



UNIVERSIDAD
POLITECNICA
DE VALENCIA

ESCUELA TÉCNICA SUPERIOR
DE INGENIERÍA INFORMÁTICA

ESCUELA TÉCNICA SUPERIOR DE INGENIERÍA INFORMÁTICA

INGENIERIA TECNICA INFORMATICA DE SISTEMAS

**APLICACIÓN WEB PARA LA GESTIÓN DE VIAJES
DE UNA EMPRESA**

Realizado por

José Manuel Ferrer Pitarch

Dirigido por

César Ferri Ramírez

Empresa

Norauto

Tutor del proyecto

José Emilio Gil

Escuela Técnica Superior de Ingeniería Informática



Índice General

Capítulo 1	4
Introducción.....	4
La compañía. Norauto España.....	4
Objetivos generales del proyecto	5
Alcance del proyecto.....	6
Gestión actual	7
Descripciones y rasgos generales del proyecto	10
Capítulo 2	13
Análisis.....	13
2.1 Metodología del desarrollo	13
2.2 Especificación de los requisitos del sistema	13
2.3 Análisis del sistema	20
2.4 Desarrollo del calendario	40
2.5 Diagrama de Gantt.....	43
Capítulo 3	44
Diseño	44
3.1 Diseño del sistema.....	44
Diagramas WAE del sistema Web.....	49
3.2 Diseño de la base de datos	55
Capítulo 4	65
Utilización del sistema.....	65
4.1 Introducción al manual de usuario.....	65
4.2 Manual de usuario de central	67
4.3 Detalle usuario centros	76
4.4 Manual usuario viajes. Gestión de los pedidos	77
4.5 Revisión de propuestas de centros y creación de pedidos	78
4.6 Recepción de pedidos	80
4.7 Manual usuario contabilidad	83
Capítulo 5	86
Conclusiones.....	86
5.1 Elección del proyecto.....	86
5.2 Futuro y valoración del producto	86
5.3 Posibles mejoras y modificaciones.....	87
Capítulo 6	88
Referencias y bibliografía.....	88
6.1 Referencias en la red.....	88

6.2 Bibliografía	88
Apéndice A	89
Estructura de directorios del sistema.....	89
A.1 Descripción del directorio	89
Apéndice B	94
Librerías. Funciones a destacar.....	94
Apéndice C	102
Consultas de la base de datos.....	102
Listado de consultas	102

Capítulo 1

Introducción

Este capítulo introductorio sirve para aclarar de forma concisa los aspectos generales del proyecto, tales como, una descripción de la compañía cliente, los objetivos principales, la organización de la presente memoria, una descripción de los antecedentes al sistema con un escenario representativo de los mismos y los beneficios que aportará la puesta en producción de la aplicación.

1.1 La compañía. Norauto España.

El presente proyecto surge de la necesidad de una aplicación para la compañía cliente para la gestión centralizada de los viajes que los colaboradores de la empresa realizan habitualmente.

La compañía es Norauto España. Este proyecto será integrado a nivel global en la compañía española. Para entender un poco mejor el entorno de la empresa, vamos a describir brevemente la situación histórica de la empresa a nivel internacional y la importancia de la compañía española en el entorno global de Norauto Internacional.

El concepto de Norauto nace de la mano de Eric Derville , como un servicio de mantenimiento y mejora de los vehículos en el ámbito de los amplios parkings de los hipermercados. Así pues, nace en Lille en el año 1970 el primer Norauto de Francia. Durante la siguiente década la empresa se expande hasta un total de 10 auto centros. El concepto se asienta y se estructuran los servicios centrales. El ritmo de expansión se incrementa progresivamente y la empresa siente la necesidad de abrirse a nivel internacional. Así pues, Norauto decide expandirse por Europa, principalmente en España (1986), después lo haría en Italia, Bélgica, Portugal, y Polonia.

Actualmente Norauto cuenta con más de 6000 colaboradores y más de 350 centros repartidos por todo el mundo. Cada país de forma autónoma, evoluciona respetando las políticas comerciales, sociales y administrativas específicas de Norauto.

Hablando ya de España, sección de la compañía que nos ocupa el proyecto, España es el país elegido por Francia como principal apuesta de expansión, tanto es así, que actualmente después de Francia, se sitúa como el país más fuerte en desarrollo entre todos

los países que engloban la Norauto. Norauto España cuenta hoy en día, con más de 1000 colaboradores y un total de 56 centros repartidos por toda la geografía española. Al igual que Francia en pocos años Norauto España evolucionó, lanzó sus campañas publicitarias, elaboró su proyecto de empresa, desarrolló su comunicación interna y estableció sus servicios centrales. Los servicios centrales de Norauto son el eje administrativo que maneja y organiza a todos los niveles los auto centros españoles. Se dividen en varios departamentos: Marketing, contabilidad, expansión, obras y mantenimiento, recursos humanos, formación interna, Informática, logística y compras. Cada auto centro se compone de un taller de mecánica y mantenimiento de vehículos, más una tienda en la que puedes encontrar todo tipo de artículos relacionados con los vehículos, con una amplia gama y una posición bastante buena en el mercado.

Una vez situada la empresa a nivel histórico y descriptivo, cabe destacar, donde surge la idea del proyecto. La mayoría de aplicaciones se heredan de Francia, otras de más envergadura e importancia las compran a grandes empresas que se dedican especialmente a esto. En el próximo punto hablaremos de los objetivos generales del proyecto.

1.2 Objetivos generales del proyecto

Los objetivos del presente proyecto son, por una lado, poner en práctica los conocimientos adquiridos a lo largo de las asignaturas de la titulación de Ingeniería Técnica en Informática de Sistemas y por otro lado, objetivo de vital importancia, contribuir con el departamento de informática a mejorar la empresa mediante el desarrollo de una herramienta necesaria para la compañía.

El resultado de este proyecto, es el desarrollo de una aplicación Web para realizar la gestión automatizada de los viajes de los colaboradores de la compañía.

Pasamos a enumerar de forma esquematizada los principales objetivos que engloba este proyecto:

- **Gestión de propuestas de pedido:** Todas las herramientas necesarias para la creación de propuestas de cualquier usuario de la compañía que necesite un desplazamiento en la empresa.
- **Gestión de los pedidos:** Herramientas necesarias para la gestión de las propuestas que los usuarios realizan a la compañía y su transformación a pedido hacia a un proveedor registrado en el sistema, generando así un pedido automático con toda la información de la propuesta que se acepte desde recepción de servicios centrales.

