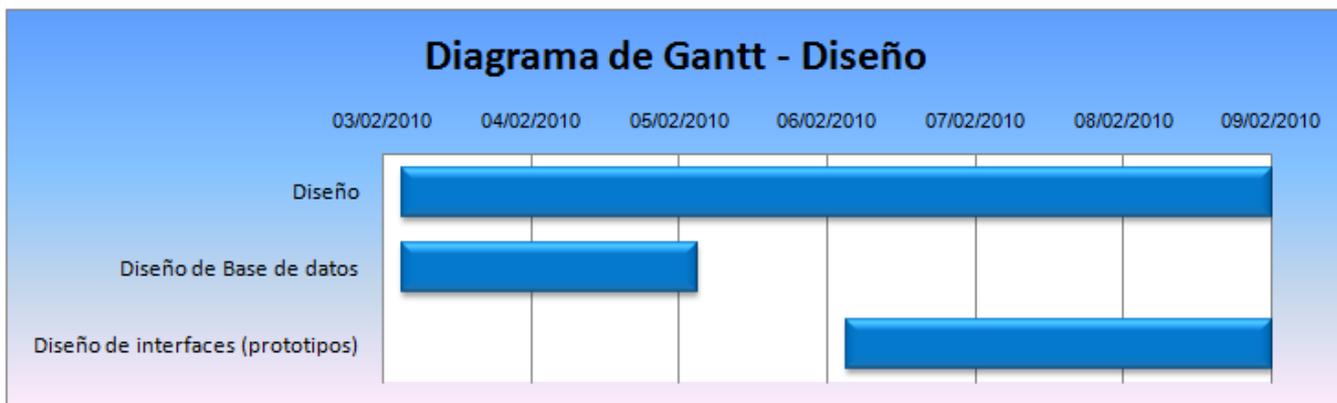
3.2.3.1.4 Requisitos

Todas las actividades son dependientes entre ellas cronológicamente y no se pueden realizar en paralelo.



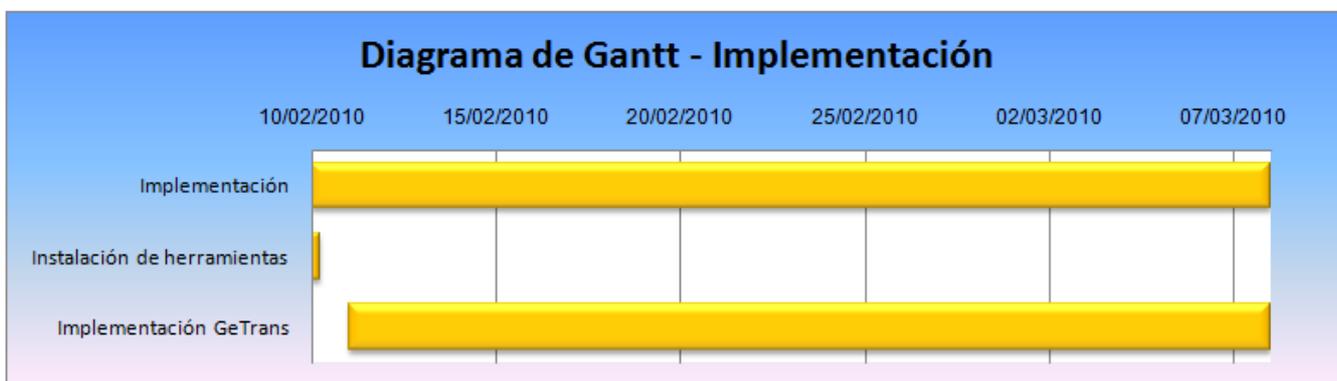
3.2.3.1.5 Diseño

Todas las actividades son dependientes entre ellas cronológicamente y no se pueden realizar en paralelo.



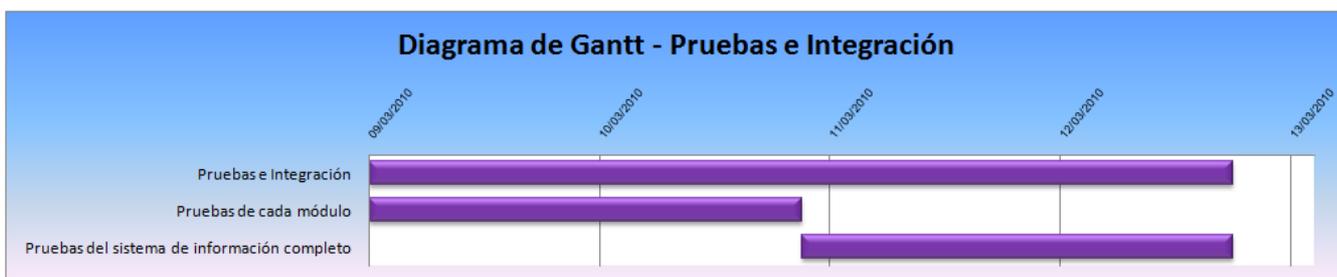
3.2.3.1.6 Implementación

Todas las actividades son dependientes entre ellas cronológicamente y no se pueden realizar en paralelo.



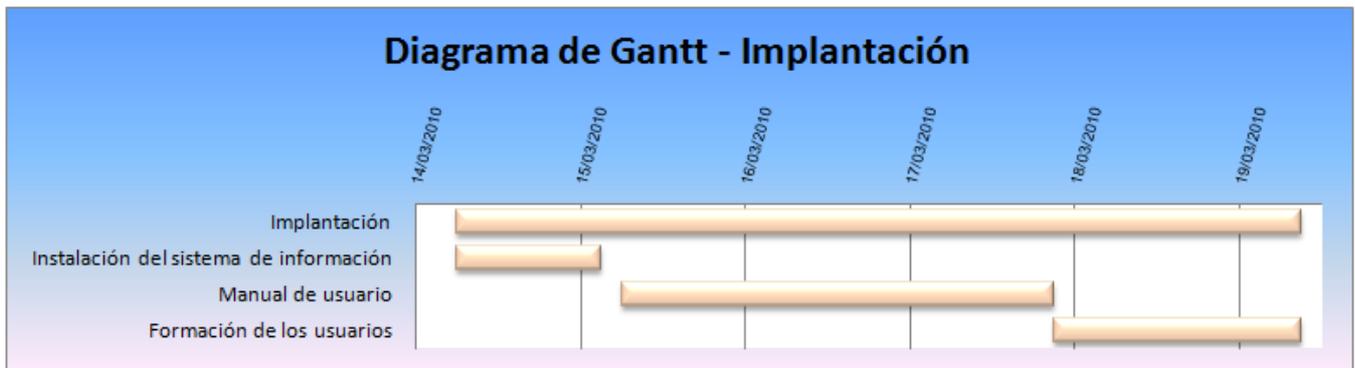
3.2.3.1.7 Pruebas e Integración

Todas las actividades son dependientes entre ellas cronológicamente y no se pueden realizar en paralelo.



3.2.3.1.8 Implantación

Todas las actividades son dependientes entre ellas cronológicamente y no se pueden realizar en paralelo.



3.2.3.1.9 Gestión del Proyecto

Hay actividades que se pueden llevar en paralelo. Además, la actividad "Seguimiento del proyecto" se va realizando a lo largo del proyecto.

Por otro lado, la actividad "Preparación de la defensa" requiere que todas las demás tareas y actividades estén realizadas, por este motivo, la barra de Gestión del Proyecto ocupa tanto tiempo, aunque haya tiempo inefectivo para esta tarea.



3.2.4 Criterios de terminación

Los criterios de terminación son establecidos para saber cuándo termina cada una de las tareas en las que se divide el proyecto. Para ello se redacta el correspondiente documento que debe ser aprobado por el cliente (FERCAEZ).

La primera actividad de la Tarea “Formación” denominada “Entrevista con el usuario” concluirá con la firma del contrato por parte del usuario final FERCAEZ, en el cual se detalla la descripción del sistema de información a desarrollar. Para el resto de actividades de esta tarea no hace falta establecer ningún criterio de terminación ya que el estudio y aprendizaje de las distintas tecnologías a utilizar es continuo puesto que se produce a lo largo de todo el proyecto. Por tanto finalizará con la conclusión del mismo.

Para la tarea “Gestión del proyecto” el mejor criterio de terminación es la memoria, ya que en ella se reflejan los aspectos que conllevan la gestión de un proyecto.

En cuanto a la tarea “Definición de objetivos del proyecto” se establece como criterio de terminación la aprobación definitiva del documento de objetivos del proyecto (DOP).

En la tarea “Requisitos” encontramos distintos criterios de terminación. En primer lugar la aprobación por parte de la empresa cliente FERCAEZ del documento de requerimientos del sistema (DOC) y posteriormente la aprobación del modelo de casos de uso.

La tarea de “Diseño” finalizará con la aceptación por parte de la empresa cliente FERCAEZ de: Prototipo de las interfaces, requisitos del sistema operativo, sistema de gestión de base de datos, tecnologías de acceso a datos.

La tarea de “Implementación” se dará por concluida a la vez que la de “pruebas” tras la comprobación del correcto funcionamiento del sistema de información.

Finalmente el proyecto con el cliente finaliza tras la implantación del sistema de información, es decir, instalación del sistema de información, adquisición por parte del usuario de los manuales y los ejecutables de la aplicación resultante (para su posible instalación en otros puestos de la empresa) y la formación básica que se impartirá a los usuarios. Todo ello quedará reflejado en el correspondiente documento firmado por ambas empresas y el proyectista.

El proyecto en su totalidad finalizará una vez concluida la defensa del mismo ante el tribunal en la facultad de informática de San Sebastián.

3.2.5 Valoraciones de riesgos (Plan de contingencia)

Los riesgos que se han establecido para este proyecto son:

Problema: Ampliación constante de los objetivos del proyecto provocando la ampliación de la planificación del proyecto, lo que puede llevar a que el proyecto se haga más complejo.

Solución: En el DOP se recogen los objetivos, duración, etapas y otros datos de interés que ayudarán al desarrollo del proyecto.

Problema: Errores de diseño, omisiones o malentendidos teniendo que volver a planificar en las fases afectadas.

Solución: Para evitar este riesgo suministramos el documento donde se recogen los requerimientos del sistema. Además se comunicará a la empresa cliente todas nuestras decisiones durante la fase de diseño.

Problema: La planificación se haya realizado demasiado ajustada y no de tiempo a completar los objetivos en las fechas marcadas.

Solución: Se planificará con mucha holgura para poder volver a planificar si es necesario. Además cabe la posibilidad de acortar algunos objetivos en la fase de implantación o implementación que se desarrollarán en futuras líneas de trabajo de la empresa.

Problema: La no disponibilidad del proyectista por bajas, enfermedades, motivos personales, etc .

Solución: En caso de baja del tutor de la Universidad se le asignará otro. En caso de baja del alumno cuando se recupere se meterán horas extras. En caso de que la baja sea por parte del contacto que tenemos en la empresa cliente FERCAEZ se asignará otra persona de contacto. Si aun aplicando estas soluciones no hubiera tiempo suficiente para cumplir todos los objetivos se procederá al cambio de objetivos del proyecto.

Problema: Realización de modificaciones sin tener en cuenta a qué pueden afectar. Esto puede provocar tener que volver a diseñar, volver a planificar, e incluso cambiar los objetivos del proyecto.

Solución: El cliente es responsable de definir los requerimientos y de aportar las ideas en la fase del diseño externo así como de ir aprobando cada uno de los acuerdos generados en cada una de las fases y en general a lo largo de todo el proyecto(Informe, hitos, resultados....).

Se recomienda que el cliente participe activamente en las reuniones para definir concienzudamente los requisitos y aprobar los diversos documentos que se vayan generando.

Problema: Pérdida de información.

Solución: Cada semana se realizará una copia de seguridad de la última versión que se posee(con las últimas modificaciones realizadas). Además se encontrará una copia en el puesto de la oficina (Sintec) y otro en el ordenador personal en casa.

A la hora de responder a cualquiera de estos riesgos se reaccionará aplicando instantáneamente la solución propuesta.

3.2.6 Supuestos claves del proyecto

Se contará con al menos un cliente de la empresa FERCAEZ que conozca todos los aspectos que debemos reflejar en la lógica de negocio y que mantenga las pertinentes reuniones con el proyectista.

La empresa FERCAEZ dispondrá al final del proyecto de tiempo suficiente para su formación en el sistema de información.

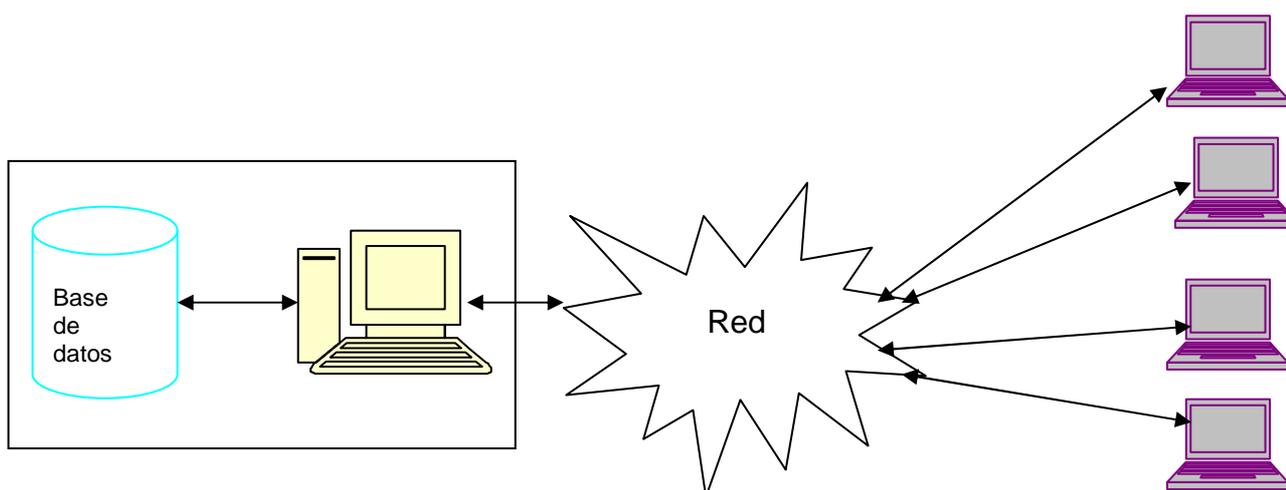
La introducción de datos en la Base de datos se realizará por parte de la empresa cliente.

3.3 Descripción del marco de trabajo

3.3.1 Flujo de Información

Como se ha comentado antes, el sistema de información está dividido en dos grandes módulos. Por un lado el programa cliente y por otro lado la base de datos. El programa cliente se puede instalar en tantos puestos como se desee, debiendo trabajar sobre la misma base de datos que puede residir en cualquier lugar de la red.

En el dibujo que se muestra a continuación (Fig5) se puede observar la distribución del sistema de información.



3.3.2 Recursos

Denominamos recursos toda herramienta, persona, material o cosa que se necesita para el desarrollo del proyecto.

3.3.2.1 Humanos

Cuando hablamos de recursos humanos nos referimos a las personas necesarias para el desarrollo del proyecto. Se han definido tres roles distintos con tareas independientes como son: jefe de proyecto, analista informático y programador.

El jefe de proyecto realiza tareas de gestión del proyecto, parte de la formación, definición de objetivos del proyecto, desarrollo de la memoria y preparación de la defensa.

El analista lleva a cabo los requisitos técnicos y funcionales, diseño y parte de la implementación.

Finalmente, el programador lleva a cabo parte de la formación, parte de la implementación, las pruebas del sistema de información, los manuales de usuario y parte de la formación de los futuros usuarios del sistema.

Todos estos roles van a ser asumidos por el alumno según la situación y labor a realizar.

3.3.2.2 Técnicos

Por recursos técnicos nos referimos tanto a las herramientas software, hardware como diversos materiales.

El hardware que se va a utilizar para la implantación de nuestro sistema de información está formado por varios ordenadores en red con sistema operativo Windows XP.

En cuanto al software las herramientas que se van a utilizar son: Visual Basic 6.0 y los componentes de Microsoft Office para la implementación. Setup Factory 9.0 para crear el programa de instalación. Microsoft Visual Source Safe para gestionar las versiones e iteraciones. Además se utilizará Microsoft Access para el desarrollo de la Base de datos, Ms Project para la planificación y Microsoft Word y Visio para el desarrollo de la memoria.

Finalmente, los materiales de oficina que se utilizarán serán pen-drivers, folios, carpetas, etc.

3.4 Factibilidad

Los recursos necesarios para llevar a cabo nuestros objetivos son los siguientes:

- Varios ordenadores con el sistema operativo Windows XP o superior instalados en red, en la oficina de la empresa.
- Microsoft Access 2003 para almacenar los datos de una manera centralizada, pudiendo radicar en cualquier máquina de red o en un servidor.
- Microsoft Visual Studio 6.0 tanto para el desarrollo como para la elaboración de determinados informes.
- Instalación de los componentes Word y Excel de Microsoft Office 2003 en los ordenadores donde se va a instalar GeTrans.
- Un programa cliente de administración de correo electrónico como Microsoft Outlook.
- Servidor SMTP para el envío de correos electrónicos de manera automática.
- Servicio de Fax.

Tenemos todos los recursos necesarios, por lo que consideramos factible la consecución de nuestros objetivos.