- **Gestión de usuarios:** Hemos descrito la amplitud de la empresa en cuanto a colaboradores. Obviamente todos los colaboradores no tienen las mismas necesidades ni tienen que tener acceso a la aplicación. La visibilidad de la aplicación dentro de la empresa debe ser controlada. Por otra parte, todos los usuarios no realizan el mismo uso, según su departamento o situación, el uso de la herramienta irá destinado a un fin u otro, todo esto debe ser gestionado por un administrador.
- **Gestión de facturas:** este apartado integra las herramientas necesarias para centralizar y almacenar las facturas en el sistema, de forma que el proceso contable se agilice al máximo. La herramienta debe ofrecer a la recepción la opción de crear facturas que contabilidad concilie a continuación y todo esto quede registrado.
- **Informes contables:** Generación de informes contables con todas las facturas de todos los ámbitos, para poder ejercer un control económico. Estos informes se deben poder imprimir además de almacenar en ficheros de hojas de datos, para su posterior tratamiento.

1.3 Alcance del proyecto

Se debe centralizar informáticamente toda la gestión de pedidos de viajes que efectúan los colaboradores. La misión del proyecto es agilizar el trabajo de recepción y contabilidad en los servicios centrales con todos los beneficios que eso conlleva.

1.3.1 Identificación de la herramienta mediante un nombre

El producto concebido se llama "Gevi" que es algo así como una abreviatura de las palabras "Gestión de viajes". Se pondrá en producción con ese nombre y toda la empresa lo conocerá e identificará de esa manera.

1.3.2 Aplicaciones del sistema: beneficios y metas

Con este producto podemos obtener diferentes ventajas ya descritas anteriormente.

Entre ellas cabe destacar el registro informático de los datos de la empresa. Los datos es el material más importante que tiene una empresa en la actualidad y su pérdida conlleva a uno de los daños más grandes que puede sufrir una empresa.

También podemos utilizar los datos y los informes almacenados para la toma de diferentes decisiones tan relevantes para el futuro de la empresa.

El incipiente proyecto, surge de la necesidad de una herramienta para el ámbito de recepción y contabilidad de los servicios centrales. La necesidad del desplazamiento de los empleados entre las sucursales de la empresa, es algo habitual y esto genera mucho trabajo para los recepcionistas de servicios centrales, encargados de gestionar todos estos pedidos. Así mismo estas personas tienen que combinar y coordinar sus movimientos con los contables, para justificar e imputar los gastos que generan estos pedidos. Ahora mismo, resulta un caos gestionar todo esto y requiere de un seguimiento y control máximo sobre el correo electrónico, algo insostenible teniendo en cuenta el crecimiento de la demanda de pedidos y de la propia empresa. Es por ello, que esta herramienta supone un beneficio directo, hablando de todas estas tareas de coordinación.

Por otro lado con la utilización del producto, la empresa dará una sensación de profesionalidad y confianza a sus usuarios. Los usuarios deben de tener a su alcance, las herramientas necesarias para elaborar su trabajo de forma eficaz y sostenible.

1.4 Gestión actual

En este apartado se va a describir de forma concisa, cuál es la situación actual en cuanto a la gestión de los pedidos de viajes, llevado a cabo prácticamente en su totalidad, por la recepción de servicios centrales. A su vez, para su mayor claridad, ejemplificaremos la descripción con un escenario particular de un pedido.

1.4.1 Descripción de la metodología y limitaciones del proceso.

Vamos a describir de forma esquemática cual es el flujo de trabajo que genera el proceso de un pedido por parte de un empleado. Además, acompañando a los ejes del flujo del proceso, añadiremos una descripción concisa de las limitaciones que supone el método para justificar la necesidad de la herramienta.

- Un empleado necesita desplazarse de un centro cualquiera de España a otro centro por los motivos que sea. Como todo en Norauto, la coordinación y gestión de la empresa se gestiona en las oficinas de servicios centrales. Es por ello que la demanda del empleado la debe traspasar a servicios centrales.
- El empleado vía correo electrónico envía un e-mail al departamento de recepción de servicios centrales. Este correo integra todos los requisitos que necesita ese desplazamiento: Medio de transporte, ciudades implicadas en los desplazamientos, alojamiento, vehículo de alquiler para desplazarse al lugar de destino, horarios etc....

- El departamento recepcionista formado por dos personas, se encarga entre otras tareas, de acoger telefónicamente todas las llamadas que reciben las oficinas centrales, así como recepcionar a la gente que acude allí, proveedores etc... El correo electrónico, es un elemento útil pero limitado. Gestionar y organizar los movimientos de una empresa con tantos colaboradores mediante correo electrónico, conduce a una situación insostenible y de desorden. Además, mantener un histórico organizado de todos los movimientos de los pedidos mediante el uso de la bandeja de correo, también resulta algo insostenible. El recepcionista, recibe el correo electrónico del pedido del empleado, entre la multitud de correos que integra su bandeja de correo diariamente (otra dificultad que manifiesta la limitación del proceso).
- Una vez leído el correo y tratados los elementos, el recepcionista genera otro correo electrónico destinado al proveedor o gerencia encargada de tramitar los elementos del viaje.
- El proveedor, automáticamente envía un correo de confirmación con los elementos tratados. El recepcionista reenvía este correo al empleado que solicitó el pedido, con copia al contable correspondiente, de modo informativo para que el colaborador sepa que se está tramitando su solicitud.
- Una vez formalizado el pedido con el proveedor, a posteriori, el proveedor enviará los bonos y la factura correspondiente en formato electrónico a Recepción. Recepción deberá de “hurgar” en su correo, para encontrar el pedido correspondiente y la información del mismo, para cotejar la factura y pasarla al departamento contable, para imputar los gastos necesarios. De aquí surge otra limitación clara de la metodología utilizada: utilizar la bandeja de correo como medio de mantener un histórico y efectuar una búsqueda de los pedidos supone un uso ineficaz del tiempo, así como un medio de almacenamiento inviable...
- Finalmente, la recepcionista pasa la factura cotejada al departamento contable, el cuál, deberá revisar su correo para encontrar el pedido y así saber que ese movimiento está tratado correctamente. Además, no tiene ningún medio de almacenamiento más o menos eficaz para hacer búsquedas. Tampoco tiene ninguna herramienta para generar un formato contable que integre todos los registros de las facturas entendibles por el sistema informático contable del departamento.

Queda claro tras todos estos puntos, que tanto el departamento de recepción como el contable, tienen una tarea complicada mediante correo electrónico. Además no se mantiene ningún histórico de los movimientos contables, ni de los desplazamientos que realizan los colaboradores.

1.4.2 Escenario particular del proceso

A continuación, como dijimos en un apartado anterior, vamos a escenificar el proceso actual con un escenario particular. Un ejemplo de pedido será descrito a continuación.

1. El empleado director de Murcia va a hacer un viaje con motivo de una reunión interna. Genera un correo a recepción con lo que necesita:

Hola Yasna:

Por favor, necesito que reserves una noche de hotel del día 8 al 9, para Pedro Manuel Martínez Lapaz y para mí. (Carlos Rodríguez Gómez)

El hotel me da igual, en Valencia o alrededores, es para evitar darnos la paliza del madrugón y viaje

El motivo es la reunión de sector. Javier Viñals esta al tanto, y me ha dado el OK

Gracias

2. Recepción trata el correo, trata las líneas de petición y reenvía el correo con la información real que formará parte del pedido al proveedor, con copia al empleado.

Asunto: PEDIDO NORAUTO: 4018

PEDIDO NORAUTO Nº 4018 (CARLSON)

DESCRIPCIÓN PEDIDO: R SECTOR MURCIA

PERSONA DE CONTACTO: YASNA PEREZ (963160300)

PRIORIDAD EN EL PRECIO

NOTA: URGENTE

DETALLE DEL PEDIDO

HOTEL: valencia

FECHA ENTRADA: 08/12/2010 FECHA SALIDA: 09/12/2010

CARLOS RODRIGUEZ GOMEZ (07231244K)

H NOVOTEL

HOTEL: VALENCIA

FECHA ENTRADA: 08/12/2010 FECHA SALIDA: 09/12/2010

PEDRO MANUEL MARTINEZ LAPAZ (34802415B)

H NOVOTEL

3. El proveedor confirma el pedido. El recepcionista reenvía el correo a contabilidad con copia al empleado.

Asunto: Bono: 1650725 Cliente: 0618461: MARTINEZ LAPAZ/PEDRO MANUELSR: 007/007

Adjuntamos bono de referencia.

Este es un envío automático, por favor no responda a este e mail, en caso de necesidad contacte con la oficina de Carlson Wagonlit indicada en el bono.

4. Recepción recibe la factura del pedido, coteja todas las líneas de la factura con la información del pedido formal y pasa la factura a contabilidad.

Para: Lista RECEPCION

Asunto: Albarán: 4751043 Cliente: 0618461: MARTINEZ LAPAZ/PEDRO MANUELSR: 007/007

Adjuntamos albarán de referencia.

Este es un envío automático, por favor no responda a este e mail, en caso de necesidad contacte con la oficina de Carlson Wagonlit indicada en el albarán.

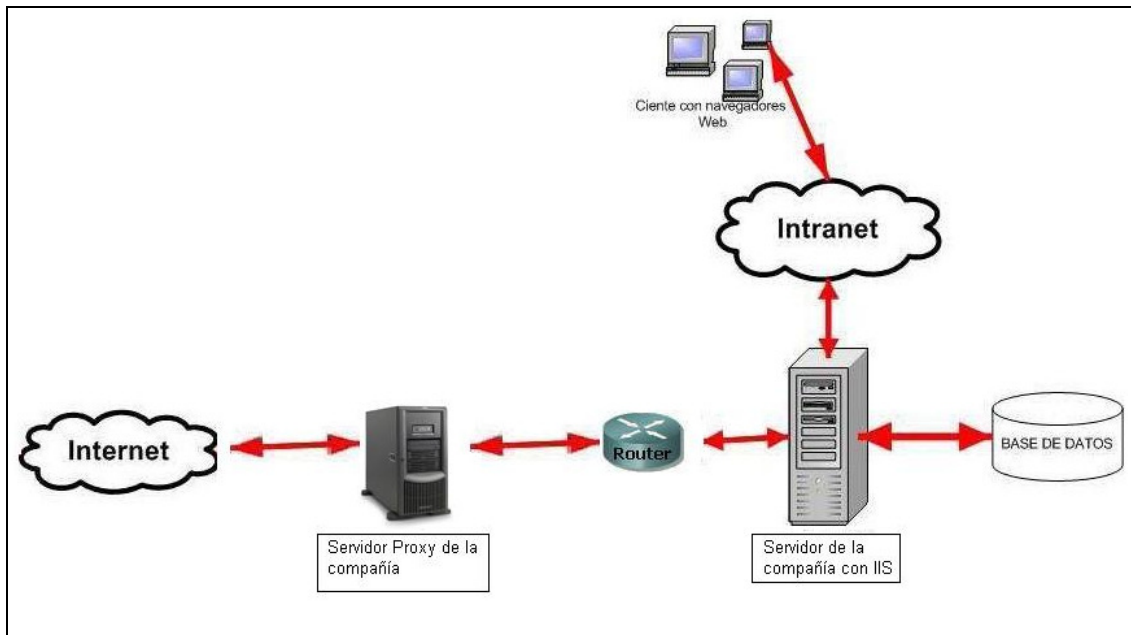
1.5 Descripciones y rasgos generales del proyecto

1.5.1 Dependencias del producto

El proyecto es independiente, no forma parte de un sistema mayor. Este producto no es continuación de ningún proyecto, puede ser continuado o actualizado en un tiempo futuro, ya sea por el creador del propio o por personas ajenas a este.

1.5.2 Perspectiva del sistema

Como hemos indicado, el producto no forma parte de un sistema mayor, pero si que va a estar integrado dentro de una intranet global de la compañía, junto con otras aplicaciones de la misma. Los usuarios que estén dentro de la red Norauto, van a poder acceder al sistema sin problemas o mediante el uso de una cuenta VPN. La siguiente figura clarifica la situación del sistema.



1.5.3 Operaciones

Podemos distinguir varios tipos de usuario dentro de la aplicación:

- **Pasajero/conductor:** Empleado que no tiene acceso a la aplicación pero puede ser incluido en un viaje.
- **Pasajero especial:** Usuario ficticio para ser incluido en pedidos.
- **Usuario Estándar:** Con acceso a la aplicación y que puede hacer propuestas de viaje (Se envían a recepción). Tendrá activos los menús "Mis viajes" y "mis datos".
- **Usuario Avanzado:** Con acceso a la aplicación y que puede hacer pedidos (Se envían directamente a la agencia). Tendrá activos los menús "Mis viajes" y "mis datos".
- **Usuario Viajes:** Con acceso a la aplicación y que tiene permisos en las secciones de creación y recepción de pedidos. Tendrá activos los menús "Mis viajes", "mis datos" y "Dpto Viajes"
- **Usuario Contabilidad:** Con acceso a la aplicación y que tiene permisos en las secciones de contabilidad. Tendrá activos los menús "Mis viajes", "mis datos" y "Dpto Conta"

- **Administrador:** Con acceso a todas las secciones de la aplicación. Tendrá activos todos los menús.

1.5.4 Características de los usuarios

La aplicación *Gevi* requiere de unos conocimientos básicos de informática para su manejo.