3.5 Desviación en la planificación

La desviación entre la planificación inicial y la real es considerable y ha sufrido una variación importante en cuanto a las fechas proyectadas inicialmente, sobre todo en la duración total de ciertas actividades del proyecto.

En la fase de implementación se ha producido un considerable aumento en su duración, debido a la cantidad de requisitos que se han solicitado por parte del cliente Fercaez y que sobrepasa la dimensión académica del proyecto.

También ha sufrido un incremento moderado sobre la estimación inicial, la fase de requisitos, diseño y pruebas debido al aumento de volumen de la aplicación.

Con respecto al resto de actividades se ha acertado en la planificación.

Como conclusión, la duración final del proyecto se ha visto afectada por la fase de la implementación de la aplicación, así como la suma del incremento de las fases de requisitos, diseño y pruebas. Por lo tanto se puede advertir que la duración del proyecto ha sufrido un notable retraso con respecto a la planificación inicial, algo que repercutirá en el coste final de la aplicación.

4. CAPTURA DE REQUISITOS

4.1 Usuarios del Sistema

El modelo de caso de uso va a presentar las funcionalidades que el sistema ofrece a los actores que interactúan con él (usuarios).

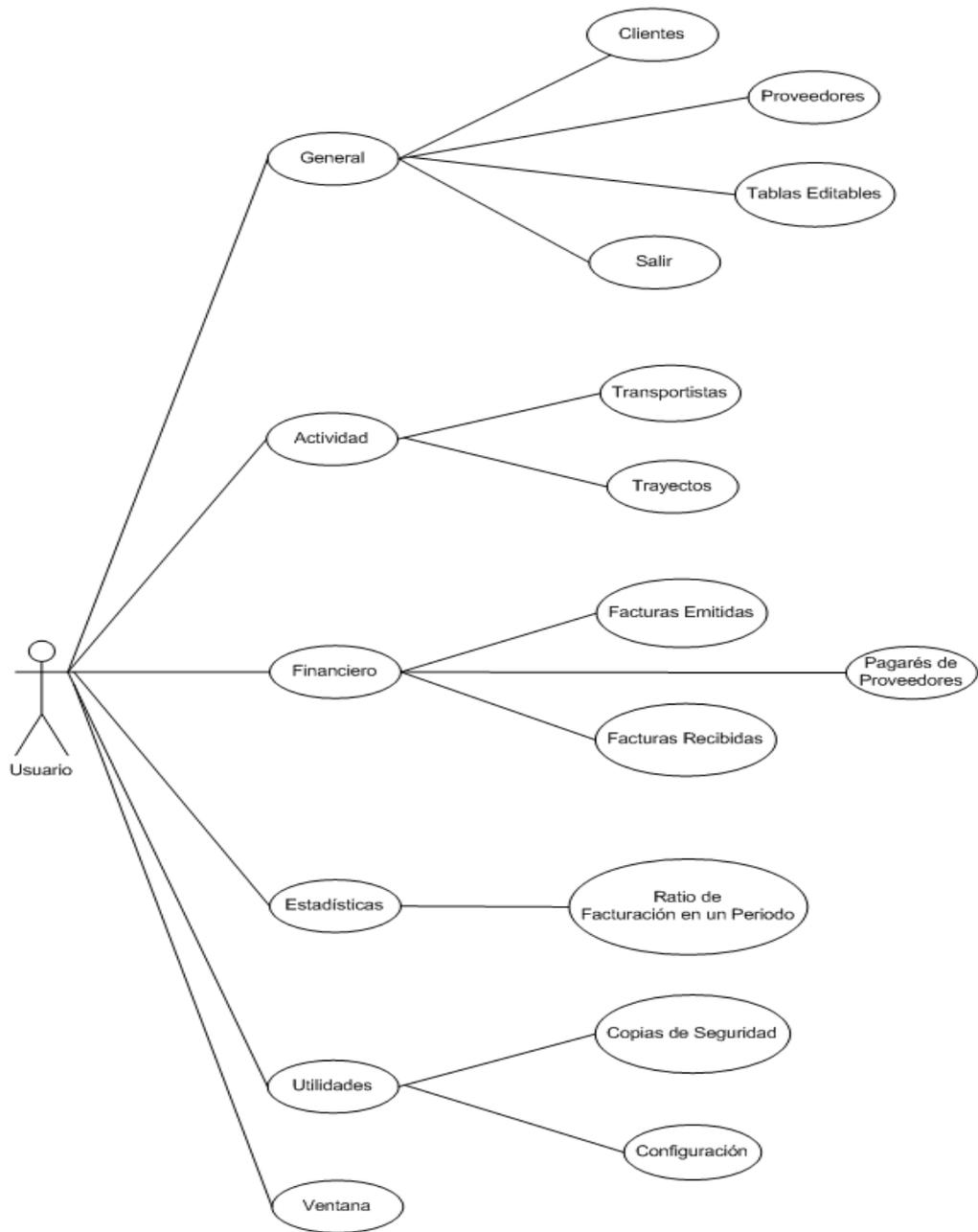
Para todo sistema de información encontramos unos actores. Actores son aquellas personas que van hacer uso de la aplicación.

En GeTrans únicamente es necesario identificar un actor que es el usuario. El usuario puede hacer uso de la funcionalidad completa de la aplicación.

4.2 Modelo de casos de uso de la aplicación

El modelo de caso de uso va a presentar las funcionalidades que el sistema ofrece a los actores que interactúan con él (usuarios). Para ello se realizará el diagrama de casos de uso para los distintos módulos que conforman el sistema junto con los prototipos de interfaz gráfico y el flujo de eventos (serie de acciones que el sistema lleva a cabo tras una acción del usuario).

En GeTrans únicamente es necesario identificar un actor que es el usuario. El usuario puede hacer uso de la funcionalidad completa de la aplicación. La aplicación se compone de varios módulos que agrupan de manera lógica, una serie de funcionalidades. Esta jerarquía se puede observar en el siguiente diagrama.



A continuación se detallan los módulos más importantes y representativos y las funcionalidades que contienen cada uno de ellos.

4.2.1 Módulo Actividad

En los siguientes apartados se detallan los diagramas de casos de uso y el flujo de eventos de las funcionalidades agrupadas de manera lógica en este módulo. Estas funcionalidades son Transportistas y Trayectos.

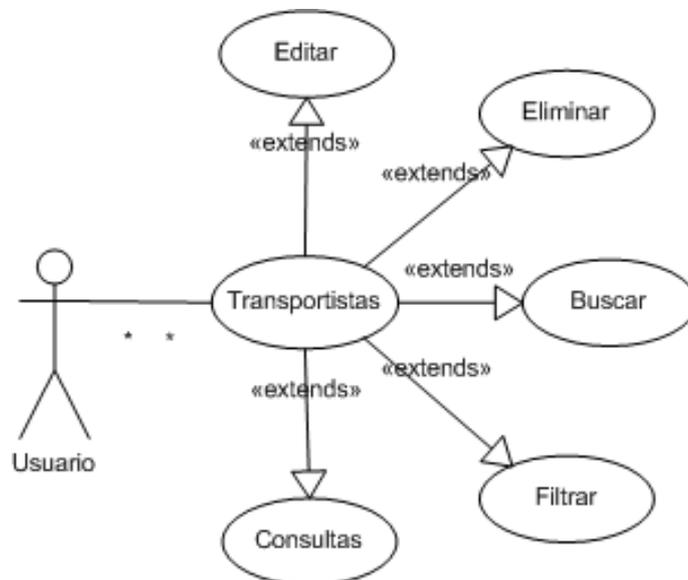
4.2.1.1 Caso de uso Transportistas

Este caso de uso tiene como base un **listado general de transportistas** sobre el que residen el resto de acciones del caso de uso.

En este caso de uso, se pueden realizar las siguientes acciones:

- Editar que consta en crear un nuevo transportista o modificar las propiedades de uno ya existente.
- Eliminar uno o varios transportistas de manera simultánea.
- Filtrar transportistas que cumplan un determinado patrón.
- Buscar transportistas que cumplan un determinado patrón
- Consultar información relacionada mediante consultas dinámicas.

4.2.1.1.1 Diagrama



4.2.1.1.2 Acciones

4.2.1.1.2.1 Editar

- **Actor:** Usuario
- **Descripción:** El usuario podrá crear un nuevo transportista o editar las propiedades de un cliente existente, en función del botón pulsado.
- **Precondiciones:** Si se pulsa el botón propiedades, debe seleccionarse un transportista existente.
- **Postcondiciones:** Una vez editadas las propiedades del transportista, estas ediciones quedarán reflejadas automáticamente en el listado general de transportistas.

4.2.1.1.2.2 Eliminar

- **Actor:** Usuario
- **Descripción:** Eliminar uno o varios transportistas del listado de transportistas.
- **Precondiciones:** Se debe seleccionar uno o varios transportistas.
- **Postcondiciones:** Una vez eliminado/s el/los transportista/s, desaparecerán del listado general de transportistas.

4.2.1.1.2.3 Filtrar

- **Actor:** Usuario
- **Descripción:** Permite seleccionar los transportistas que se van a mostrar en el listado de transportistas, en función de uno o varios criterios establecidos. Por defecto, se mostrará el valor Todos/as. Este valor anula el criterio de filtrado correspondiente.
- **Precondiciones:** Se debe pulsar el botón de Filtros.
- **Postcondiciones:** Se mostrarán los transportistas que cumplan con el valor establecido en los valores de filtrado.

4.2.1.1.2.4 *Buscar*

- **Actor:** Usuario
- **Descripción:** Permite buscar transportistas por coincidencia de patrón en determinadas columnas del listado. Una vez encontrado el primer transportista que cumple la coincidencia, pulsando el botón 'Buscar' el cursor se desplazará al siguiente transportista que cumpla la coincidencia. En caso de no encontrar ninguna coincidencia o de llegar al final del listado, la aplicación mostrará un aviso.

Se podrá introducir los caracteres comodín propios de Microsoft Access.

- **Precondiciones:** Pulsar el botón de Buscar para que aparezca la sección.
- **Postcondiciones:** El cursor se posicionará en el primer transportista que cumpla el criterio establecido. En caso de que no haya más transportistas que cumplan el criterio, se mostrará un mensaje.

4.2.1.1.2.5 *Consultas*

- **Actor:** Usuario
- **Descripción:** En el listado de transportistas se cargarán, de manera dinámica, las consultas diseñadas en base de datos con los criterios establecidos para cada consulta, en caso de que los hubiera. Una vez seleccionados los datos requeridos en la consulta, se mostrará un formulario que permitirá, por cada ítem mostrado, navegar al objeto destino, así como exportar todos los resultados a un nuevo libro Excel.
- **Precondiciones:** Seleccionar una consulta.
- **Postcondiciones:** Una vez establecidos los valores de los criterios, en caso de que los hubiera, se mostrará un formulario con los ítems que cumplan los criterios establecidos, permitiendo navegar al ítem destino, así como exportar todo el resultado a un nuevo libro Excel.

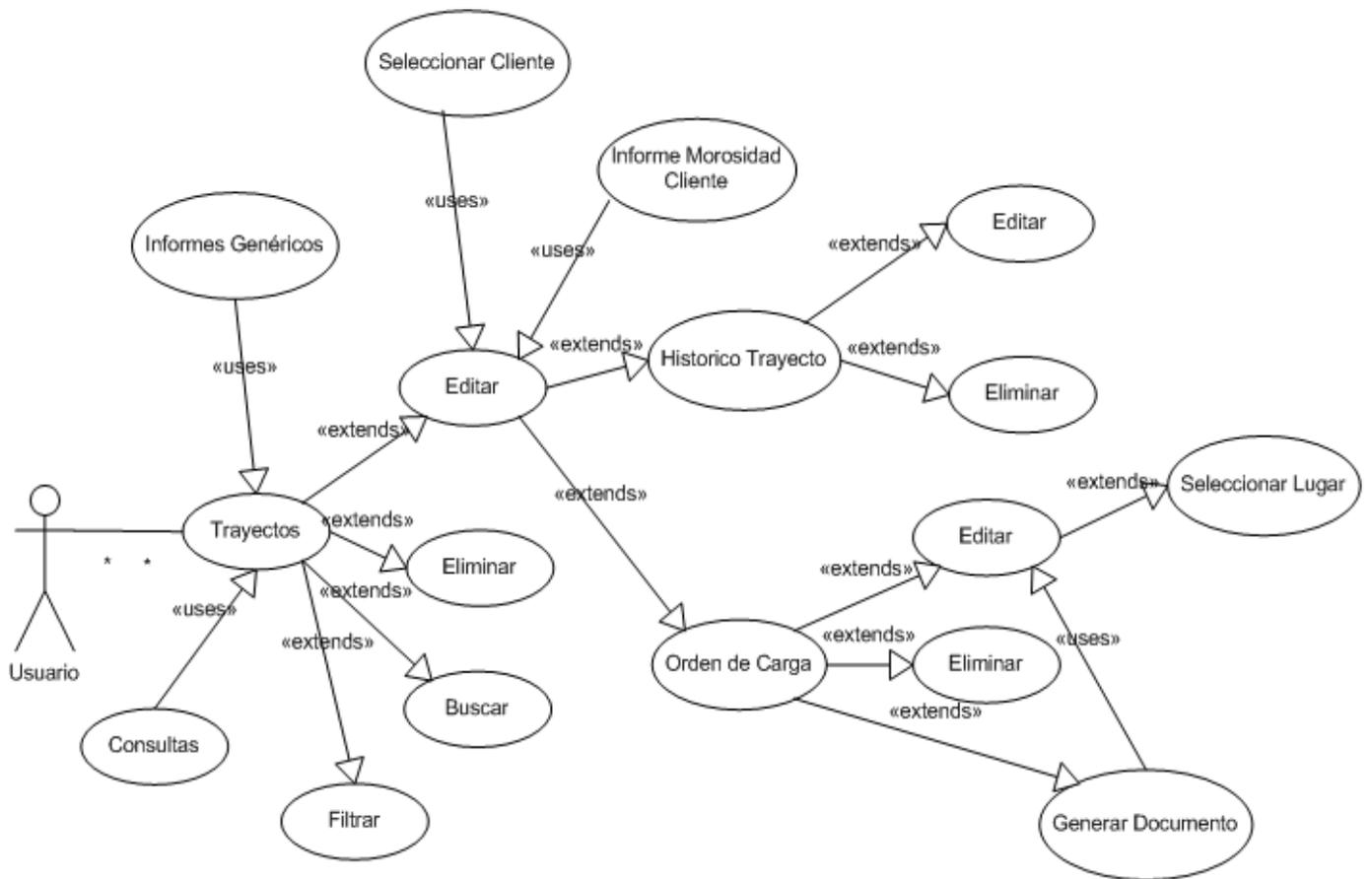
4.2.1.2 Caso de uso Trayectos

Este caso de uso tiene como base **un listado general de trayectos** sobre el que residen el resto de acciones del caso de uso.

Resaltar la complejidad de este caso de uso, ya que incluye otras entidades que hay que gestionar simultáneamente con trayectos y que ostentan sus propias acciones. En este caso de uso, se pueden realizar las siguientes acciones:

- Informes genéricos.
- Editar que consta en crear un nuevo transportista o modificar las propiedades de uno ya existente. Además, esta acción incluye las siguientes acciones:
 - Seleccionar Cliente
 - Informe Morosidad cliente
 - Histórico de trayecto que a su vez incluye las siguientes acciones:
 - Editar
 - Eliminar
 - Orden de carga del trayecto que a su vez incluye las siguientes acciones:
 - Editar
 - Eliminar
 - Seleccionar lugar
 - Generar documento
- Eliminar uno o varios transportistas de manera simultánea.
- Filtrar transportistas que cumplan un determinado patrón.
- Buscar transportistas que cumplan un determinado patrón
- Consultar información relacionada mediante consultas dinámicas.

4.2.1.2.1 Diagrama



4.2.1.2.2 Acciones

4.2.1.2.2.1 Informes Genéricos

- **Actor:** Usuario
- **Descripción:** El usuario podrá seleccionar el informe de trayectos en el que se mostrarán los trayectos que aparecen en el listado (filtrado o no). La información que aparecerá en el informe es Código, Fecha, Chofer, Origen, Destino, Precio, Total kms (VC) y €/km. Al final del listado se totaliza el nº de trayectos, el precio, el total kms (VC) y la media de €/km de los trayectos.
- **Precondiciones:** No aplica.
- **Postcondiciones:** El informe se podrá imprimir o exportar a cualquiera de los formatos configurados por el visualizador de informes.

4.2.1.2.2.2 *Editar*

- **Actor:** Usuario
- **Descripción:** El usuario podrá crear un nuevo trayecto o editar las propiedades de un trayecto existente, en función del botón pulsado.
- **Precondiciones:** Si se pulsa el botón propiedades, debe seleccionarse un trayecto existente.
- **Postcondiciones:** Una vez editadas las propiedades del trayecto, estas ediciones quedarán reflejadas automáticamente en el listado general de trayectos.

4.2.1.2.2.3 *Seleccionar Cliente*

- **Actor:** Usuario
- **Descripción:** Al editar las propiedades de un trayecto, para establecer el cliente del trayecto, se ofrecerá la posibilidad de seleccionarlo mediante una ventana con un listado abreviado de todos los clientes en los que se puede filtrar la búsqueda por TVA o por Empresa.
- **Precondiciones:** Deben existir clientes.
- **Postcondiciones:** Una vez seleccionado el cliente, en las propiedades del trayecto debe estar asignado el cliente seleccionado.

4.2.1.2.2.4 *Informe morosidad cliente*

- **Actor:** Usuario
- **Descripción:** Mostrar el nivel de morosidad de un determinado cliente a diferentes niveles de detalle.
- **Precondiciones:** Se debe seleccionar un cliente determinado para el trayecto.
- **Postcondiciones:** Una vez seleccionado el cliente, se mostrará un informe del nivel de morosidad general.

La pestaña Formas de Pago deberá mostrar el nivel de morosidad por cada diferente forma de pago asociadas a las facturas del cliente, así como el nº de día de demora en las facturas.

Una vez seleccionada una forma de pago, en la pestaña Detalle Forma Pago, se podrán visualizar todas aquellas facturas del cliente correspondiente a la forma de pago en las que se mostrará el nivel de morosidad por factura así como el nº de factura, fecha, vencimiento, importe y % de retraso como el nº de días de demora.

4.2.1.2.2.5 *Histórico de trayecto - Editar*

- **Actor:** Usuario
- **Descripción:** Crear un nuevo histórico de trayecto o editar las propiedades de un histórico de trayecto existente, en función del botón pulsado.
- **Precondiciones:** Se debe editar un trayecto.
- **Postcondiciones:** Una vez editadas las propiedades del histórico de trayecto, estas ediciones quedarán reflejadas automáticamente en el listado general de histórico de trayectos que estará incluido en las propiedades del trayecto.

Además, del trayecto se deberá actualizar la siguiente información: Origen (primer origen del histórico), Destino (último destino del histórico), Estado (el último del histórico), Tipo Transportista (el último del histórico), Transportista (el último del histórico), Tractor (el último del histórico), Remolque (el último del histórico), y la información kilométrica en función de todas las líneas del histórico del trayecto.

4.2.1.2.2.6 *Histórico de trayecto - Eliminar*

- **Actor:** Usuario
- **Descripción:** Eliminar una línea de histórico del trayecto del listado.
- **Precondiciones:**
 - Se debe editar un trayecto.
 - Se debe seleccionar una línea de histórico de trayecto.
- **Postcondiciones:** Una vez eliminada la línea de histórico de trayecto deberá desaparecer del listado histórico del trayecto.

Además, del trayecto se deberá actualizar la siguiente información: Origen (primer origen del histórico), Destino (último destino del histórico), Estado (el último del histórico), Tipo Transportista (el último del histórico), Transportista (el último del histórico), Tractor (el último del histórico), Remolque (el último del histórico), y la información kilométrica en función de todas las líneas del histórico del trayecto.

4.2.1.2.2.7 *Orden de carga – Seleccionar Lugar*

- **Actor:** Usuario
- **Descripción:** En la edición de una orden de carga, poder seleccionar una dirección asociada a un cliente determinado.
- **Precondiciones:**
 - Se debe editar un trayecto determinado
 - Deben existir el cliente.
 - Deben existir direcciones asociadas al cliente.
- **Postcondiciones:** La dirección seleccionada quedará asignada a la orden de carga.

4.2.1.2.2.8 Orden de carga – Generar Documento

- **Actor:** Usuario
- **Descripción:** En la edición de un trayecto o en las propiedades de la orden de carga, generar el documento OrdenDeCarga_<nnnnnn>.doc, donde nnnnnn es el nº de trayecto, en base a la plantilla PlantillaOrdenCarga.doc. A continuación, ofrecer la opción de visualizar el documento generado.
- **Precondiciones:**
 - Se debe editar un trayecto determinado
 - Deben existir el cliente.
 - Deben existir direcciones asociadas al cliente.
- **Postcondiciones:** La dirección seleccionada quedará asignada a la orden de carga.

4.2.1.2.2.9 Orden de carga - Editar

- **Actor:** Usuario
- **Descripción:** Editar las propiedades de una orden de carga para un trayecto determinado. Estas propiedades servirán para rellenar el documento OrdenDeCarga_<nnnnnn>.doc, donde nnnnnn es el nº de trayecto, en base a la plantilla PlantillaOrdenCarga.doc.
- **Precondiciones:** El trayecto debe existir.
- **Postcondiciones:** No aplica.

4.2.1.2.2.10 Orden de carga - Eliminar

- **Actor:** Usuario
- **Descripción:** Eliminar la orden de carga asociada a un trayecto determinado.
- **Precondiciones:** Se debe editar un trayecto existente.
- **Postcondiciones:** No aplica.

4.2.1.2.2.11 Eliminar

- **Actor:** Usuario
- **Descripción:** Eliminar uno o varios trayectos del listado de trayectos. Además deberá eliminar, si las hubiere, tanto la orden de carga como las líneas de histórico de trayecto asociadas.
- **Precondiciones:** Se debe seleccionar uno o varios trayectos.
- **Postcondiciones:** Una vez eliminado/s el/los trayecto/s, desaparecerá/n del listado general de trayectos.

4.2.1.2.2.12 *Filtrar*

Idem a la misma acción de otros casos de uso.

4.2.1.2.2.13 *Buscar*

Idem a la misma acción de otros casos de uso.

4.2.1.2.2.14 *Consultas*

Idem a la misma acción de otros casos de uso.

4.2.2 Módulo Financiero

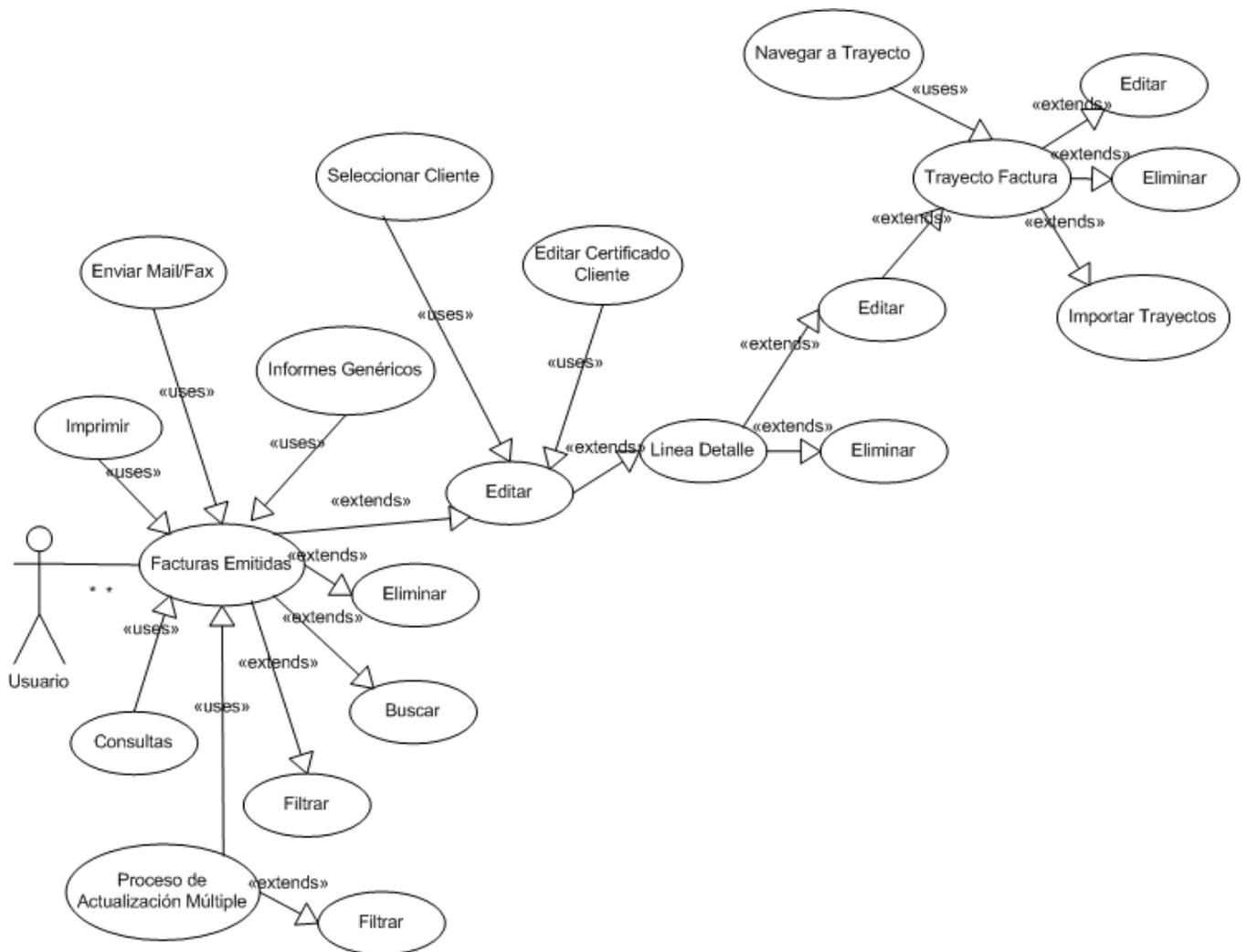
En los siguientes apartados se detallan los diagramas de casos de uso y el flujo de eventos de las funcionalidades agrupadas de manera lógica en este módulo. Estas funcionalidades son Facturas Emitidas, Facturas Recibidas y Pagarés de Proveedores.

4.2.2.1 Caso de uso Facturas Emitidas

Este caso de uso tiene como base **un listado general de facturas emitidas** sobre el que residen el resto de acciones del caso de uso. En este caso de uso, se pueden realizar las siguientes acciones:

- Imprimir. Imprimir una determinada factura por impresora o a Excel.
- Enviar Mail/Fax
- Informes Genéricos. Mostrar varios tipos de informe. Dos listados:
 - Listado Facturas
 - Listado Facturas agrupadas por cliente.
- Editar que consta en crear una nueva factura o en modificar las propiedades de una ya existente. En esta acción se pueden realizar las siguientes acciones:
 - Seleccionar cliente
 - Editar el certificado de un cliente
 - Línea Detalle que contiene las siguientes acciones:
 - Editar. Consta en crear una nueva línea de detalle de la factura o en modificar las propiedades de una ya existente. Se pueden realizar las siguientes acciones:
 - Trayecto Factura que contiene las siguientes acciones:
 - Navegar a Trayecto asociado.
 - Editar un trayecto asociado a la línea de detalle.
 - Eliminar un determinado trayecto.
 - Importar Trayectos no facturados del cliente.
 - Eliminar una línea de detalle.
- Eliminar una determinada factura.
- Ver Facturas. Todas/Pendientes/Retrasadas/Cobradas.
- Filtrar facturas que cumplan unos determinados criterios.
- Buscar facturas que cumplan un determinado patrón.
- Consultar información relacionada mediante consultas dinámicas.

4.2.2.1.1 Diagrama



4.2.2.1.2 Acciones

4.2.2.1.2.1 Imprimir

- **Actor:** Usuario
- **Descripción:** El usuario podrá imprimir una determinada factura por impresora, o generar la factura en Excel con el nombre Factura_<aaaa>_<nnnnnnnn>.xls, donde aaaa corresponderá al año en que se genera la factura y nnnnnnnn al formato de nº de factura.
- **Precondiciones:** Seleccionar una determinada factura.
- **Postcondiciones:** Imprimir la factura o visualizar la factura en Excel en función de la opción seleccionada.

4.2.2.1.2.2 *Enviar Mail/Fax*

- **Actor:** Usuario
- **Descripción:** Mostrar agrupadas por cliente, todas las facturas retrasadas, pudiéndose seleccionar únicamente el cliente para enviar mediante Fax o Mail la reclamación correspondiente. La reclamación se hará en base a la plantilla PlantillaCastellano.doc, PlantillaFrances.doc, PlantillaIngles.doc en función del idioma asignado al cliente.
- **Precondiciones:** Seleccionar la vista de Facturas Retrasadas.
- **Postcondiciones:** Debe mostrar en una nueva ventana, un listado de las facturas retrasadas agrupadas por cliente. Se podrá seleccionar los clientes que se consideren oportunos y el envío de la reclamación por Mail o por Fax. Además, generará por cliente, un documento EnvioMail_<nnnn>_<aaaa>_<mm>_<dd>.doc, en base a la plantilla correspondiente comentada en el apartado descripción, donde nnnn es el formato del nº de cliente, aaaa el formato del año, mm el formato del mes y dd el formato del día en que se genera el documento.

4.2.2.1.2.3 *Informes Genéricos - Facturas*

- **Actor:** Usuario
- **Descripción:** El usuario podrá seleccionar el informe de facturas en el que se mostrarán las facturas que aparecen en el listado (según la vista seleccionada y filtrado o no). La información que se mostrará en el informe es Nº Factura, Fecha, Cliente, %IVA, Total, Total Factura, Vencimiento y Forma de Pago. Al final del listado se mostrará la cantidad de facturas, y la suma de la columna Total y Total Factura.
- **Precondiciones:** No aplica.
- **Postcondiciones:** El informe se mostrará en el visualizador correspondiente. Se podrá imprimir o exportar a cualquiera de los formatos configurados por el visualizador de informes.

4.2.2.1.2.4 *Informes Genéricos – Facturas agrupadas por cliente*

- **Actor:** Usuario
- **Descripción:** El usuario podrá seleccionar el informe de facturas agrupadas por cliente en el que se mostrarán las facturas que aparecen en el listado (según la vista seleccionada y filtrado o no). La información que se mostrará en el informe es el nombre de cliente y para cada cliente Nº Factura, Fecha, %IVA, Total, Total Factura, Vencimiento y Forma de Pago. Por cada cliente se mostrará un resumen indicando el nº de facturas, así como la suma de Total Facturas. Al final del listado se mostrará la cantidad global de facturas, y la suma global de Total y Total Factura.
- **Precondiciones:** No aplica.
- **Postcondiciones:** El informe se mostrará en el visualizador correspondiente. Se podrá imprimir o exportar a cualquiera de los formatos configurados por el visualizador de informes.

4.2.2.1.2.5 *Editar*

- **Actor:** Usuario
- **Descripción:** El usuario podrá crear una factura o editar las propiedades de una factura existente, en función del botón pulsado.
- **Precondiciones:** Si se pulsa el botón propiedades, debe seleccionarse una factura existente.
- **Postcondiciones:** Una vez editadas las propiedades de la factura, estas ediciones quedarán reflejadas automáticamente en el listado general de facturas.

4.2.2.1.2.6 *Editar - Seleccionar Cliente*

- **Actor:** Usuario
- **Descripción:** Al editar las propiedades de una factura emitida, para establecer el cliente de la factura emitida, se ofrecerá la posibilidad de seleccionarlo mediante una ventana con un listado abreviado de todos los clientes en los que se puede filtrar la búsqueda por TVA o por Empresa.
- **Precondiciones:** Deben existir clientes.
- **Postcondiciones:** Una vez seleccionado el cliente, en las propiedades de la factura emitida debe estar asignado el cliente seleccionado.

4.2.2.1.2.7 *Editar – Certificado Cliente*

- **Actor:** Usuario
- **Descripción:** Cuando se selecciona un cliente, se mostrará un aviso del estado del certificado de cliente (si no lo tiene o si está caducado) para los clientes nacionales.
- **Precondiciones:**
 - Deben existir clientes.
 - Se debe seleccionar un cliente.
- **Postcondiciones:** Se muestra el aviso correspondiente que permitirá editar el certificado del cliente.

4.2.2.1.2.8 *Editar – Línea Detalle - Editar*

- **Actor:** Usuario
- **Descripción:** El usuario podrá crear una nueva línea de detalle o editar las propiedades de una línea de detalle para la factura.
- **Precondiciones:** Si se pulsa el botón propiedades, debe seleccionarse una línea de detalle existente.
- **Postcondiciones:** Una vez editadas las propiedades de la línea de detalle, estas ediciones quedarán reflejadas automáticamente en el listado de líneas de detalles correspondientes a la factura.

4.2.2.1.2.9 *Editar – Línea Detalle – Editar – Trayectos - Navegar a Trayecto*

- **Actor:** Usuario
- **Descripción:** El usuario podrá navegar en modo consulta a las propiedades de un trayecto determinado.
- **Precondiciones:** El trayecto debe existir.
- **Postcondiciones:** No aplica.

4.2.2.1.2.10 *Editar – Línea Detalle – Editar – Trayectos - Editar*

- **Actor:** Usuario
- **Descripción:** El usuario podrá crear, un nuevo trayecto o editar las propiedades de un trayecto determinado, asociado a una línea de detalle asociada a una factura.
- **Precondiciones:** Si se pulsa el botón propiedades, debe seleccionarse una línea de detalle existente.
- **Postcondiciones:** Una vez editadas las propiedades de la línea de detalle, estas ediciones quedarán reflejadas automáticas en el listado de líneas de detalles correspondientes a la factura.

4.2.2.1.2.11 *Editar – Línea Detalle – Editar- Trayectos- Importar Trayectos*

- **Actor:** Usuario
- **Descripción:** El usuario podrá importar en la línea de detalle actual, los trayectos del cliente no facturados.
- **Precondiciones:**
 - Seleccionar un cliente determinado.
 - Editar una línea de detalle.
- **Postcondiciones:** Mostrar en una ventana los trayectos no facturados del cliente. El usuario podrá seleccionar los que desea importar. Una vez importados, los trayectos quedarán asociados a la línea de detalle que deberá acumular el importe de todos los trayectos asociados.