Recomendamos a los usuarios que no dispongan de esos conocimientos de:

- Asistir a cursos de iniciación a la informática
- Familiarizarse con aplicaciones semejantes
- Familiarizarse con los términos de la aplicación

Para el usuario Administrador. se requiere un usuario con conocimientos un poco mas avanzados de informática y si es posible. sustentar un cargo de responsable o directivo dentro de la empresa.

Capítulo 2

Análisis

2.1 Metodología del desarrollo

La metodología empleada para el desarrollo de este proyecto, se basa en la utilización del Proceso Unificado Rational (*Rational Unified Process (RUP)*) este sistema junto con el Lenguaje Unificado Modelado (UML) constituyen la metodología estándar más utilizada para el análisis, implementación y documentación de sistemas informáticos.

Esta herramienta, no es una aplicación informática orientado a objetos, se trata de un sistema Web basado en páginas con secuencias de comandos, ejecutados en el servidor (código asp) y páginas cliente con lenguaje HTML. Así pues, adaptaremos el proceso unificado en la medida de lo posible como modelo de metodología y emplearemos el sistema WAE como principio de modelado UML, una extensión que se adapta a nuestras necesidades y que detallaremos en la sección correspondiente.

2.2 Especificación de los requisitos del sistema

2.2.1 Requisitos de la interfaz

2.2.1.1 Requisitos de la interfaz de usuario

A continuación pasaremos a enumerar los diferentes requisitos de la interfaz gráfica de usuario:

- *Interfaz atractiva para el usuario:* No es el requisito más importante del sistema, pero si es necesario un mínimo de atractivo. Con una interfaz atractiva conseguiremos captar la aceptación del usuario por la nueva aplicación y evitaremos el rechazo. Para conseguir este requisito hemos empleado el uso de hojas de estilo CSS, que ajustan el estilo de las páginas Web de todas las páginas, sin tener que ir modificando manualmente el código.
- *Menú principal:* A todas las partes del sistema y a todas las secciones de la herramienta, se accede desde un menú principal situado en la parte superior

de las páginas y que está visible en todo momento. Este menú además contiene y diferencia claramente los 4 bloques o subsistemas en que se divide la herramienta.

- *Interfaz sencilla:* Este quizás, sea el requisito más importante y en el que más han insistido los promotores del proyecto. Cabe destacar que la herramienta va a ser utilizada por muchos colaboradores de la compañía, de todos los ámbitos y perfiles, así pues es indispensable presentar una interfaz sencilla que se adapte a las necesidades y capacitaciones de la amplia gama de usuarios de la empresa.
- *Interfaz cómoda:* Este requisito junto al citado justo antes, constituyen los pilares básicos de los requisitos que estamos tratando en esta sección. Como dijimos anteriormente, una herramienta de alto impacto debe ser cómoda e intuitiva. Para ello el sistema estará dotado de un asistente detallado y de fácil manejo para la creación de propuestas de viaje. Además, no será necesario rebuscar en menús y opciones para manejar la herramienta. La utilización de colores e iconos facilita y responde al requisito de la intuición, en el momento de manejar la aplicación.
- *Ventanas de diálogo:* Cuando el usuario se disponga a efectuar una acción crítica o decisiva, el sistema lanzará una ventana de diálogo que advertirá de la importancia de la acción, además de solicitarle la confirmación o rechazo de la misma. Para ello, hemos hecho uso de alertas y funciones de confirmación basadas en tecnología JavaScript, cumpliendo pues el requisito de informar al usuario, del impacto de las acciones más importantes: creación, modificación y visualización de pedidos.

2.2.1.2 Requisitos de informes generados

Los informes, son otra forma que tiene la herramienta de interactuar con el usuario. La utilización de reportes, son fundamentales en esta aplicación, ya que a través de ellos, los usuarios autorizados y no autorizados, obtienen la información que necesitan en todo momento tanto en versión digital como impresa. A continuación detallamos las propiedades necesarias que deben de cumplir estos reportes:

- *Cantidad de información necesaria:* Los reportes generados por la herramienta deben contener la máxima información posible y en el menor espacio posible, de forma organizada y estructurada. Para ello, la sección contable la herramienta, es capaz de

generar reportes de facturas en estructura de tabla, en las cuales vemos los datos necesarios para el departamento contable. Además, todos los pedidos y propuestas pendientes, se visualizan en la herramienta de forma estructurada y clara.

- *Claridad y veracidad:* Los informes deben ser correctos y mostrar la información precisa siempre basados, en los movimientos de la aplicación en conjunto con los almacenes de datos del sistema. Deben de ser legibles y claros para facilitar y responder a las necesidades de agilización.
- *Formato:* Como se mencionó antes, el formato de los informes debe tener una estructura familiar y atractiva para el usuario específico. En este caso hemos adoptado la estructura de tabla, que es la que más se adecua a las necesidades del usuario.
- *Reportes de hojas de cálculo:* El sistema generará unos reportes en formato .xls adaptables para su comunicación las herramientas contables de la empresa. Sobre esta característica hablaremos en los requisitos funcionales.

2.2.2 Requisitos funcionales

A continuación numeramos los requisitos funcionales de la herramienta de forma categórica:

- **Requisitos funcionales de identificación**
 - La aplicación debe ser capaz de identificar a los usuarios (distinguiendo por supuesto a los tipos de usuario) a través del login que utilizan en el sistema. Para la presentación de la aplicación así como para las previas pruebas pertinentes y antes de la puesta en producción, el sistema deberá presentar un modelo de autenticación de usuario, mediante un nombre de usuario y un password. Según el login empleado, el usuario accederá con un rol u otro, al sistema.
- **Requisitos funcionales de la gestión de usuarios**
 - El sistema deberá de informar a través de la pestaña administración, una lista de los usuarios con acceso al sistema, con detalles específicos de los mismos.
 - Se podrá crear un nuevo usuario a través de la pestaña administración. Este usuario, se creará con un código de usuario que lo identificará unívocamente en el sistema. Además el usuario deberá de tener un superior jerárquico (el

correspondiente en la empresa) y una correspondiente cuenta contable asociada.

- Se podrán editar/modificar y eliminar los datos de los usuarios registrados
- Se podrá realizar la búsqueda de un usuario que cumpla con determinados criterios de selección.
- El usuario facilitará el resto de informaciones, a través de su propia ficha después de haberle dado de alta en el sistema.

▪ **Requisitos funcionales de la gestión de proveedores**

- El sistema estará definido con varios proveedores entre ellos uno será el utilizado para envío de pedidos. Esto dependerá de las negociaciones de la empresa para la elección de la opción más rentable y viable.
- Los usuarios con perfil de gestión de pedidos (usuarios de central, recepcionista contable y administrador) podrán elegir al proveedor destinatario del pedido.
- Se podrá crear un nuevo proveedor si la empresa lo estima conveniente, mediante el registro de sus datos en la base de datos del sistema.