4.2.2.1.2.12 *Editar – Línea Detalle – Eliminar*

- **Actor:** Usuario
- **Descripción:** El usuario podrá eliminar una o varias líneas de detalle del listado que está incluido en la factura.
- **Precondiciones:** Se debe seleccionar una o varias líneas de detalle.
- **Postcondiciones:** Una vez eliminada/s la/s línea/s de detalle, desaparecerán del listado. Esto supone que deben quedar desasociados (trayectos importados) o eliminadas (trayectos editados manualmente) en su caso, los trayectos asociados la línea.

4.2.2.1.2.13 *Eliminar*

- **Actor:** Usuario
- **Descripción:** Eliminar una factura del listado de factura.
- **Precondiciones:** Se debe seleccionar una factura del listado.
- **Postcondiciones:** Una vez eliminada la factura, desaparecerá del listado general de facturas.

4.2.2.1.2.14 *Filtrar*

Idem a la misma acción de otros casos de uso.

4.2.2.1.2.15 *Buscar*

Idem a la misma acción de otros casos de uso.

4.2.2.1.2.16 *Consultas*

Idem a la misma acción de otros casos de uso.

4.2.2.2 Caso de uso Facturas Recibidas

Este caso de uso tiene como base **un listado general de facturas recibidas** sobre el que residen el resto de acciones del caso de uso.

En este caso de uso, se pueden realizar las siguientes acciones:

- Editar que consta en crear una nueva factura recibida o en modificar las propiedades de una ya existente. En esta acción se pueden realizar las siguientes acciones:
 - Seleccionar proveedor
 - Editar el certificado de un proveedor
 - Importar Trayectos del proveedor. Permite importar a la factura, trayectos no asociados a ninguna factura del proveedor. En caso de no tener ningún trayecto pendiente, permite importar cualquier trayecto no facturado.
 - Navegar a Trayecto en modo Consulta
 - Desvincular Trayectos
- Eliminar una determinada factura recibida.
- Filtrar facturas recibidas que cumplan unos determinados criterios.
- Buscar facturas recibidas que cumplan un determinado patrón.
- Consultar información relacionada mediante consultas dinámicas.

4.2.2.2.1 Diagrama



4.2.2.2.2 Acciones

4.2.2.2.2.1 Editar

- **Actor:** Usuario
- **Descripción:** El usuario podrá crear una factura recibida o editar las propiedades de una factura recibida existente, en función del botón pulsado. Además, deberán vincularse a la factura recibida los trayectos vinculados que a su vez deberán constar como pagados.
- **Precondiciones:** Si se pulsa el botón propiedades, debe seleccionarse una factura recibida existente.
- **Postcondiciones:** Una vez editadas las propiedades de la factura recibida, estas ediciones quedarán reflejadas automáticamente en el listado general de facturas recibidas.

4.2.2.2.2 *Editar - Seleccionar Proveedor*

- **Actor:** Usuario
- **Descripción:** Al editar las propiedades de una factura recibida, para establecer el proveedor de la factura recibida, se ofrecerá la posibilidad de seleccionarlo mediante una ventana con un listado abreviado de todos los proveedores en los que se puede filtrar la búsqueda por TVA o por Empresa.
- **Precondiciones:** Deben existir proveedores.
- **Postcondiciones:** Una vez seleccionado el proveedor, en las propiedades de la factura recibida debe estar asignado el proveedor seleccionado.

4.2.2.2.3 *Editar – Certificado Proveedor*

- **Actor:** Usuario
- **Descripción:** Cuando se selecciona un proveedor, se mostrará un aviso del estado del certificado de proveedor (si no lo tiene o si está caducado) para los proveedores nacionales.
- **Precondiciones:**
 - Deben existir proveedores.
 - Se debe seleccionar un proveedor.
- **Postcondiciones:** Se muestra el aviso correspondiente que permitirá editar el certificado del proveedor.

4.2.2.2.4 *Editar – Importar Trayectos del Proveedor*

- **Actor:** Usuario
- **Descripción:** El usuario podrá incluir en la factura trayectos no facturados. De manera predeterminada, se mostrará una ventana con los trayectos no facturados asociados al proveedor (será el transportista del trayecto). En caso de que el proveedor no tuviera trayectos no facturados asociados, se mostrarían todos los trayectos no facturados de cualquier proveedor.
- **Precondiciones:**
 - Proveedores existentes.
 - Trayectos no facturados.
- **Postcondiciones:** Una vez importados los trayectos, aparecerán en el listado de trayectos de la factura recibida y se calculará el importe de la factura recibida con el importe subcontratado de los trayectos importados.

4.2.2.2.2.5 *Editar – Navegar a Trayecto en modo Consulta*

- **Actor:** Usuario
- **Descripción:** El usuario podrá navegar en modo consulta, a cualquier trayecto del listado de trayectos incluido en la factura recibida.
- **Precondiciones:** Deben existir trayectos incluidos en la factura recibida.
- **Postcondiciones:** Se mostrará una ventana en la que aparecerán las propiedades del trayecto en modo consulta (lectura).

4.2.2.2.2.6 *Editar – Desvincular Trayecto*

- **Actor:** Usuario
- **Descripción:** El usuario podrá desvincular un trayecto de la factura recibida. Además, el trayecto debe quedar como no facturado.
- **Precondiciones:** Debe seleccionar un trayecto.
- **Postcondiciones:** Una vez desvinculado el trayecto, desaparecerá del listado. Además, se calculará el importe de la factura emitida teniendo en cuenta el importe subcontratado del trayecto desvinculado.

4.2.2.2.2.7 *Eliminar*

- **Actor:** Usuario
- **Descripción:** Eliminar una factura recibida del listado de factura. Además, se deberán desvincular los trayectos que la factura recibida tuviera vinculados. Estos trayectos deberán constar como pendientes de pago.
- **Precondiciones:** Se debe seleccionar una factura recibida del listado.
- **Postcondiciones:** Una vez eliminada la factura recibida, desaparecerá del listado general de facturas recibidas.

4.2.2.2.2.8 *Filtrar*

- **Actor:** Usuario
- **Descripción:** Permite seleccionar las facturas recibidas que se van a mostrar en el listado de facturas, en función de uno o varios criterios establecidos. Por defecto, se mostrará el valor Todos/as. Este valor anula el criterio de filtrado correspondiente.
- **Precondiciones:** Se debe pulsar el botón de Filtros para que aparezca la sección.
- **Postcondiciones:** Se mostrarán las facturas recibidas que cumplan con el valor establecido en los valores de filtrado..

4.2.2.2.2.9 *Buscar*

Idem a la misma acción de otros casos de uso.

4.2.2.2.2.10 *Consultas*

Idem a la misma acción de otros casos de uso.

4.2.2.3 Caso de uso Pagarés a Proveedores

Este caso de uso tiene como base **un listado general de pagarés a proveedores** sobre el que residen el resto de acciones del caso de uso.

En este caso de uso, se pueden realizar las siguientes acciones:

- Imprimir. Imprimir un determinado pagaré a proveedor por impresora.
- Generar Documento Word. Rellenar la plantilla PagareProveedor.doc con los datos de un determinado pagaré a proveedor.
- Editar que consta en crear un nuevo pagaré a proveedor o en modificar las propiedades de uno ya existente. En esta acción se pueden realizar las siguientes acciones:
 - Seleccionar proveedor
 - Editar el certificado de un proveedor
 - Importar facturas recibidas pendientes del proveedor
 - Navegar a factura recibida
 - Desvincular facturas recibidas.
- Eliminar un determinado pagaré a proveedor.
- Filtrar pagarés a proveedor que cumplan unos determinados criterios.
- Buscar pagarés a proveedor que cumplan un determinado patrón.
- Consultar información relacionada mediante consultas dinámicas.

4.2.2.3.2.3 *Editar*

- **Actor:** Usuario
- **Descripción:** El usuario podrá crear un pagaré a proveedor o editar las propiedades de uno existente, en función del botón pulsado. Además, deberán vincularse al pagaré, las facturas recibidas del proveedor, que a su vez deberán constar como pagados.
- **Precondiciones:** Si se pulsa el botón propiedades, debe seleccionarse un pagaré existente.
- **Postcondiciones:** Una vez editadas las propiedades del pagaré, estas ediciones quedarán reflejadas automáticamente en el listado general de pagarés a proveedor.

4.2.2.3.2.4 *Editar - Seleccionar Proveedor*

Idem a la misma acción de otros casos de uso.

4.2.2.3.2.5 *Editar – Editar Certificado Proveedor*

Idem a la misma acción de otros casos de uso.

4.2.2.3.2.6 *Editar – Importar Facturas Pendientes de Proveedor*

- **Actor:** Usuario
- **Descripción:** El usuario podrá incluir facturas recibidas pendientes de pago en el pagaré. De manera predeterminada, se mostrará una ventana con las facturas recibidas pendientes de pago asociadas al proveedor. En caso de que el proveedor no tuviera facturas recibidas pendientes de pago, se mostrarán todas las facturas recibidas pendientes de pago de cualquier proveedor.
- **Precondiciones:**
 - Proveedores existentes.
 - Facturas pendientes de pago.
- **Postcondiciones:** Una vez importadas las facturas, aparecerán en el listado de facturas recibidas del pagaré y se calculará el importe del pagaré con el importe de las facturas vinculadas.

4.2.2.3.2.7 *Editar – Navegar a Factura Recibida de Proveedor*

- **Actor:** Usuario
- **Descripción:** El usuario podrá navegar en modo consulta, a cualquier factura recibida del listado de facturas recibidas incluidos en el pagaré.
- **Precondiciones:** Deben existir facturas recibidas vinculadas al pagaré.
- **Postcondiciones:** Se mostrará una ventana en la que aparecerán las propiedades de la factura recibida en modo consulta (lectura).

4.2.2.3.2.8 *Editar – Desvincular Factura*

- **Actor:** Usuario
- **Descripción:** El usuario podrá desvincular una factura recibida del pagaré. Además, esta factura recibida debe quedar como pendiente de pago.
- **Precondiciones:** Debe seleccionar una factura recibida.
- **Postcondiciones:** Una vez desvinculada la factura, desaparecerá del listado. Además, se calculará el importe del pagaré teniendo en cuenta el importe de la factura recibida desvinculada.

4.2.2.3.2.9 *Eliminar*

- **Actor:** Usuario
- **Descripción:** Eliminar un pagaré del listado de pagarés. Además, se deberán desvincular las facturas recibidas que el pagaré tuviera vinculados. Estas facturas recibidas deberán constar como pendientes de pago.
- **Precondiciones:** Se debe seleccionar un pagaré del listado de pagarés.
- **Postcondiciones:** Una vez eliminado el pagaré, desaparecerá del listado general de pagarés.

4.2.2.3.2.10 *Filtrar*

Idem a la misma acción de otros casos de uso.

4.2.2.3.2.11 *Buscar*

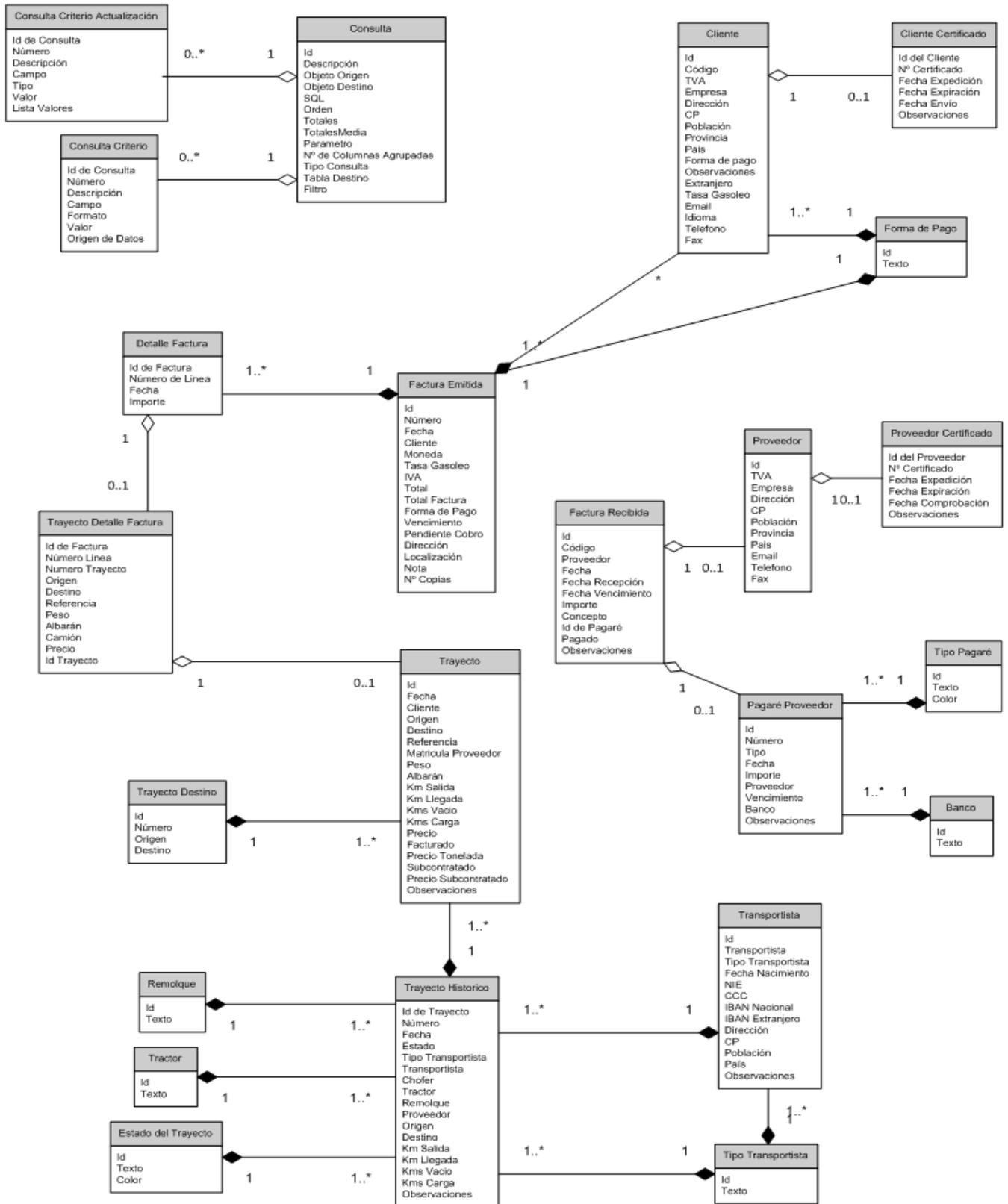
Idem a la misma acción de otros casos de uso.

4.2.2.3.2.12 *Consultas*

Idem a la misma acción de otros casos de uso.

5. MODELO DE DATOS

La siguiente imagen muestra el modelo conceptual que permite describir el conjunto de datos que manejará la aplicación.



6. DECISIONES TÉCNICAS

La elección de la solución tecnológica (lenguaje de la aplicación y productos adicionales) está condicionada al software existente en la empresa. Para no suponer un coste adicional de cambio de entorno (sistemas operativos, adquisición de nuevos productos), tanto en el plano económico como en lo que supone el cambio de hábito de trabajo, se ha concluido que lo mejor era adaptarse al entorno existente.

Por lo anteriormente descrito, se ha elegido para el desarrollo el lenguaje Visual Basic 6.0, por su integración en Windows XP, rapidez de ejecución (es un lenguaje compilado frente a .NET que es semi-compilado, semi-interpretado en ILT) y bajo requerimiento de recursos (disco y memoria RAM).

Para contener la base de datos y generar los documentos oportunos se opta por los productos de Microsoft Office Access, Word y Excel respectivamente, ya que son productos existentes en la empresa, por lo que no supone ningún coste económico adicional y con los que están familiarizados los futuros usuarios de GeTrans.

En este apartado se describen herramientas como las técnicas utilizadas para desarrollar la aplicación.

- **Microsoft Visual Basic 6.0**

Visual Basic es un lenguaje de programación dirigido por eventos, desarrollado por Alan Cooper para Microsoft. Este lenguaje de programación es un dialecto de BASIC, con importantes agregados. Su primera versión fue presentada en 1991, con la intención de simplificar la programación utilizando un ambiente de desarrollo completamente gráfico que facilitara la creación de interfaces gráficas y, en cierta medida, también la programación misma.

- **Visual Basic Data Report**

Se trata de unas librerías ActiveX escritas para Visual Basic, soportadas en tecnología ADO. Un DataReport se asimila mucho a un formulario, con su diseñador y todo. Sirve para crear informes. Data Report nos va a permitir crear niveles dentro del informe, totales, etc.

- **XML**

XML, siglas en inglés de *eXtensible Markup Language* ('lenguaje de marcas extensible'), es un lenguaje de marcas desarrollado por el World Wide Web Consortium (W3C). Deriva del lenguaje SGML y permite definir la gramática de lenguajes específicos para estructurar documentos grandes. XML no ha nacido sólo para su aplicación en Internet, sino que se propone como un estándar para el intercambio de información estructurada entre diferentes plataformas. Se puede usar en bases de datos, editores de texto, hojas de cálculo y casi cualquier cosa imaginable.

- **Microsoft MAPI Controls**

Se trata de unas librerías ActiveX escritas para Visual Basic, que sirven para enviar correos electrónicos.

- **Microsoft Visual Source Safe**

Microsoft Visual SourceSafe (también conocido por sus siglas VSS) es una herramienta de Control de versiones que forma parte de Microsoft Visual Studio. SourceSafe fue creado por la compañía americana One Tree Software a principios de los años noventa. Su primera versión fue la 3.1 coincidiendo con el mismo número de versión del sistema operativo Windows. Microsoft adquirió los derechos sobre SourceSafe que era un programa de 16 bits y lo liberó en 1995 como un programa de 32 bits, con la versión 4.0.

- **Setup Factory**

Setup Factory ofrece a los desarrolladores una solución para la creación de sistemas flexibles de instalación sin necesidad de aprender un lenguaje de programación propio. Setup Factory ofrece la personalización y funciones avanzadas de control que requieren los desarrolladores, todo ello desde el entorno de la fábrica de instalación de Diseño Visual. Incluso puede desplegar sus aplicaciones para servidores Web.

- **Microsoft Office**

Microsoft Office es una suite de oficina que abarca e interrelaciona aplicaciones de escritorio, servidores y servicios para los sistemas operativos Microsoft Windows y Mac OS X. Microsoft Office fue lanzado por Microsoft en 1989 para Apple Macintosh, Más tarde seguido por una versión para Windows, en 1990. La primera versión de Office contenía Microsoft Word, Microsoft Excel y Microsoft PowerPoint. Además, una versión "Pro" (profesional) de Office incluía Microsoft Access y Schedule Plus. Con el tiempo, las aplicaciones de Office han crecido sustancialmente y de forma más estrecha con características compartidas, como un corrector ortográfico común, la integración de datos OLE y el lenguaje de secuencias de comandos de Microsoft, Visual Basic para aplicaciones. Microsoft también posiciona Office como una plataforma de desarrollo para software de línea de negocios, bajo la marca de Office Business Applications (aplicaciones empresariales de Office u OBA por sus siglas en inglés).

7. DISEÑO

El diseño es la fase donde se define cómo se van a realizar las tareas analizadas.

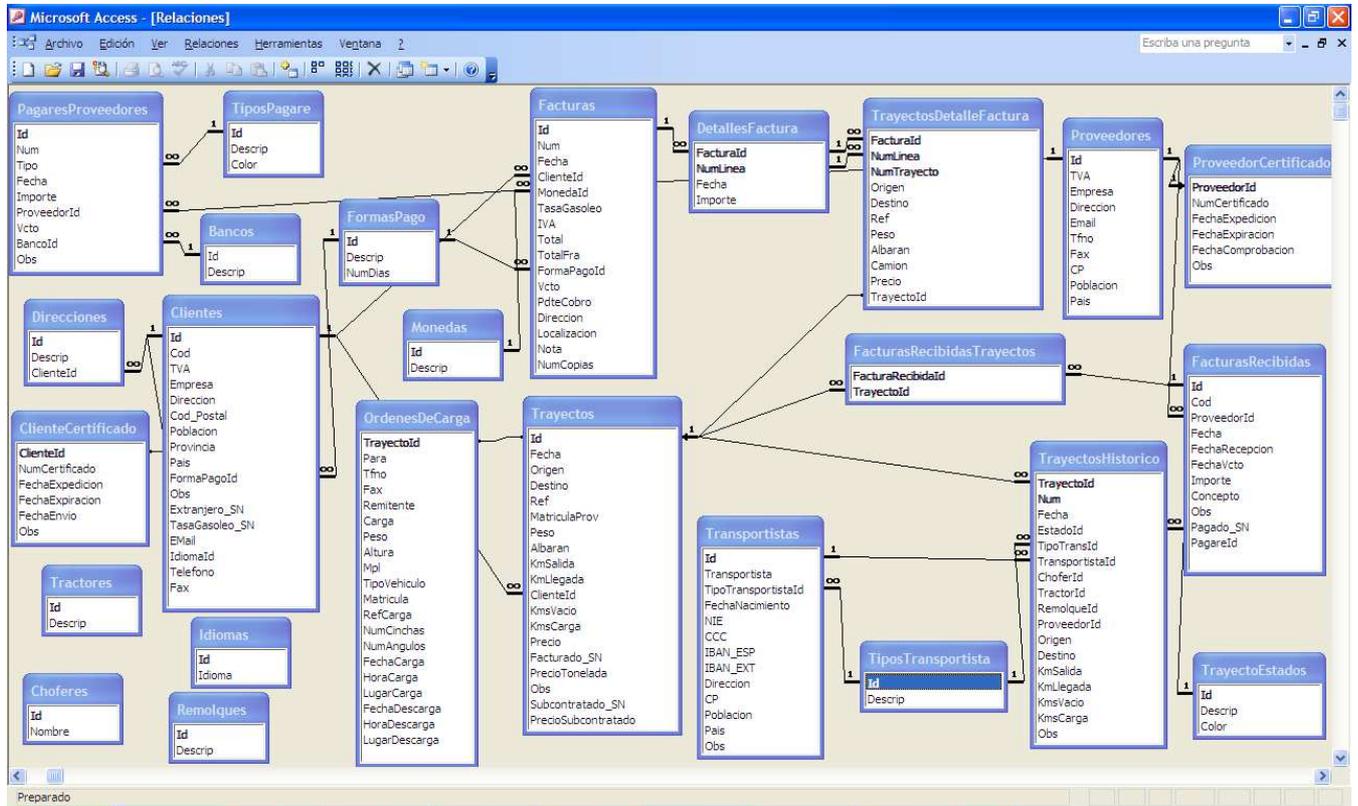
En esta capítulo se va a describir el diseño tanto de la base de datos como de las clases de los objetos más importantes (Transportistas, Trayectos, Facturas Emitidas, Facturas Recibidas y Pagares a Proveedores).

Los motivos de reflejar en el presente documento únicamente los objetos más importantes y representativos de la aplicación son los siguientes:

- No redundar en la explicación de la funcionalidad del resto de caso de uso que es muy similar a los explicados en el presente documento. Uno de los objetivos básicos en la aplicación es que resulte intuitiva y fácil de manejar por parte de los usuarios.
- El cliente GeTrans solicita más funcionalidades por parte de la aplicación por lo que la dimensión del proyecto excede de los límites académicos lógicos.

7.1 Diseño de la base de datos

A partir del modelo de datos conceptual y de los requerimientos solicitados, la estructura de la base de datos definitiva es la siguiente:



7.2 Diseño de las clases más significativas

7.2.1 Módulo Actividad

7.2.1.1 Clase Transportista

7.2.1.1.1 Propiedades

- Id → Identificador del Transportista. De tipo entero.
- Ids → Contiene una lista de identificadores de transportistas separados por coma. De tipo carácter.
- Transportista → El nombre del transportista. De tipo carácter.
- TipoTransportistaId → Identificador del tipo de transportista. De tipo entero.
- TipoTransportista → Descripción del tipo de transportista. De tipo carácter.
- FechaNacimiento → Fecha de nacimiento del transportista. De tipo fecha.
- NIE → El NIE del transportista. De tipo carácter.
- CCC → El CCC del transportista. De tipo carácter.
- IBAN_ESP → El IBAN para transportistas nacionales. De tipo carácter.
- IBAN_EXT → El IBAN para transportistas extranjeros. De tipo carácter.
- Dirección → La dirección del transportista. De tipo carácter.
- CP → El código postal del transportista. De tipo carácter.
- Población → La población donde reside el transportista. De tipo carácter.
- Pais → El país donde reside el transportista. De tipo carácter.
- Observaciones → Información adicional. De tipo carácter.
- ErrorDescripción → Descripción del error producido en algún método de la clase.

7.2.1.1.2 Métodos

- **Obtener**
 - **Descripción:**
Obtener las propiedades de un determinado transportista.
Se consultará en la base de datos las propiedades del transportista establecido en la propiedad Id y se refrescarán las propiedades de la clase. En caso de error, se establecerá la descripción del error en la propiedad ErrorDescripción.
 - **Recibe:** No aplica
 - **Devuelve:** No aplica.

- **ObtenerTodos**
 - **Descripción:**
Obtener las propiedades de todos los transportistas existentes.
En caso de error, se establecerá la descripción del error en la propiedad ErrorDescripción.
 - **Recibe:** Recibirá como argumento un parámetro opcional para ordenar la consulta.
 - **Devuelve:** Un recordset con el conjunto de transportistas obtenido.

- **Actualizar**
 - **Descripción:**
Actualizar las propiedades de un determinado transportista en la base de datos.
Previamente se habrán establecido las propiedades de la clase.
En caso de error, se establecerá la descripción del error en la propiedad ErrorDescripción.
 - **Recibe:** No aplica.
 - **Devuelve:** Un valor booleano. Verdadero si se ha ejecutado el procedimiento correctamente, falso en caso contrario.

- **Insertar**
 - **Descripción:**
Insertar las propiedades de un nuevo transportista en la base de datos. Previamente se habrán establecido las propiedades de la clase.
En caso de error, se establecerá la descripción del error en la propiedad ErrorDescripción.
 - **Recibe:** No aplica.
 - **Devuelve:** Un valor booleano. Verdadero si se ha ejecutado el procedimiento correctamente, falso en caso contrario.

- **Eliminar**
 - **Descripción:**
Eliminar uno o varios transportistas en la base de datos. Previamente se habrá establecido la lista de identificadores en la propiedad IDs de la clase.
En caso de error, se establecerá la descripción del error en la propiedad ErrorDescripción.
 - **Recibe:** No aplica.
 - **Devuelve:** Un valor booleano. Verdadero si se ha ejecutado el procedimiento correctamente, falso en caso contrario.

7.2.1.2 Clase Trayecto

7.2.1.2.1 Propiedades

- Id → Identificador del Trayecto. De tipo entero.
- Ids → Contiene una lista de identificadores de trayectos separados por coma. De tipo carácter.
- Fecha → Fecha del trayecto. De tipo fecha.
- Clienteld → Identificador del cliente asociado al trayecto. De tipo entero.
- Cliente → Nombre del cliente. De tipo carácter.
- Estadold → Identificador del estado del trayecto. De tipo entero.
- Estado → Descripción del estado del trayecto. De tipo carácter.
- TipoTransportistald → Identificador del tipo de transportista asociado al trayecto. De tipo entero.
- TipoTransportista → Descripción del tipo de transportista. De tipo carácter.
- Transportistald → Identificador del transportista asociado al trayecto. De tipo entero.
- Transportista → Nombre del transportista. De tipo carácter.
- TractorId → Identificador del tractor asociado al trayecto. De tipo entero.
- Tractor → Matrícula del tractor. De tipo carácter.
- Remolqueld → Identificador del remolque asociado al trayecto. De tipo entero.
- Remolque → Matrícula del remolque. De tipo carácter.
- Origen → Primer origen del trayecto. De tipo carácter.
- Destino → Último origen del trayecto. De tipo carácter.
- Referencia → Referencia del trayecto. De tipo carácter.
- Peso → Peso cargado en el trayecto. De tipo carácter.
- Albarán → Albarán asociado al trayecto. De tipo carácter.
- KmSalida → Kilómetro de salida del trayecto. De tipo entero.
- KmLlegada → Kilómetro de llegada del trayecto. De tipo entero.
- TotalKmsSL → Total kilómetros desde la salida hasta la llegada. De tipo entero.
- KmsVacio → Kilómetros con el remolque en vacío. De tipo entero.
- KmsCarga → Kilómetros con el remolque con carga. De tipo entero.
- TotalKmsVC → Total kilómetros vacío más carga. De tipo entero.
- Precio → Precio del trayecto. De tipo decimal.
- Facturado_SN → Indica si el trayecto ha sido facturado o no. De tipo booleano.
- Precio Tonelada → Indica el precio que cuesta cada tonelada de carga. De tipo decimal.
- Subcontratado_SN → Indica si el trayecto ha sido subcontratado. De tipo booleano.
- Precio Subcontratado → Precio para un trayecto subcontratado. De tipo decimal.
- Nº de Facturas → Nº de facturas asociadas al trayecto o trayectos. De tipo entero.
- Observaciones → Información adicional. De tipo carácter.
- ErrorDescripción → Descripción del error producido en algún método de la clase.

7.2.1.2.2 Métodos

- **Obtener**

- **Descripción:**

Obtener las propiedades de un determinado trayecto.

Se consultará en la base de datos las propiedades del trayecto establecido en la propiedad Id y se refrescarán las propiedades de la clase. En caso de error, se establecerá la descripción del error en la propiedad ErrorDescripción.

- **Recibe:** No aplica

- **Devuelve:** No aplica.

- **ObtenerTodos**

- **Descripción:**

Obtener las propiedades de todos los transportistas existentes.

En caso de error, se establecerá la descripción del error en la propiedad ErrorDescripción.

- **Recibe:** Recibirá como argumento los siguientes parámetros:

- NoFacturados. Parámetro opcional. Para obtener únicamente los trayectos no facturados.
 - Clienteld. Parámetro Opcional. Identificador de cliente para obtener sus trayectos asociados.
 - OrderBy: Parámetro Opcional. Para ordenar la consulta.

- **Devuelve:** Un recordset con el conjunto de trayectos obtenido.

- **Actualizar**

- **Descripción:**

Actualizar las propiedades de un determinado trayecto en la base de datos. Previamente se habrán establecido las propiedades de la clase.

En caso de error, se establecerá la descripción del error en la propiedad ErrorDescripción.

- **Recibe:** No aplica.

- **Devuelve:** Un valor booleano. Verdadero si se ha ejecutado el procedimiento correctamente, falso en caso contrario.

- **Insertar**
 - **Descripción:**

Insertar las propiedades de un nuevo trayecto en la base de datos. Previamente se habrán establecido las propiedades de la clase.

En caso de error, se establecerá la descripción del error en la propiedad ErrorDescripción.
 - **Recibe:** No aplica.
 - **Devuelve:** Un valor booleano. Verdadero si se ha ejecutado el procedimiento correctamente, falso en caso contrario.

- **Eliminar**
 - **Descripción:**

Eliminar uno o varios trayectos en la base de datos. Previamente se habrá establecido, para eliminar un único trayecto la propiedad ID, para eliminar varios una lista de identificadores en la propiedad IDs de la clase.

Además se eliminarán los TrayectosHistórico asociados al trayecto y se desvinculará la factura que estuviera vinculada al trayecto en la tabla TrayectosDetalleFactura.

Todas estas acciones de base de datos se realizarán de manera transaccional.

En caso de error, se establecerá la descripción del error en la propiedad ErrorDescripción.
 - **Recibe:** No aplica.
 - **Devuelve:** Un valor booleano. Verdadero si se ha ejecutado el procedimiento correctamente, falso en caso contrario.

- **ObtenerNumFacturas**
 - **Descripción:**

Obtener el nº de facturas asociadas a el trayecto/s seleccionado/s. La cantidad obtenida se establecerá a la propiedad NumFacturas del trayecto.

En caso de error, se establecerá la descripción del error en la propiedad ErrorDescripción.
 - **Recibe:** No aplica.
 - **Devuelve:** Un valor booleano. Verdadero si se ha ejecutado el procedimiento correctamente, falso en caso contrario.

7.2.2 Módulo Financiero

7.2.2.1 Clase Factura Emitida

7.2.2.1.1 Propiedades

- Id → Identificador de la factura. De tipo entero.
- Ids → Contiene una lista de identificadores de facturas separados por coma. De tipo carácter.
- Num → N° de la factura. De tipo entero.
- Fecha → Fecha del trayecto. De tipo fecha.
- Clienteld → Identificador del cliente asociado a la factura. De tipo entero.
- Cliente → Nombre del cliente. De tipo carácter.
- TasaGasoleo → Tasa de gasóleo a aplicar a la factura. De tipo decimal.
- IVA → Porcentaje de IVA a aplicar a la factura. De tipo decimal.
- Total → Total importe de la factura. De tipo decimal.
- Total Factura → Importe total de la factura una vez aplicado el IVA correspondiente. De tipo decimal.
- Forma De Pago Id → Identificador de la forma de pago asociada a la factura. De tipo entero.
- Forma De Pago → Descripción de la forma de pago asociada a la factura. De tipo carácter.
- Vencimiento → Vencimiento de la factura. De tipo fecha.
- Pendiente de Cobro → Determina si la factura está pendiente de cobro o no. De tipo booleano.
- Dirección → Dirección donde se envía la factura. De tipo carácter.
- Nota → Almacena información adicional. De tipo carácter.
- N° Copias → N° de copias para imprimir la factura.

7.2.2.1.2 Métodos

- **Obtener**
 - **Descripción:**
Obtener las propiedades de una determinada factura.
Se consultará en la base de datos las propiedades de la factura establecida en la propiedad Id y se refrescarán las propiedades de la clase. En caso de error, se establecerá la descripción del error en la propiedad ErrorDescripción.
 - **Recibe:** No aplica
 - **Devuelve:** No aplica.

- **ObtenerTodas**
 - **Descripción:**
Obtener las propiedades de todas las facturas existentes.
En caso de error, se establecerá la descripción del error en la propiedad ErrorDescripción.
 - **Recibe:** Recibirá como argumento los siguientes parámetros:
 - Where. Parámetro Opcional. La cláusula where de la consulta para restringir los datos.
 - OrderBy: Parámetro Opcional. Para ordenar la consulta.
 - **Devuelve:** Un recordset con el conjunto de facturas obtenido.