▪ **Requisitos funcionales de la creación y gestión de propuestas**

- Todos los usuarios con acceso, podrán crear propuestas y modificarlas. Podrán, una vez creada su propuesta de viaje, añadir más líneas de detalle a la propuesta. Podrán formalizar la propuesta, borrar la propuesta, además de borrar líneas de propuesta.
- Los usuarios con acceso limitado, esto es, los usuarios de centros, podrán hacer todo lo arriba mencionado, pero no crear pedidos. Estos pedidos los creará el departamento de viajes, una vez tramitada su propuesta, no podrán modificarla. El resto de usuarios podrán modificar sus propuestas y crear pedidos.
- Todos los usuarios (exceptuando los miembros del departamento de viajes y los administradores) cuando accedan al sistema, podrán visualizar una lista de sus viajes, pendientes de envío o tramitados. Tendrán una vista de todos sus

viajes (tramitados y/o no) y podrán acceder al detalle de ellos e incluso, modificarlos si tienen permisos para ello.

- Los usuarios recepcionistas, cuando accedan al sistema, visualizarán las propuestas enviadas por los usuarios de centros así como las suyas propias.
- El administrador, verá lo mismo que el recepcionista (en la vista principal de viajes), pero su misión no será la de tramitar pedidos, salvo los suyos propios.
- Únicamente los usuarios de centros (acceso limitado) tramitarán sus propuestas de pedido a través del departamento de viajes, el resto de usuarios tramitarán sus propias propuestas.

▪ **Requisitos funcionales de la creación y gestión de pedidos**

- Los pedidos de viajes, serán generados a partir de una propuesta de viaje. Un pedido puede incluir parte de la propuesta inicial o toda. Esto es, una propuesta puede derivar en varios pedidos para el proveedor seleccionado.
- Los usuarios registrados en el sistema (exceptuando los limitados) podrán crear, modificar y editar pedidos. Los pedidos de los usuarios limitados como citábamos antes, serán tramitados por los recepcionistas.
- Los usuarios recepcionistas (miembros del departamento de viajes) tendrán una vista de Gestión de pedidos en la cual, visualizarán con todo detalle una estructura de todos los pedidos abiertos en el sistema y pendientes de visado. Podrán acceder a cualquiera de ellos y modificarlos si cabe. Todo exceptuando las cuentas contables si ya han sido modificadas por algún usuario contable.

▪ **Requisitos funcionales de las propuestas enviadas al Departamento de viajes**

- Como hemos citado en un requisito anterior, los usuarios del departamento de viajes, recibirán un correo electrónico cuando un usuario de centros genere una propuesta y proceda al envío de la misma para su trámite. Estas propuestas, serán visibles para los usuarios recepcionistas en su vista de viajes del sistema.

- **Requisitos funcionales de la creación y gestión de facturas**

- Así mismo los usuarios del departamento de viajes, serán los responsables de generar los documentos de los pedidos o facturas correspondientes de cada pedido. Estas facturas sólo serán visibles para el departamento contable.
- Los usuarios contables registrados, podrán conciliar las facturas enviadas por el departamento de viajes, así como visualizar una lista de todas las facturas visadas o no para su trato. También tendrán acceso a una vista de los pedidos pendientes de factura, en tanto que podrán generar las respectivas facturas si procede.

- **Requisitos funcionales de la generación de informes contables**

- Los usuarios contables a través de la sección contable de la aplicación podrán generar reportes en formato de hoja de cálculo, que a posteriori utilizarán para integrar las facturas en el sistema contable de la compañía.
- Estos reportes se almacenarán con el nombre de usuario que lo genera con extensión .csv y serán accesibles dentro del propio servidor, para su posterior uso.

- **Requisitos funcionales del envío de pedidos**

- El sistema enviará un correo automático en el momento de la creación del pedido, este correo tendrá unos campos estipulados, y contendrá las líneas de detalle del pedido generado por el usuario (bien sea en el departamento de viajes o un usuario con permisos) todo fruto de la previa propuesta generada.
- El correo electrónico generado automáticamente, se enviará al proveedor por defecto con copia al superior jerárquico del usuario que solicita el pedido.

- **Requisitos del registro de movimientos del sistema**

Los requisitos más importantes que la empresa estima a la hora de mantener un histórico son:

- Dejar almacenado en la base de datos todas las propuestas creadas y pedidos. Almacenar todas las líneas de detalle de todos ellos a fin de mantener un histórico de movimientos.

- Guardar el histórico de documentos/facturas emitidas por el departamento de viajes, para su futura conciliación con el sistema contable.
- **Requisitos funcionales de la administración del sistema**
 - Cualquier usuario debe de poder ver el detalle de sus propios datos. Debe ser capaz de modificar casi todos los contenidos de su ficha en el sistema, salvo los permisos (tipo de usuario) y el superior jerárquico.
 - Sólo el usuario con perfil administrador debe de llevar a cabo la gestión de los usuarios. Esta gestión debe integrar: ser capaz de dar de alta a un usuario, editar todos los contenidos de la ficha, eliminarlo, editarlo, y como no, cambiarle el perfil o tipo de usuario, incluso hacerlo administrador si cabe.
 - Sólo el administrador debe ser capaz de ver la pestaña administración del menú principal.
 - El usuario administrador tendrá acceso a todas y cada una de las secciones de la herramienta, en modo consulta y edición. Aunque su misión no será otra que mantener la aplicación y gestionar los usuarios.
 - El usuario administrador podrá delegar la función de administración a ciertos usuarios avanzados del sistema si ellos lo necesitan, siempre bajo la supervisión de un superior jerárquico.

2.2.3 Requisitos funcionales

En cuanto al rendimiento se refiere, no se trata de un aspecto importante en el análisis del sistema. Hemos de tener en cuenta que no se dispone de una aplicación que haga muchos cálculos y requiera de grandes manejos de datos. Tampoco es una aplicación de uso muy constante en tiempo real. Se trata de una aplicación de gestión.

La tecnología del lenguaje de desarrollo no interfiere en demasía en el rendimiento del sistema (ASP). Y la tecnología de base de datos (Microsoft Access) es más que suficiente para soportar el rendimiento del sistema. El único hándicap del sistema en cuanto a rendimiento se refiere, radica en la consulta que se puede realizar del histórico total de movimientos de viajes o facturas, desde el arranque de la aplicación hasta el momento en que se hayan generado muchos movimientos, hablamos de miles de registros, aunque entendemos que esta consulta no es de mucha utilidad y solo la podrían hacer los responsables del departamento de viajes.