- **Actualizar**
 - **Descripción:**
Actualizar las propiedades de una determinada factura en la base de datos. Previamente se habrán establecido las propiedades de la clase.
En caso de error, se establecerá la descripción del error en la propiedad ErrorDescripción.
 - **Recibe:** No aplica.
 - **Devuelve:** Un valor booleano. Verdadero si se ha ejecutado el procedimiento correctamente, falso en caso contrario.

- **Insertar**
 - **Descripción:**
Insertar las propiedades de una nueva factura en la base de datos. Previamente se habrán establecido las propiedades de la clase.
En caso de error, se establecerá la descripción del error en la propiedad ErrorDescripción.
 - **Recibe:** No aplica.
 - **Devuelve:** Un valor booleano. Verdadero si se ha ejecutado el procedimiento correctamente, falso en caso contrario.

- **Eliminar**

- **Descripción:**

- Eliminar una o varias facturas en la base de datos. Previamente se habrá establecido, para eliminar una única factura la propiedad ID, para eliminar varios una lista de identificadores en la propiedad IDs de la clase.

- Además se eliminarán las filas de la tabla DetallesFactura y las filas de la tabla TrayectosDetalleFactura asociada/s a la/s factura/s a eliminar.

- Todas estas acciones de base de datos se realizarán de manera transaccional.

- En caso de error, se establecerá la descripción del error en la propiedad ErrorDescripción.

- **Recibe:** No aplica.

- **Devuelve:** Un valor booleano. Verdadero si se ha ejecutado el procedimiento correctamente, falso en caso contrario.

- **ExisteNumFacturaEnAño**

- **Descripción:**

- Determinar si existe el nº de factura en un año determinado. El valor del nº de factura vendrá definido en la propiedad Num de la factura. De esta manera se valida que el usuario no establezca dos nºs de factura iguales para el mismo año.

- En caso de error, se establecerá la descripción del error en la propiedad ErrorDescripción.

- **Recibe:** Recibirá como argumento los siguientes parámetros:

- Año. El año a examinar.

- **Devuelve:** Un valor booleano. Verdadero si existe el número, falso en caso contrario.

7.2.2.2 Clase Factura Recibida

7.2.2.2.1 Propiedades

- **Id** → Identificador de la factura recibida. De tipo entero.
- **Ids** → Contiene una lista de identificadores de facturas recibidas separados por coma. De tipo carácter.
- **Cod** → Código de la factura recibida. De tipo carácter.
- **ProveedorId** → Identificador del proveedor asociado a la factura recibida. De tipo entero.
- **Proveedor** → Nombre del proveedor. De tipo carácter.
- **Fecha** → Fecha de registro de la factura recibida. De tipo fecha.
- **FechaRecepción** → Fecha de recepción de la factura recibida. De tipo fecha.
- **FechaVencimiento** → Fecha de vencimiento de la factura recibida. De tipo fecha.
- **Importe** → Importe de la factura recibida. De tipo decimal.
- **Concepto** → Concepto de la factura recibida. De tipo carácter.
- **Pagado_SN** → Determina si la factura recibida se ha pagado o no. De tipo booleano.
- **PagareId** → Identificador del pagaré asociado a la factura recibida. De tipo entero.
- **Observaciones** → Información adicional. De tipo carácter.

7.2.2.2.2 Métodos

- **Obtener**
 - **Descripción:**

Obtener las propiedades de una determinada factura recibida.
Se consultará en la base de datos las propiedades de la factura recibida establecida en la propiedad **Id** y se refrescarán las propiedades de la clase. En caso de error, se establecerá la descripción del error en la propiedad **ErrorDescripción**.
 - **Recibe:** No aplica
 - **Devuelve:** No aplica.

- **ObtenerTodas**
 - **Descripción:**
Obtener las propiedades de todas o determinadas facturas recibidas.
En caso de error, se establecerá la descripción del error en la propiedad ErrorDescripción.
 - **Recibe:** Recibirá como argumento los siguientes parámetros:
 - Where. Parámetro Opcional. La cláusula where de la consulta para filtrar la consulta.
 - OrderBy: Parámetro Opcional. Para ordenar la consulta.
 - **Devuelve:** Un recordset con el conjunto de facturas recibidas obtenido.

- **ObtenerTrayectos**
 - **Descripción:**
Obtener todos los trayectos de una determinada factura recibida.
En caso de error, se establecerá la descripción del error en la propiedad ErrorDescripción.
 - **Recibe:** Recibirá como argumento los siguientes parámetros:
 - Where. Parámetro Opcional. La cláusula where de la consulta para filtrar la consulta.
 - **Devuelve:** Un recordset con el conjunto de trayectos obtenido. Vacío en caso de error.

- **Actualizar**
 - **Descripción:**
Actualizar las propiedades de una determinada factura recibida en la base de datos.
Previamente se habrán establecido las propiedades de la clase.
En caso de error, se establecerá la descripción del error en la propiedad ErrorDescripción.
 - **Recibe:** No aplica.
 - **Devuelve:** Un valor booleano. Verdadero si se ha ejecutado el procedimiento correctamente, falso en caso contrario.

- **Insertar**
 - **Descripción:**
Insertar las propiedades de una nueva factura recibida en la base de datos. Previamente se habrán establecido las propiedades de la clase.
En caso de error, se establecerá la descripción del error en la propiedad ErrorDescripción.
 - **Recibe:** No aplica.
 - **Devuelve:** Un valor booleano. Verdadero si se ha ejecutado el procedimiento correctamente, falso en caso contrario.

- **Eliminar**

- **Descripción:**

Eliminar una o varias facturas recibidas en la base de datos. Previamente se habrá establecido, para eliminar una única factura recibida la propiedad ID, para eliminar varios una lista de identificadores en la propiedad IDs de la clase.

Además se eliminarán las filas de la tabla FacturasRecibidasTrayectos (tabla donde se vinculan facturas recibidas y trayectos) asociada/s a la/s factura/s recibidas a eliminar.

Todas estas acciones de base de datos se realizarán de manera transaccional.

En caso de error, se establecerá la descripción del error en la propiedad ErrorDescripción.

- **Recibe:** No aplica.

- **Devuelve:** Un valor booleano. Verdadero si se ha ejecutado el procedimiento correctamente, falso en caso contrario.

- **DesvincularTrayecto**

- **Descripción:**

Desvincular uno o varios trayectos asociados a una factura recibida determinada (propiedad ID de la clase).

En caso de error, se establecerá la descripción del error en la propiedad ErrorDescripción.

- **Recibe:** Recibirá como argumento los siguientes parámetros:

- TrayectoID. El identificador de trayecto a desvincular de la factura recibida.

- **Devuelve:** Un valor booleano. Verdadero si se ha ejecutado el procedimiento correctamente, falso en caso contrario.

- **VincularTrayectos**

- **Descripción:**

Vincular una lista de trayectos a una factura recibida determinada (propiedad ID de la clase).

En caso de error, se establecerá la descripción del error en la propiedad ErrorDescripción.

- **Recibe:** Recibirá como argumento los siguientes parámetros:

- TrayectosID. Una lista de identificadores de trayectos.

- **Devuelve:** Un valor booleano. Verdadero si se ha ejecutado el procedimiento correctamente, falso en caso contrario.

- **ObtenerTrayectosSinFacturaDelProveedor**

- **Descripción:**

- Obtener todos los trayectos que no están asociados a una factura recibida del proveedor pasado como argumento.

- En caso de error, se establecerá la descripción del error en la propiedad ErrorDescripción.

- **Recibe:** Recibirá como argumento los siguientes parámetros:

- ProveedorID. El identificador de proveedor.

- **Devuelve:** Un recordset con el conjunto de trayectos obtenido. Vacío en caso de error.

- **ObtenerTrayectosSinFactura**

- **Descripción:**

- Obtener todos los trayectos que no están asociados a una factura recibida.

- En caso de error, se establecerá la descripción del error en la propiedad ErrorDescripción.

- **Recibe:** No aplica.

- **Devuelve:** Un recordset con el conjunto de trayectos obtenido. Vacío en caso de error.

7.2.2.3 Clase Pagaré a Proveedor

7.2.2.3.1 Propiedades

- Id → Identificador del pagaré a proveedor. De tipo entero.
- Ids → Contiene una lista de identificadores de pagaré a proveedor. De tipo carácter.
- Num → Numero de pagaré a proveedor. De tipo carácter.
- Tipo → Tipo de pagaré a proveedor (valores de la tabla TiposPagare). De tipo entero.
- Fecha → Fecha del pagaré a proveedor. De tipo fecha.
- Importe → Importe del pagaré a proveedor. De tipo decimal.
- ProveedorId → Identificador del proveedor asociado al pagaré. De tipo entero.
- Proveedor → Nombre del proveedor. De tipo carácter.
- Vencimiento → Vencimiento del pagaré. De tipo fecha.
- Bancold → Identificador del banco asociado al pagaré (Valores de la tabla Bancos). De tipo entero.
- Banco → Nombre del banco. De tipo carácter.
- N°Facturas → Número de facturas asociadas al pagaré. De tipo entero.
- Observaciones → Información adicional al pagaré. De tipo carácter.

7.2.2.3.2 Métodos

- **Obtener**
 - **Descripción:**
Obtener las propiedades de un determinado pagaré.
Se consultará en la base de datos las propiedades del pagaré establecido en la propiedad Id y se refrescarán las propiedades de la clase. En caso de error, se establecerá la descripción del error en la propiedad ErrorDescripción.
 - **Recibe:** No aplica
 - **Devuelve:** No aplica.
- **ObtenerTodos**
 - **Descripción:**
Obtener las propiedades de todos los pagarés.
En caso de error, se establecerá la descripción del error en la propiedad ErrorDescripción.
 - **Recibe:** Recibirá como argumento los siguientes parámetros:
 - **OrderBy:** Parámetro Opcional. Para ordenar la consulta.
 - **Devuelve:** Un recordset con el conjunto de pagarés obtenido. Vacío en caso de error.

- **ObtenerDatosDocumentoPagare**
 - **Descripción:**
Obtener todos los datos necesarios para rellenar el documento de pagaré a los proveedores establecido en la propiedad ID.
En caso de error, se establecerá la descripción del error en la propiedad ErrorDescripción.
 - **Recibe:** No aplica.
 - **Devuelve:** Un recordset con el conjunto de datos (facturas) para el pagaré. Vacío en caso de error.

- **ObtenerFacturasSinPagareDelProveedor**
 - **Descripción:**
Obtener todas las facturas recibidas que no están asociadas a ningún pagaré, del proveedor pasado como argumento.
En caso de error, se establecerá la descripción del error en la propiedad ErrorDescripción.
 - **Recibe:**
 - ProveedorId. Identificador del proveedor asociado a las facturas recibidas.
 - **Devuelve:** Un recordset con el conjunto de facturas recibidas asociadas al proveedor. Vacío en caso de error.

- **TieneFacturasAsociadas**
 - **Descripción:**
Determinar si un determinado pagaré (establecido en la propiedad ID) tiene facturas asociadas, para rellenar el documento correspondiente.
En caso de error, se establecerá la descripción del error en la propiedad ErrorDescripción.
 - **Recibe:** No aplica.
 - **Devuelve:** Un valor booleano. Verdadero si tiene facturas asociadas, falso en caso contrario o de error.

- **ObtenerFacturas**
 - **Descripción:**
Obtener todas las facturas asociadas al pagaré a proveedor.
En caso de error, se establecerá la descripción del error en la propiedad ErrorDescripción.
 - **Recibe:** No aplica.
 - **Devuelve:** Un recordset con el conjunto de facturas recibidas asociadas al pagaré. Vacío en caso de error

- **Actualizar**
 - **Descripción:**
Actualizar las propiedades de un determinado pagaré a proveedor en la base de datos. Previamente se habrán establecido las propiedades de la clase.
En caso de error, se establecerá la descripción del error en la propiedad ErrorDescripción.
 - **Recibe:** No aplica.
 - **Devuelve:** Un valor booleano. Verdadero si se ha ejecutado el procedimiento correctamente, falso en caso contrario.

- **Insertar**
 - **Descripción:**
Insertar las propiedades de un nuevo pagaré a proveedor en la base de datos. Previamente se habrán establecido las propiedades de la clase.
En caso de error, se establecerá la descripción del error en la propiedad ErrorDescripción.
 - **Recibe:** No aplica.
 - **Devuelve:** Un valor booleano. Verdadero si se ha ejecutado el procedimiento correctamente, falso en caso contrario.

- **ImportarFacturasRecibidas**
 - **Descripción:**
Importar (vincular) un/as factura/s recibida/s a un pagaré a proveedor.
En caso de error, se establecerá la descripción del error en la propiedad ErrorDescripción.
 - **Recibe:**
 - FacturasID: Una lista con los ids de las facturas recibidas a vincular.
 - **Devuelve:** Un valor booleano. Verdadero si se ha ejecutado el procedimiento correctamente, falso en caso contrario.

- **Eliminar**

- **Descripción:**

- Eliminar uno o varios pagarés a proveedores en la base de datos. Previamente se habrá establecido, para eliminar un único pagaré la propiedad ID, para eliminar varios una lista de identificadores en la propiedad IDs de la clase.

- Además se establecerán a no pagados (propiedad PagadoSN) y se desvincularán del pagaré (propiedad PagareId) las facturas recibidas asociada/s a la/s factura/s recibidas a eliminar.

- Todas estas acciones de base de datos se realizarán de manera transaccional.

- En caso de error, se establecerá la descripción del error en la propiedad ErrorDescripción.

- **Recibe:** No aplica.

- **Devuelve:** Un valor booleano. Verdadero si se ha ejecutado el procedimiento correctamente, falso en caso contrario.

- **DesvincularFactura**

- **Descripción:**

- Desvincular una o varias facturas recibidas de un determinado pagaré a proveedor por la propiedad ID, en la base de datos.

- En caso de error, se establecerá la descripción del error en la propiedad ErrorDescripción.

- **Recibe:**

- FacturasID: Una lista de identificadores de facturas recibidas. De tipo lista.

- **Devuelve:** Un valor booleano. Verdadero si se ha ejecutado el procedimiento correctamente, falso en caso contrario.

8. PRUEBAS DEL SISTEMA

En este apartado se muestra la validación realizada sobre cada una de las funcionalidades mostradas anteriormente. Se detalla las pruebas realizadas a las clases más importantes por los motivos explicados en el punto 7 Diseño, aunque la batería de pruebas se ha realizado a toda la aplicación.

Cada funcionalidad se ha sometido a baterías de pruebas en las que se ha experimentado un amplio abanico de situaciones. Cada grupo de pruebas se ha centrado en un único aspecto del sistema, lo que ha permitido emitir una valoración general sobre la fiabilidad del sistema y comprobar si cumple o no con las especificaciones que se identificaron al comienzo del proyecto.

Las pruebas se han realizado de forma manual, introduciendo la entrada desde la interfaz y comprobando, con la respuesta esperada de antemano, si la respuesta del sistema es correcta.

En el siguiente apartado se presenta el plan de pruebas donde se resume la metodología empleada y los resultados de todas las pruebas sobre los diferentes componentes del sistema.

8.1 Plan de pruebas

La metodología empleada en las pruebas del sistema usa el modelo de caja negra para verificar la fiabilidad del sistema. El objetivo principal es exponer condiciones que pongan a prueba las funcionalidades presentadas en el apartado 4.2 – Modelo de casos de uso de la aplicación.

La evaluación consistirá en la presentación de una acción del caso de uso a evaluar junto con el caso de prueba correspondiente y a continuación se mencionarán los resultados esperados junto con los obtenidos.

8.1.1 Casos de Prueba del Caso de Uso Transportistas

- **Editar**
 - **Acción:** El usuario pulsa el botón Nuevo, rellena la ficha de propiedades del transportista y pulsa aceptar.
 - **Resultado esperado:** Debe insertar el nuevo transportista en la tabla Transportistas de la base de datos y se actualiza el listado de transportistas mostrando las propiedades del nuevo transportista.
 - **Resultado obtenido:** Insertado 1 nuevo transportista que aparece en el listado de transportistas.

 - **Acción:** El usuario selecciona un transportista y pulsa el botón Propiedades, modifica sus propiedades en la ficha de propiedades del transportista y pulsa aceptar.
 - **Resultado esperado:** Debe actualizar las propiedades del transportista seleccionado en la tabla Transportistas de la base de datos y se actualiza el listado de transportistas mostrando sus propiedades actualizadas.
 - **Resultado obtenido:** Actualiza las propiedades del transportista seleccionado y aparecen sus propiedades actualizadas en el listado de transportistas.

- **Eliminar**
 - **Acción:** El usuario selecciona dos transportistas del listado y pulsa el botón de eliminar. Aparece una ventana de confirmación y el usuario pulsa aceptar.
 - **Resultado esperado:** Debe eliminar los transportistas seleccionados en la tabla Transportistas de la base de datos y se actualiza el listado de transportistas, desapareciendo los transportistas seleccionados.
 - **Resultado obtenido:** Eliminado dos transportistas que desaparecen del listado de transportistas.

- **Buscar**
 - **Acción:** El usuario pulsa el botón buscar, selecciona el campo Transportista e introduce el valor JOSE*. A continuación pulsa el botón Buscar.
 - **Resultado esperado:** Debe posicionarse en el primer transportista cuyo nombre empiece por JOSE.
 - **Resultado obtenido:** Se posiciona en el transportista 'JOSE ANTONIO SOARES PEREIRA DA LUZ' que es el primer transportista cuyo nombre empieza por JOSE.