2.2.4 Requisitos lógicos de la base de datos

La base de datos constituye un pilar fundamental en la aplicación, en ella se van almacenar todos los datos registrados en la herramienta.

La aplicación solo va hacer uso de una base de datos llamada viajesDATOS. Esta base de datos, bastante amplia y albergada en el servidor de la aplicación, estará formada por un total de 19 tablas y 30 consultas.

No vamos a citar aquí todas las tablas existentes y consultas, pues se detallará en la sección de diseño. Simplemente mencionar que la tabla usuarios será la que albergue los datos de los usuarios así como sus permisos, y será consultada en el arranque del sistema para ofrecer un perfil de conexión al usuario que se conecta.

2.2.5 Restricciones en el diseño del sistema

No existe restricción importante en el diseño de la aplicación. Simplemente el sistema debe cumplir con la especificación de requisitos realizada.

Cabe señalar en este apartado algunas peticiones aportadas por el departamento de contabilidad. El sistema, en este caso la base de datos, está expuesto como todo sistema informático a la pérdida y destrucción de la información. Así pues es necesaria la realización de copias de seguridad tanto de la herramienta como de la base de datos, para tener siempre disponibles los movimientos generados por el sistema, pedidos y detalles de los mismos.

Es importante destacar en este apartado (como se cita en otra sección de la presente memoria) que la herramienta está alojada en una red privada y en un servidor dedicado para varias aplicaciones desarrolladas dentro de la propia compañía. El departamento de informática posee un sistema de copias de *backup* que garantiza la seguridad de los datos y de la propia aplicación. No hay más detalles a mencionar en esta sección.

2.3 Análisis del sistema

2.3.1 Modelos de casos de uso

A continuación mostraremos las especificaciones de los diferentes casos de uso con sus respectivas descripciones y en el siguiente apartado los respectivos diagramas.

2.3.1 Especificaciones de casos de uso

- **Casos de uso destinados al Login**

Caso de uso: Login de Usuario

- **Descripción:** Login del usuario
- **Actores:** Usuario
- **Resumen:** Cuando el usuario arranca la aplicación o desea abrir otra sesión se conecta el sistema con su nombre de login y su password. Después pulsa el botón enviar.
- **Escenario principal**
 1. El caso de uso empieza cuando el usuario accede a la aplicación o inicia una nueva sesión.
 2. EL usuario teclea su login y su password.
 3. El sistema comprueba que tanto su login como su password son correctos y valida el acceso mostrando la aplicación según el perfil del usuario.
- **Escenario alternativo**
 0. En cualquier caso el usuario puede salir del proceso de login.
 3. a. Alguno de los campos del formulario de login esta vacío, se recarga la página con los formularios vacíos.
 3. b. El login o contraseña introducidos no son válidos. Mensaje de error y recarga de la página.

- **Casos de uso destinados a propuestas y viajes**

Caso de uso: Nueva propuesta

Caso de uso: Crear propuesta de viaje

- **Descripción:** Crear propuesta
- **Actores:** Cualquier usuario

- **Resumen:** El usuario accede a la sección de viajes, Pincha en nuevo, elige el modo de creación (usando el asistente o de forma manual) rellena el formulario/os de la propuesta.
- **Escenario principal**
 1. El usuario pincha en nuevo
 2. Asigna una descripción/nombre a la propuesta
 3. El sistema crea una nueva propuesta de viaje y muestra la cabecera del mismo: descripción, estado (pendiente de envío), fecha de creación y con el detalle de la propuesta (de qué está compuesto el viaje)
- **Escenario alternativo**
 0. En cualquier caso el administrador puede cancelar el proceso creación a través del asistente

Caso de uso: Editar propuesta de viaje

- **Descripción:** Editar propuesta
- **Actores:** Cualquier usuario
- **Resumen:** El usuario accede a la sección de viajes, elige la propuesta, pincha en editar la cabecera. Edita las características disponibles. Puede además añadir una línea de detalle.
- **Escenario principal**
 1. El usuario elige la propuesta de lista de propuestas que tiene en registradas
 - 2.
 3. El sistema crea una nueva propuesta de viaje y muestra la cabecera del mismo: descripción, estado (pendiente de envío), fecha de creación y con el detalle de la propuesta (de qué está compuesto el viaje)
- **Escenario alternativo**
 1. En cualquier caso el administrador puede cancelar el proceso creación a través del asistente

Caso de uso: Añadir líneas detalles a la propuesta

- **Descripción:** Añadir detalle
- **Actores:** Cualquier usuario
- **Resumen:** El usuario accede a la sección de viajes, elige la propuesta, va a añadir detalle, selecciona el tipo de artículo del detalle y rellena el formulario del detalle correspondiente.
- **Escenario principal**
 1. El usuario elige la propuesta de la lista de propuestas que tiene en registradas.
 2. Selecciona añadir detalle y rellena el formulario del detalle. Guarda los datos.
 3. El sistema valida los datos introducidos y crea una nueva línea de detalle a la propuesta, con un código de línea de detalle ligada a la propuesta total y asociado como pasajero por defecto al usuario que crea la propuesta.
- **Escenario alternativo**
 0. En cualquier caso el usuario puede cancelar el detalle.
 3. a El usuario deja campos obligatorios en blanco del formulario. Mensaje de error.
 3. b El usuario introduce una fecha con formato incorrecto. Mensaje de error.
 3. c El usuario introduce una hora en el formulario con formato incorrecto. Mensaje de error.

Caso de uso: Quitar líneas de detalle a la propuesta

- **Descripción:** Quitar detalle
- **Actores:** Cualquier usuario
- **Resumen:** El usuario accede a la sección de viajes, elige la propuesta, elige el detalle y va a quitar detalle. El sistema pide confirmación de eliminación y en caso afirmativo, elimina el detalle de la propuesta.
- **Escenario principal**
 1. El usuario elige la propuesta de la lista de propuestas que tiene registradas y selecciona el detalle a eliminar.

2. El usuario ejecuta la acción de quitar detalle. El sistema pide confirmación y elimina la línea de detalle.

➤ **Escenario alternativo**

0. El usuario cuando el sistema le pide confirmación de eliminación, puede elegir respuesta negativa y abortar así la eliminación.

Caso de uso: Editar líneas detalles de la propuesta

➤ **Descripción:** Editar detalle

➤ **Actores:** Cualquier usuario

➤ **Resumen:** El usuario accede a la sección de viajes, elige la propuesta, va a la sección de detalles, editar detalle. Una vez ahí dentro, el proceso es como si creara un nuevo detalle. El detalle final quedará modificado.