- **Filtrar**
 - **Acción:** El usuario pulsa el botón filtrar que mostrará los campos disponibles para filtrar (Tipo y NIE), selecciona el tipo Chofer.
 - **Resultado esperado:** Debe posicionarse en el primer transportista de tipo chofer.
 - **Resultado obtenido:** Se posiciona en el primer transportista de tipo chofer.

- **Consultas**

- **Acción:** El usuario selecciona un transportista del listado de transportistas, y pulsa en el botón consultas, la consulta 'Trayectos del transportista en un periodo'. A continuación le aparece una ventana con los criterios establecidos para la consulta (Remolque, Fecha Inicio y Fecha Final). Selecciona los datos y pulsa Aceptar.
- **Resultado esperado:** Debe aparecer los trayectos realizados por el transportista con el remolque definido en el periodo de tiempo establecido, en la ventana general de consultas.
- **Resultado obtenido:** El resultado esperado.

8.1.2 Casos de Prueba del Caso de Uso Trayectos

- **Informes Genéricos**
 - **Acción:** El usuario pulsa el botón Informe en el listado de trayectos.
 - **Resultado esperado:** Debe aparecer un informe con todos los trayectos que aparecen en el listado. La información que se muestra por cliente es Código, Fecha, Chofer, Origen, Destino, Precio, Total kms (VC) y €/km. Al final del listado se totaliza el nº de trayectos, el precio, el total kms (VC) y la media de €/km de los trayectos.
 - **Resultado obtenido:** El resultado esperado.

- **Editar**
 - **Acción:** El usuario pulsa el botón Nuevo y pulsa el botón Cliente. Aparece un listado abreviado de todos los clientes en los que se puede filtrar la búsqueda por TVA o por Empresa. Selecciona la búsqueda por Empresa y se introduce en patrón PAP*. Aparecen las empresas que empiezan por PAP. El usuario selecciona la empresa PAPREC y pulsa Aceptar.
 - **Resultado esperado:** Debe quedar seleccionado en las propiedades del trayecto, el cliente PAPREC.
 - **Resultado obtenido:** El resultado esperado.

 - **Acción:** Una vez seleccionado el cliente en la acción anterior, el usuario pulsa Morosidad.
 - **Resultado esperado:** Debe aparecer el informe de morosidad (ver caso de uso clientes) del cliente PAPREC.
 - **Resultado obtenido:** El resultado esperado.

- **Histórico Trayecto Editar**
 - **Acción:** Una vez que el usuario está en la ficha de propiedades del trayecto, el usuario selecciona la pestaña Histórico y pulsa el botón Nuevo. Aparece las propiedades de la línea histórico del trayecto. El usuario rellena las propiedades de la línea del histórico del trayecto y pulsa Aceptar.
 - **Resultado esperado:** Debe insertar la nueva línea del histórico del trayecto y actualizar en el trayecto la siguiente información: Origen (primer origen del histórico), Destino (último destino del histórico), Estado (el último del histórico), Tipo Transportista (el último del histórico), Transportista (el último del histórico), Tractor (el último del histórico), Remolque (el último del histórico), y la información kilométrica en función de todas las líneas del histórico del trayecto.
 - **Resultado obtenido:** El resultado esperado.

- **Acción:** Una vez que el usuario está en la ficha de propiedades del trayecto, el usuario selecciona la pestaña Histórico, selecciona una línea del histórico del trayecto y pulsa el botón Propiedades. Aparece las propiedades de la línea histórico del trayecto. El usuario rellena las propiedades de la línea del histórico del trayecto y pulsa Aceptar.
 - **Resultado esperado:** Debe modificar las propiedades de la línea de histórico del trayecto y actualizar en el trayecto la siguiente información: Origen (primer origen del histórico), Destino (último destino del histórico), Estado (el último del histórico), Tipo Transportista(el último del histórico), Transportista (el último del histórico), Tractor (el último del histórico), Remolque (el último del histórico), y la información kilométrica en función de todas las líneas del histórico del trayecto.
 - **Resultado obtenido:** El resultado esperado.
- **Histórico Trayecto Eliminar**
 - **Acción:** Una vez que el usuario está en la ficha de propiedades del trayecto, el usuario selecciona la pestaña Histórico, selecciona una línea del histórico del trayecto y pulsa el botón Eliminar. Tras mensaje de confirmación de la acción, el usuario pulsa Aceptar.
 - **Resultado esperado:** Debe eliminar la línea de histórico del trayecto seleccionada y actualizar en el trayecto la siguiente información: Origen (primer origen del histórico), Destino (último destino del histórico), Estado (el último del histórico), Tipo Transportista(el último del histórico), Transportista (el último del histórico), Tractor (el último del histórico), Remolque (el último del histórico), y la información kilométrica en función de todas las líneas del histórico del trayecto.
 - **Resultado obtenido:** El resultado esperado.
- **Editar**
 - **Acción:** Una vez que el usuario ha establecido las líneas de histórico del trayecto pertinentes, pulsa Aceptar.
 - **Resultado esperado:** Debe insertar en las tablas TrayectosHistorico y Trayectos las propiedades del nuevo trayecto y aparecer en el listado de trayectos el nuevo trayecto.
 - **Resultado obtenido:** El resultado esperado.
 - **Acción:** El usuario selecciona un trayecto y pulsa el botón Propiedades, modifica sus propiedades en la ficha de propiedades del trayecto, así como las líneas del histórico del trayecto y pulsa aceptar.
 - **Resultado esperado:** Debe actualizar las propiedades del trayecto seleccionado en la tabla Trayectos, así como las actualizaciones realizadas en la tabla TrayectosHistorico de la base de datos y se actualiza el listado de trayectos mostrando sus propiedades actualizadas.
 - **Resultado obtenido:** Actualiza las propiedades del trayecto seleccionado, así como las líneas de histórico del trayecto y aparecen sus propiedades actualizadas en el listado de trayectos.

- **Orden de Carga**

- **Acción:** Una vez que el usuario está en la ficha de propiedades del trayecto, el usuario pulsa Editar Orden Carga. Aparece una ventana con las diferentes direcciones asociadas al cliente establecido en el trayecto. El usuario introduce *PAP* y selecciona la primera dirección.
- **Resultado esperado:** Debe aparecer la dirección seleccionada en las propiedades de la orden de carga.
- **Resultado obtenido:** El resultado esperado.

- **Acción:** En la edición de la orden de carga, el usuario rellena las propiedades de la orden de carga y pulsa Generar Documento. Aparece una ventana en la que se pregunta al usuario si desea visualizar el documento. El usuario pulsa Aceptar.
- **Resultado esperado:** Debe aparecer el documento Word basado en la plantilla PlantillaOrdenCarga.doc cuyo nombre es OrdenDeCarga_<nnnnnn>.doc que se guarda en la carpeta configurada, donde nnnnnn es el nº de trayecto, con los datos introducidos por el usuario. Además, en las propiedades del Trayecto aparece la opción Ver Documento Orden de Carga.
- **Resultado obtenido:** Se genera el documento OrdenDeCarga_001448.doc con los datos introducidos en la carpeta configurada, y aparece la opción Ver Documento Orden de Carga en las propiedades del trayecto.

- **Acción:** En las propiedades del trayecto, el usuario pulsa Ver Documento Orden de Carga.
- **Resultado esperado:** Tras pedir la confirmación del usuario, visualizar el documento OrdenDeCarga_001448.doc anteriormente generado.
- **Resultado obtenido:** El resultado esperado.

- **Acción:** En la edición de la orden de carga, el usuario rellena las propiedades de la orden de carga y pulsa Aceptar.
- **Resultado esperado:** Debe crear una orden de carga para el trayecto en la tabla OrdenesDeCarga. Además, en las propiedades del Trayecto aparece la opción Eliminar Orden de Carga.
- **Resultado obtenido:** Se genera el documento OrdenDeCarga_001448.doc con los datos introducidos en la carpeta configurada, y aparece la opción Eliminar Orden de Carga en las propiedades del trayecto.

- **Acción:** En las propiedades del trayecto, el usuario pulsa Eliminar Orden de Carga.
- **Resultado esperado:** Tras pedir la confirmación del usuario, eliminar la orden de carga generada en la tabla OrdenesDeCarga para el trayecto así como la opción Eliminar Orden de Carga de las propiedades del trayecto.
- **Resultado obtenido:** El resultado esperado.

- **Eliminar**
 - **Acción:** El usuario selecciona un trayecto del listado y pulsa el botón de eliminar. Aparece una ventana de confirmación y el usuario pulsa aceptar.
 - **Resultado esperado:** Debe eliminar el trayecto seleccionado de las tablas TrayectosHistorico, OrdenDeCarga (si la hubiere) y Trayectos, además se actualiza el listado de trayectos, desapareciendo el trayecto seleccionado.
 - **Resultado obtenido:** Eliminado el trayecto que desaparecen del listado de trayectos, así como de las tablas TrayectosHistorico y Trayectos de la base de datos.

- **Buscar**
 - **Acción:** El usuario pulsa el botón buscar, selecciona la columna Origen e introduce el valor F31*. A continuación pulsa el botón Buscar.
 - **Resultado esperado:** Debe posicionarse en el primer trayecto cuyo origen empiece por F31.
 - **Resultado obtenido:** Se posiciona en el primer trayecto cuyo origen es 'F31 BRUGUIERES'.

- **Filtrar**
 - **Acción:** El usuario pulsa el botón filtrar que mostrará los campos disponibles para filtrar (Tipo Transportista, Transportista, Tractor, Remolque, Cliente, Facturado), selecciona el tipo de transportista 'Autónomo' y el cliente 'PAPREC'.
 - **Resultado esperado:** Debe seleccionar aquellos trayectos cuyo transportista es de tipo 'Autónomo' y el cliente es 'PAPREC'.
 - **Resultado obtenido:** El resultado esperado.

- **Consultas**
 - **Acción:** El usuario pulsa el botón consultas, y selecciona la consulta 'Trayectos subcontratados en un periodo'. A continuación le aparece una ventana con los criterios establecidos para la consulta (Fecha Inicio y Fecha Final). Selecciona los datos y pulsa Aceptar.
 - **Resultado esperado:** Debe aparecer los trayectos subcontratados (atributo Subcontratado del trayecto) en el periodo de tiempo establecido, en la ventana general de consultas.
 - **Resultado obtenido:** El resultado esperado.

8.1.3 Casos de Prueba del Caso de Uso Facturas Emitidas

- **Imprimir**
 - **Acción:** El usuario selecciona una factura, pulsa el botón Imprimir y selecciona la opción 'A Excel'.
 - **Resultado esperado:** Tras aceptar la confirmación de visualización de la factura, debe mostrar la factura en Excel. El libro Excel tendrá el nombre Factura_<aaaa>_<nnnnnnnn>.xls, donde aaaa es el año y nnnnnnnn el nº de factura.
 - **Resultado obtenido:** Se genera y se visualiza correctamente la factura Factura_2009_00000530.xls.

 - **Acción:** El usuario selecciona una factura, pulsa el botón Imprimir y selecciona la opción 'A Impresora'.
 - **Resultado esperado:** Se deben imprimir por impresora la factura seleccionada, con tantas copias como se hayan definido para la factura (atributo Nº de Copias de la factura).
 - **Resultado obtenido:** Se imprimen 2 copias (la primera pone el texto original y la segunda copia) por la impresora.

- **Enviar Mail/Fax**
 - **Acción:** El usuario selecciona la vista de facturas retrasadas y pulsa el botón Enviar Mail/Fax que se encuentra en la barra de botones situado encima del Informe en el listado de proveedores.
 - **Resultado esperado:** Debe aparecer una nueva ventana con una lista de clientes y su relación de facturas retrasadas.
 - **Resultado obtenido:** El resultado esperado.

 - **Acción:** El usuario selecciona varios clientes de los que aparecen como resultado de la acción anterior. A continuación selecciona 'por correo electrónico' y pulsa el botón Enviar.
 - **Resultado esperado:** Debe rellenar la plantilla EnvioMail_<cccc>_<yyyy_mm_dd>.doc por cliente, donde cccc es el identificador de cliente con formato de 5 dígitos, yyyy es el año, mm el mes y dd el día de la generación del documento. Una vez rellenado el documento por cliente, envía un mail con los datos del mail que se han configurado (ver sección configuración mail) y adjunta el documento correspondiente.
 - **Resultado obtenido:** El resultado esperado.

- **Informes Genéricos**

- **Acción:** El usuario pulsa el botón Informe en el listado de facturas y selecciona la opción 'Listado de Facturas'.
- **Resultado esperado:** Debe aparecer un informe con las facturas que aparecen en el listado. La información que se muestra N° Factura, Fecha, Cliente, % iva, Total, Total Factura, Vencimiento y forma de pago. Al final del informe, se muestra la información de cuenta de facturas, suma de total y suma de total factura.
- **Resultado obtenido:** El resultado esperado.

- **Acción:** El usuario pulsa el botón Informe en el listado de facturas y selecciona la opción 'Listado de Facturas'.
- **Resultado esperado:** Debe aparecer un informe con las facturas que aparecen en el listado agrupadas por cliente. La información que muestra es el nombre de cliente y para las facturas w/ N° Factura, Fecha, Cliente, % iva, Total, Total Factura, Vencimiento y forma de pago. Al final de cada grupo se muestra el n° de facturas que tiene cada cliente así como el total factura de cada cliente. Al final del informe se muestra la información de cuenta de clientes, cuenta total de facturas, y el total de Total Factura.
- **Resultado obtenido:** El resultado esperado.

- **Editar**

- **Acción:** El usuario pulsa el botón Nuevo y pulsa el botón Buscar. Aparece un listado abreviado de todos los clientes en los que se puede filtrar la búsqueda por TVA o por Empresa. Selecciona la búsqueda por Empresa y se introduce en patrón PAP*. Aparecen las empresas que empiezan por PAP. El usuario selecciona la empresa PAPREC y pulsa Aceptar.
- **Resultado esperado:** Debe quedar seleccionado en las propiedades de la factura, el tva de cliente en el combo cliente, debe aparecer el nombre de cliente y la dirección completa del cliente en los datos del destinatario.
- **Resultado obtenido:** El resultado esperado.

- **Acción:** Una vez que el usuario ha seleccionado el cliente, si es un cliente no extranjero, comprueba si posee certificado. En caso de que no tenga certificado, avisa al usuario de tal circunstancia. Pulsando doble click en el aviso permite crear el certificado del cliente. Aparecerá una nueva ventana en la que el usuario deberá rellenar las propiedades del certificado y pulsar aceptar.
- **Resultado esperado:** Deberá insertar el nuevo certificado en la tabla ClienteCertificado de la base de datos, además deberá desaparecer el aviso en la factura, mostrando el nombre de cliente.
- **Resultado obtenido:** Insertado 1 nuevo certificado para el cliente seleccionado y desaparece el aviso mostrando el nombre del cliente.

- **Importar Trayectos**
 - **Acción:** Dentro de las propiedades de una línea de detalle determinada de una factura determinada, el usuario pulsa el botón Importar.
 - **Resultado esperado:** Debe aparecer un listado con los trayectos pendientes de facturar del cliente seleccionado en la factura.
 - **Resultado obtenido:** El resultado esperado.

 - **Acción:** Del listado de trayectos anterior, el usuario selecciona dos trayectos y pulsa aceptar.
 - **Resultado esperado:** Debe incluir en el listado de trayectos de la línea de detalle los trayectos seleccionados. Además, debe incrementar el importe de la línea de detalle teniendo en cuenta la suma de los importes de los trayectos importados.
 - **Resultado obtenido:** El resultado esperado.

- **Editar Trayecto**
 - **Acción:** El usuario pulsa el botón Nuevo del listado de trayectos de una línea de detalle determinada de una factura determinada. Aparece una ventana con las propiedades del trayecto y pulsa aceptar.
 - **Resultado esperado:** Debe aparecer el nuevo trayecto en el listado de trayectos de la línea de detalle y factura actual. Además, debe incrementar el importe de la línea de detalle en función del importe del nuevo trayecto.
 - **Resultado obtenido:** El resultado esperado.

- **Navegar a Trayecto**
 - **Acción:** El usuario pulsa el botón Propiedades en un trayecto determinado de una línea de detalle determinada de una factura determinada. Si el trayecto ha sido importado, tendrá asociado un objeto trayecto asociado y aparecerá el link 'Ver Trayecto'. El usuario pulsa el link 'Ver Trayecto'.
 - **Resultado esperado:** Debe aparecer una nueva ventana con las propiedades del trayecto asociado en modo consulta.
 - **Resultado obtenido:** El resultado esperado.

- **Eliminar Trayecto**
 - **Acción:** El usuario selecciona un trayecto y pulsa el botón Eliminar. Tras pedir confirmación, el usuario pulsa aceptar.
 - **Resultado esperado:** Debe desaparecer el trayecto seleccionado del listado de trayectos de la línea de detalle y factura actual. Además, debe decrementar el importe de la línea de detalle en función del importe del trayecto eliminado.
 - **Resultado obtenido:** El resultado esperado.

- **Editar Línea Detalle**
 - **Acción:** El usuario pulsa el botón Nuevo del listado de líneas de detalle de una factura determinada. Aparece una ventana con las propiedades de la línea de detalle y pulsa aceptar.
 - **Resultado esperado:** Debe aparecer la nueva línea de detalle en el listado de detalle de la factura determinada. Además, debe incrementar el importe de la factura en función del importe de la nueva línea de detalle.
 - **Resultado obtenido:** El resultado esperado.