➤ **Escenario principal**

1. El usuario elige la propuesta de lista de propuestas que tiene registradas
2. Selecciona editar detalle y rellena el formulario del detalle. Guarda los datos.
3. El sistema valida los datos introducidos y modifica el detalle con el mismo sistema que en el caso de uso de creación de detalle.

➤ **Escenario alternativo**

Mismo escenario que el caso de uso de creación de detalle.

Caso de uso: Editar líneas detalles de la propuesta

➤ **Descripción:** Editar detalle

➤ **Actores:** Cualquier usuario

➤ **Resumen:** El usuario accede a la sección de viajes, elige la propuesta, va a la sección de detalles, editar detalle. Una vez ahí dentro el proceso es como si creara un nuevo detalle. El detalle final quedará modificado.

➤ **Escenario principal**

1. El usuario elige la propuesta de lista de propuestas que tiene en registradas

2. Selecciona editar detalle y rellena el formulario del detalle. Guarda los datos.
3. El sistema valida los datos introducidos y modifica el detalle con el mismo sistema que en el caso de uso de creación de detalle.

➤ **Escenario alternativo**

Mismo escenario que el caso de uso de creación de detalle.

Caso de uso: Configurar habitación doble de hotel

➤ **Descripción:** Habitación doble

➤ **Actores:** Cualquier usuario

➤ **Resumen:** El usuario si tiene dos habitaciones dobles con dos pasajeros, tiene la opción de solicitar la fusión de las dos habitaciones en una habitación doble. Y después de validar el detalle el sistema fusionará las dos líneas de detalle en una.

➤ **Escenario principal**

1. El usuario elige los dos detalles con hotel y ejecuta la acción de habitación doble.
2. El sistema unifica las dos líneas de detalle en una con los pertinentes cambios en el almacén de datos. La línea pasa a tener el mismo artículo pero con los dos pasajeros en dos camas.

➤ **Escenario alternativo**

0. El usuario puede abortar el proceso de habitación doble.
2. a El usuario selecciona un número de líneas de detalle con hotel distinto de 2, mensaje de error.
2. b El usuario intenta integrar dos hoteles con fechas distintas, el sistema advierte de ello.

Caso de uso: Repartir Costes del viaje

➤ **Descripción:** Repartir Costes

➤ **Actores:** Cualquier usuario

- **Resumen:** El usuario elige una de los viajes creados. Accede al detalle de la cabecera a través de la opción de editar. Selecciona la opción modificar porcentaje de coste. Reparte el porcentaje de coste entre los centros de forma que sume 100.

- **Escenario principal**
 1. El usuario elige el viaje. Accede a Editar cabecera propuesta. Modificar porcentaje.
 2. El usuario reparte el porcentaje de coste entre los centros que aparecen. Y guarda los cambios.
 3. El sistema comprueba los valores del formulario.

- **Escenario alternativo**
 0. El usuario puede abandonar el proceso de modificación.
 1. La suma de porcentajes introducidas por el usuario no suma 100. Mensaje de error.

Caso de uso: Añadir y/o quitar centros de coste del viaje

- **Descripción:** Añadir/Quitar centros de coste

- **Actores:** Cualquier usuario

- **Resumen:** El usuario elige una de los viajes creados. Accede al detalle de la cabecera a través de la opción de editar. Selecciona la opción Añadir/quitar centro de coste. Selecciona los centros de coste que formarán el reparto.

- **Escenario principal**
 1. El usuario elige el viaje. Accede a Editar cabecera propuesta. Añadir/Quitar centros de coste.
 2. El usuario elige los centros de coste. Guarda cambios.
 3. El sistema comprueba los valores del formulario.

- **Escenario alternativo**
 0. El usuario puede abandonar el proceso.
 2. El usuario no elige ningún centro de coste. Mensaje de error.

Caso de uso: Transformar propuesta en pedido

- **Descripción:** Enviar pedido
- **Actores:** Todos los usuarios con acceso al sistema.
- **Resumen:** El usuario elige una de las propuestas. Ejecuta la acción de enviar pedido. El sistema le sugiere un proveedor para el pedido. El usuario lo selecciona y propone el envío, tras una petición de confirmación el sistema crea un pedido asociado a la propuesta.
- **Escenario principal**
 1. El usuario elige el viaje. Accede al detalle de la propuesta.
 2. El usuario elige los centros de coste. Guarda cambios. Ejecuta la opción de enviar pedido al proveedor seleccionado (el usuario centros no verá esta opción).
 3. El sistema comprueba los datos de la propuesta y registra el pedido con los detalles asociados. Si se trata de un usuario de centros, el sistema genera un correo electrónico al departamento de viajes y la propuesta pasará a formar parte de la lista de propuestas a tratar por el departamento de viajes.
- **Escenario alternativo**
 1. El usuario puede abandonar el proceso.
 3. a. El usuario no elige ningún motivo ni centro de coste. Mensaje de error.

▪ **Casos de uso destinados a pedidos**

Caso de uso: Eliminar detalle del pedido

- **Descripción:** Eliminar detalle pedido
- **Actores:** Usuarios central. Usuarios contables, usuario de viajes y administrador.
- **Resumen:** El usuario elige uno de los pedidos. Selecciona un detalle del pedido y ejecuta quitar detalle.

➤ **Escenario principal**

1. El usuario elige un viaje tramitado. Accede al detalle del pedido.
2. El usuario ejecuta la eliminación del detalla y confirma cambios.
3. El sistema comprueba elimina la línea de detalle del registro. El usuario deberá reenviar el pedido al proveedor si quiere que se tomen en cuenta los cambios.

Caso de uso: Añadir detalle del pedido

➤ **Descripción:** Añadir detalle pedido

➤ **Actores:** Usuarios central. Usuarios contables, usuario de viajes y administrador

➤ **Resumen:** El usuario elige uno de los pedidos. Selecciona un detalle del pedido y ejecuta añadir detalle.

➤ **Escenario principal**

1. El usuario elige un viaje tramitado. Accede al detalle del pedido.
2. El usuario ejecuta la opción añadir detalla y confirma cambios.
3. El sistema presenta un formulario idéntico al de creación de detalle de la propuesta, y el usuario rellena el formulario.

➤ **Escenario alternativo**

2. El usuario puede abandonar el proceso de añadir.

El resto de casos alternativos quedan contemplados en el caso de uso de creación y edición de detalle de propuesta.

Caso de uso: Crear pedido a partir de líneas de propuesta sin tratar

➤ **Descripción:** Crear pedido

➤ **Actores:** Usuarios central. Usuarios contables, usuario de viajes y administrador

➤ **Resumen:** El usuario elige uno de los viajes tramitados, accede al detalle, selecciona las líneas de propuestas sin tratar, y ejecuta la acción de crear

pedido. Posteriormente el sistema crea el pedido con las líneas de detalle asociadas.