- **Eliminar Línea Detalle**
 - **Acción:** El usuario selecciona una línea de detalle y pulsa el botón Eliminar. Tras pedir confirmación, el usuario pulsa aceptar.
 - **Resultado esperado:** Debe desaparecer la línea de detalle del listado de líneas de detalle de factura actual. Además, debe decrementar el importe de la factura en función del importe de la línea de detalle.
 - **Resultado obtenido:** El resultado esperado.

- **Eliminar**
 - **Acción:** El usuario selecciona una factura emitida del listado y pulsa el botón de eliminar. Aparece una ventana de confirmación y el usuario pulsa aceptar.
 - **Resultado esperado:** Debe eliminar la factura emitida seleccionada en las tablas TrayectosDetalleFactura, DetallesFactura y Facturas de la base de datos y se actualiza el listado de facturas emitidas, desapareciendo las facturas emitidas seleccionadas.
 - **Resultado obtenido:** Eliminada la factura emitida seleccionada que desaparece del listado de facturas emitidas.

- **Filtrar**
 - **Acción:** El usuario pulsa el botón filtrar que mostrará los campos disponibles para filtrar (Fecha factura, Vencimiento, Cliente y Forma de Pago), selecciona la forma de pago '30 Días'.
 - **Resultado esperado:** Deben aparecer en el listado únicamente las facturas emitidas cuya forma de pago es la seleccionada.
 - **Resultado obtenido:** Aparecen en el listado únicamente las facturas emitidas cuya forma de pago es '30 Días'.

- **Consultas**

No hay ninguna consulta dinámica asociada a las facturas emitidas.

8.1.4 Casos de Prueba del Caso de Uso Facturas Recibidas

- **Seleccionar Proveedor**

- **Acción:** El usuario pulsa el botón Nuevo en el listado de facturas recibidas y pulsa el link 'Proveedor'. Aparece un listado abreviado de todos los proveedores en los que se puede filtrar la búsqueda por TVA o por Empresa. Selecciona la búsqueda por Empresa y se introduce en patrón PEDRO*. Aparecen los proveedores que empiezan por PEDRO. El usuario selecciona el proveedor PEDRO MANUEL GONZALEZ BLANCO y pulsa Aceptar.
- **Resultado esperado:** Debe quedar seleccionado en las propiedades de la factura recibida, el proveedor en el combo de proveedores.
- **Resultado obtenido:** El resultado esperado.

- **Editar Certificado Proveedor**

- **Acción:** Una vez que el usuario ha seleccionado el proveedor, comprueba si posee certificado. En caso de que no tenga certificado, avisa al usuario de tal circunstancia. Pulsando doble click en el aviso permite crear el certificado del proveedor. Aparecerá una nueva ventana en la que el usuario deberá rellenar las propiedades del certificado y pulsar aceptar.
- **Resultado esperado:** Deberá insertar el nuevo certificado en la tabla ProveedorCertificado de la base de datos, además deberá desaparecer el aviso en la factura recibida.
- **Resultado obtenido:** Insertado 1 nuevo certificado para el proveedor seleccionado desapareciendo el aviso anteriormente mostrado.

- **Importar Trayectos del Proveedor**

- **Acción:** Una vez que el usuario ha seleccionado el proveedor, el sistema comprueba si tiene trayectos no facturados pendientes asociados al proveedor. El sistema muestra un listado con los trayectos no facturados pendientes del proveedor. El usuario selecciona los 2 primeros trayectos y pulsa aceptar.
- **Resultado esperado:** Deberá vincular (insertar) los trayectos seleccionados en la tabla FacturasRecibidasTrayectos de la base de datos, además estos trayectos deberán aparecer en el listado de trayectos de la factura recibida.
- **Resultado obtenido:** El resultado esperado.

- **Navegar al Trayecto en modo Consulta**
 - **Acción:** En el listado de trayectos que aparece en las propiedades de la factura, el usuario pulsa doble click en un trayecto.
 - **Resultado esperado:** Deberá aparecer la ventana de propiedades del trayecto en modo consulta, es decir, con todas sus propiedades en lectura.
 - **Resultado obtenido:** El resultado esperado.

- **Eliminar (desvincular) Trayectos**
 - **Acción:** Del listado de trayectos que aparece en la ventana de propiedades de la factura emitida, el usuario selecciona dos trayectos y pulsa el botón Eliminar.
 - **Resultado esperado:** Deberá desvincular (eliminar) los trayectos seleccionados en la tabla FacturasRecibidasTrayectos de la base de datos, además estos trayectos deberán desaparecer en el listado de trayectos de la factura recibida.
 - **Resultado obtenido:** El resultado esperado.

- **Editar**
 - **Acción:** El usuario pulsa el botón Nuevo, rellena la ficha de propiedades de la factura recibida y pulsa aceptar.
 - **Resultado esperado:** Debe insertar la nueva factura recibida en la tabla FacturasRecibidas y los trayectos vinculados en la tabla FacturasRecibidasTrayectos y debe actualizar el listado de facturas recibidas mostrando las propiedades de la nueva factura recibida.
 - **Resultado obtenido:** Insertado 1 nueva factura recibida con sus trayectos vinculados que aparece en el listado de facturas recibidas.

 - **Acción:** El usuario selecciona una factura recibida y pulsa el botón Propiedades, modifica sus propiedades en la ficha de propiedades de la factura recibida y pulsa aceptar.
 - **Resultado esperado:** Debe actualizar las propiedades de la factura recibida seleccionada en la tabla FacturasRecibidas y en el caso de que hayan variado, desvincular los trayectos que tuviera y vincular los nuevos trayectos en la tabla FacturasRecibidasTrayectos. Se debe actualizar el listado de facturas recibidas mostrando sus propiedades actualizadas.
 - **Resultado obtenido:** El resultado esperado.

- **Eliminar**
 - **Acción:** El usuario selecciona una factura recibida del listado y pulsa el botón de eliminar. Aparece una ventana de confirmación y el usuario pulsa aceptar.
 - **Resultado esperado:** Debe eliminar la factura recibida seleccionada en la tabla FacturasRecibidas y de la tabla FacturasRecibidasTrayectos de la base de datos y se actualiza el listado de facturas recibidas, desapareciendo la factura recibida seleccionada.
 - **Resultado obtenido:** Eliminada la factura recibida que desaparece del listado de facturas recibidas.

- **Buscar**
 - **Acción:** El usuario pulsa el botón buscar, selecciona el campo Proveedor e introduce el valor ROMAN*. A continuación pulsa el botón Buscar.
 - **Resultado esperado:** Debe posicionarse en la primera factura recibida cuyo nombre de proveedor empiece por ROMAN.
 - **Resultado obtenido:** El resultado esperado.

- **Filtrar**
 - **Acción:** El usuario pulsa el botón filtrar que mostrará los campos disponibles para filtrar (Fecha Factura, Vencimiento y Proveedor). En el campo proveedor selecciona el proveedor 'ROMAN DE DIEGO RUBIALES'.
 - **Resultado esperado:** En el listado de facturas emitidas únicamente deben aparecer las facturas cuyo proveedor sea 'ROMAN DE DIEGO RUBIALES'.
 - **Resultado obtenido:** En el listado de facturas aparecen 19 facturas todas ellas asociadas al proveedor 'ROMAN DE DIEGO RUBIALES'.

- **Consultas**

No hay ninguna consulta dinámica asociada a las facturas recibidas.

8.1.5 Casos de Prueba del Caso de Uso Pagarés a Proveedores

- **Imprimir**
 - **Acción:** El usuario selecciona el pagaré, pulsa el botón Imprimir y selecciona la opción 'A Impresora'.
 - **Resultado esperado:** Se deben imprimir por impresora el pagaré seleccionado.
 - **Resultado obtenido:** Se imprime el pagaré por la impresora.

- **Generar Documento Word**
 - **Acción:** El usuario selecciona un pagaré, pulsa el botón Imprimir y selecciona la opción 'A Word'.
 - **Resultado esperado:** Tras aceptar la confirmación de visualización de la factura, debe mostrar el pagaré en Word. El documento Word generado tendrá el nombre PagareProveedor_<aaaa>_<nnnn>.doc donde aaaa es el año y nnnn el nº de pagaré.
 - **Resultado obtenido:** Se genera y se visualiza correctamente el pagaré PagareProveedor_2010_0329.doc.

- **Seleccionar Proveedor**
 - **Acción:** El usuario pulsa el botón Nuevo en el listado de pagarés y pulsa el link 'Proveedor'. Aparece un listado abreviado de todos los proveedores en los que se puede filtrar la búsqueda por TVA o por Empresa. Selecciona la búsqueda por Empresa y se introduce en patrón PEDRO*. Aparecen los proveedores que empiezan por PEDRO. El usuario selecciona el proveedor PEDRO MANUEL GONZALEZ BLANCO y pulsa Aceptar.
 - **Resultado esperado:** Debe quedar seleccionado en las propiedades del pagaré, el proveedor en el combo de proveedores.
 - **Resultado obtenido:** El resultado esperado.

- **Editar Certificado Proveedor**
 - **Acción:** Una vez que el usuario ha seleccionado el proveedor, comprueba si posee certificado. En caso de que no tenga certificado, avisa al usuario de tal circunstancia. Pulsando doble click en el aviso permite crear el certificado del proveedor. Aparecerá una nueva ventana en la que el usuario deberá rellenar las propiedades del certificado y pulsar aceptar.
 - **Resultado esperado:** Deberá insertar el nuevo certificado en la tabla ProveedorCertificado de la base de datos, además deberá desaparecer el aviso en el pagaré.
 - **Resultado obtenido:** Insertado 1 nuevo certificado para el proveedor seleccionado desapareciendo el aviso anteriormente mostrado.

- **Importar Facturas Recibidas del Proveedor**
 - **Acción:** Una vez que el usuario ha seleccionado el proveedor, el sistema comprueba si tiene facturas recibidas sin pagaré asociadas al proveedor. El sistema muestra un listado con las facturas recibidas sin pagaré asociadas al proveedor. El usuario selecciona las 2 primeras facturas y pulsa aceptar.
 - **Resultado esperado:** Deberá vincular (actualizar el campo pagaréID y establecer como pagadas) las facturas recibidas seleccionadas en la tabla FacturasRecibidas de la base de datos, además estas facturas deberán aparecer en el listado de facturas recibidas del pagaré a proveedor.
 - **Resultado obtenido:** El resultado esperado.

- **Navegar a la Factura Recibida en modo Consulta**
 - **Acción:** En el listado de facturas recibidas que aparece en las propiedades del pagaré, el usuario pulsa doble click en una factura.
 - **Resultado esperado:** Deberá aparecer la ventana de propiedades de la factura recibida en modo consulta, es decir, con todas sus propiedades en lectura.
 - **Resultado obtenido:** El resultado esperado.

- **Eliminar (desvincular) Facturas recibidas**
 - **Acción:** Del listado de facturas que aparece en la ventana de propiedades del pagaré, el usuario selecciona dos facturas y pulsa el botón Eliminar.
 - **Resultado esperado:** Deberá desvincular (actualizar el campo pagaréID a null y establecer como no pagadas) las facturas seleccionadas en la tabla FacturasRecibidas de la base de datos, además estas facturas deberán desaparecer en el listado de facturas del pagaré.
 - **Resultado obtenido:** El resultado esperado.

- **Editar**

- **Acción:** El usuario pulsa el botón Nuevo, rellena la ficha de propiedades del pagaré a proveedor y pulsa aceptar.
- **Resultado esperado:** Debe insertar el nuevo pagaré en la tabla PagaresProveedores y vincular (actualizar el campo pagareId y establecer como pagada) las facturas seleccionadas en la tabla FacturasRecibidas. Además, debe actualizar el listado de pagarés a proveedor mostrando las propiedades del nuevo pagaré.
- **Resultado obtenido:** Insertado 1 nuevo pagaré con sus facturas recibidas vinculadas que aparece en el listado de pagarés a proveedores.

- **Acción:** El usuario selecciona un pagaré y pulsa el botón Propiedades, modifica sus propiedades en la ficha de propiedades del pagaré y pulsa aceptar.
- **Resultado esperado:** Debe actualizar las propiedades del pagaré seleccionado en la tabla PagaresProveedores y en el caso de que hayan variado, desvincular las facturas que tuviera y vincular las nuevas facturas en la tabla FacturasRecibidas. Se debe actualizar el listado de pagarés a proveedores mostrando sus propiedades actualizadas.
- **Resultado obtenido:** El resultado esperado.

- **Eliminar**

- **Acción:** El usuario selecciona un pagaré del listado y pulsa el botón de eliminar. Aparece una ventana de confirmación y el usuario pulsa aceptar.
- **Resultado esperado:** Debe eliminar el pagaré seleccionado en la tabla PagaresProveedores y desvincular (establecer las propiedades PagadoSN = No y PagareID=NULL) las facturas recibidas que tuviera asociadas en la tabla FacturasRecibidas de la base de datos. Además debe actualizar el listado de pagarés a proveedor, desapareciendo el pagaré seleccionado.
- **Resultado obtenido:** Eliminado el pagaré que desaparece del listado de pagarés a proveedores.

- **Buscar**

- **Acción:** El usuario pulsa el botón buscar, selecciona el campo Proveedor e introduce el valor ROMAN*. A continuación pulsa el botón Buscar.
- **Resultado esperado:** Debe posicionarse en el primer pagaré cuyo nombre de proveedor empiece por ROMAN.
- **Resultado obtenido:** El resultado esperado.

- **Filtrar**
 - **Acción:** El usuario pulsa el botón filtrar que mostrará los campos disponibles para filtrar (Fecha Pagaré, Vencimiento, Proveedor y Tipo). En el campo Tipo selecciona el tipo Cheque.
 - **Resultado esperado:** En el listado de pagarés únicamente deben aparecer los pagarés cuyo tipo sea Cheque.
 - **Resultado obtenido:** En el listado de pagarés a proveedores aparecen 39 pagarés, todas ellos del tipo Cheque.

- **Consultas**

No hay ninguna consulta dinámica asociada a las facturas emitidas.

8.2 Conclusiones

La realización de las pruebas descritas en los puntos anteriores no ha dado lugar a ningún error importante ya que éstos se han ido subsanando a medida que se ha ido desarrollando el código y probado con las pruebas pertinentes.

Además, durante el periodo de implantación y explotación en la empresa Fercaez, se han ido solucionando las incidencias surgidas.

Una vez corregidos los fallos detectados se ha verificado que el sistema funciona correctamente y actúa de la manera esperada, por lo que se puede concluir que la aplicación es altamente fiable en las condiciones de sistemas descritas a lo largo de la memoria.

9. CONCLUSIONES GENERALES

Técnicamente, el proyecto desarrollado cumple de forma satisfactoria con los objetivos y expectativas planteados en un inicio. Se ha conseguido desarrollar una aplicación Cliente-Servidor orientada a una gestión muy completa de la actividad de una pequeña empresa de transporte. La aplicación muestra una interfaz compacta, uniforme, lógica y sencilla de utilizar que resulta de fácil comprensión al usuario.

Personalmente, el proyecto ha supuesto un reto personal y académico. Este proyecto se ha basado fundamentalmente en el aprendizaje de una empresa de transporte de mercancías, además de aportar la experiencia suficiente para gestionar proyectos de mediana envergadura.

Por último, deseo resaltar la satisfacción de la empresa Fercaez con el producto desarrollado, que supone una garantía para explorar vías comerciales con el fin de exponer el producto en el mercado.

10. LÍNEAS FUTURAS

La aplicación GeTrans está diseñada para permitir agregar nuevas funcionalidades en un futuro de manera sencilla y práctica, por su estructura en módulos y su diseño técnico. Estas mejoras, precisan para su detección de una investigación más exhaustiva de la actividad del transporte, para poder proporcionar soluciones más satisfactorias.

El aspecto fundamental de mejora es migrar la aplicación a un lenguaje más moderno y que se integrará mejor en los sistemas operativos Windows del futuro (Windows 7 y 8). En nuestro caso, la migración natural, más sencilla y eficiente es la migración al entorno .NET, en concreto Visual Basic .NET.

Otro aspecto muy importante a tener en cuenta, es la migración de la base de datos de Microsoft Access a una base de datos de características más orientadas al modelo Cliente-Servidor. En nuestro caso, la migración natural, más sencilla y eficiente es la migración a Microsoft SQL Server, en concreto a partir de la versión 2008 (2008, 2012). La migración en este caso es sencilla y supone muy pocos cambios ya que se conserva el modelo de datos y únicamente sería necesario modificar la conexión de la aplicación con la base de datos.

11. BIBLIOGRAFÍA

- 1.- Programación avanzada con Microsoft Visual Basic 6.0 de Francesco Balena.
- 2.- ¿Cómo se hace con Visual Basic 6 Bases de datos? De Eric Winemiller, Jason Roff, Bill Heyman y Ryan Groom.
- 3.- El Guille, la Web del Visual Basic, C#, .NET y más...
<http://www.elguille.info/default.aspx>
- 4.- La web del programador – Comunidad de Programadores
<http://www.lawebdelprogramador.com>
- 5.- Microsoft Office – Office.com
<http://office.microsoft.com/es-es/?CTT=97>
- 6.- MSDN de Visual Studio 6.0