➤ **Escenario principal**

1. El usuario elige un viaje tramitado. Accede al detalle del pedido.
2. El usuario selecciona las líneas de propuesta que quiere pasar al pedido y ejecuta la opción crear pedido y confirma cambios.
3. El sistema crear un pedido lo registra con un código de pedido, y registra las líneas de detalle del pedido cada uno con un código de línea.

➤ **Escenario alternativo**

3. El usuario puede abandonar el proceso de añadir.
- El resto de casos alternativos quedan contemplados en

Caso de uso: Enviar Pedido

➤ **Descripción:** Enviar pedido

➤ **Actores:** Todos los usuarios menos centros

➤ **Resumen:** El usuario elige uno de los viajes tramitados, accede al detalle, selecciona un pedido y ejecuta la opción de enviar. Esto lo que hace es enviar al proveedor por defecto un e-mail con el pedido a gestionar.

➤ **Escenario principal**

1. El usuario elige un viaje tramitado. Accede al detalle del pedido.
2. El usuario ejecuta enviar pedido y confirma acción.
3. El sistema valida el estado del pedido, pide confirmación y luego cambia el estado del pedido. Envía un correo al proveedor.

➤ **Escenario alternativo**

0. El usuario puede denegar la operación en el momento de confirmación.

▪ **Casos de uso destinados a usuarios**

Caso de uso: Alta de Usuario

➤ **Descripción:** Alta del usuario

➤ **Actores:** Usuario Administrador

- **Resumen:** Si el usuario ya está creado en el sistema, el administrador desea dar de alta a un usuario en la aplicación, introduce sus un login y un e-mail para el usuario y acepta la operación.
- **Escenario principal**
 1. El administrador empieza a buscar un usuario.
 2. EL administrador rellena los campos pertinentes.
 3. El sistema comprueba los campos introducidos.
 4. El sistema asigna al usuario los permisos pertinentes
- **Escenario alternativo**
 3. En cualquier caso el administrador puede salir del proceso de alta
 3. a. El campo del e-mail está vacío

Caso de uso: Creación de un nuevo Usuario

- **Descripción:** Nuevo usuario
- **Actores:** Usuario Administrador
- **Resumen:** El administrador desea crear a un usuario en la aplicación. El administrador accede a la sección Administración. Pincha en el botón nuevo, nuevamente pincha en nuevo. Rellena los campos del formulario de creación. Valida la operación.
- **Escenario principal**
 1. El administrador pincha en nuevo usuario.
 2. EL administrador rellena los campos pertinentes. (Nombre, apellidos, DNI y centro) son obligatorios.
 3. El sistema comprueba los campos introducidos.
 4. El sistema crea al usuario con un codusr que le identifica.
- **Escenario alternativo**
 4. En cualquier caso el administrador puede salir del proceso de creación.
 3. a. Alguno de los campos obligatorios están vacíos.
 3. b El campo DNI está duplicado en el sistema. Mensaje de error.

Caso de uso: Dar de baja a un usuario

- **Descripción:** Baja de usuario
- **Actores:** Usuario Administrador
- **Resumen:** El administrador desea dar de baja a un usuario de la aplicación, a través de la lista de usuarios de la sección Administración, elige el usuario. Y en el formulario del usuario le cambia el perfil a baja. Valida la operación.
- **Escenario principal**
 1. El administrador empieza eligiendo a un usuario.
 2. EL administrador cambia el perfil a baja. Pincha en guardar.
 3. El sistema asigna al usuario los permisos de baja en el sistema(-1).
- **Escenario alternativo**
 0. En cualquier caso el administrador puede salir del proceso de baja

Caso de uso: Editar a un Usuario

- **Descripción:** Editar un usuario
- **Actores:** Usuario Administrador
- **Resumen:** El administrador desea editar los datos de un usuario de la aplicación, a través de la lista de usuarios de la sección Administración, elige el usuario. Y en el formulario del usuario modifica sus datos. Valida la operación.
- **Escenario principal**
 1. El administrador empieza elige a un usuario.
 2. EL administrador cambia los datos. Pincha en guardar.
 3. El sistema cambia los datos pertinentes.
- **Escenario alternativo**
 0. En cualquier caso el administrador puede salir del proceso de edición

3. a El login que modifica ya existe en el sistema. Mensaje de error

Caso de uso: Listar Usuarios

- **Descripción:** Listar usuarios.
- **Actores:** Usuario Administrador.
- **Resumen:** El administrador desea generar una lista de usuarios según el perfil, aplica filtro de perfil.
- **Escenario principal**
 1. El administrador elige un filtro para listar
 2. Selecciona el filtro.
 3. El sistema genera una lista de los usuarios con el perfil seleccionado
- **Escenario alternativo**

No existen casos alternativos para esta operación.

Caso de uso: Modificar datos de usuario

- **Descripción:** Modificar ficha usuario
- **Actores:** Usuario.
- **Resumen:** El usuario desea modificar sus datos de usuario, a través de la sección Mis datos de la herramienta.
- **Escenario principal**
 1. El usuario accede a su ficha de usuario.
 2. Modifica los campos que desea, menos los reservados para la administración (login, tipo de usuario, superior jerárquico)
 3. El sistema valida los cambios y los guarda en el sistema.

➤ **Escenario alternativo**

3. El DNI introducido ya existe (duplicado) Mensaje de error.

▪ **Casos de uso destinados a Facturas**

Caso de uso: Recepcionar Pedido o Generar Factura

➤ **Descripción:** Recepcionar pedido

➤ **Actores:** usuario de viajes

- **Resumen:** El usuario elige uno de los viajes tramitados, accede al detalle, selecciona las líneas de pedido a facturar, y ejecuta la acción de generar documento. Posteriormente el usuario de viajes rellena el formulario de factura y guarda datos.

➤ **Escenario principal**

1. El usuario elige un viaje tramitado. Accede al detalle del pedido.
2. El usuario selecciona las líneas de pedido que quiere recepcionar y ejecuta generar documento.
3. El sistema valida el formulario y lo registra con un código de factura.

➤ **Escenario alternativo**

0. El usuario siempre puede abortar el proceso.
3. a. El usuario no selecciona líneas a generar documento.
3. b. En el formulario de la factura no rellena algún campo.
3. c. No selecciona el detalle a facturar

Caso de uso: Listar Facturas

➤ **Descripción:** Listar Facturas

➤ **Actores:** usuario contable

- **Resumen:** El usuario contable accede a los informes de facturas enviados a contabilidad. Una vez ahí a través de un filtro puede generar listas de facturas.

- **Escenario principal**

1. El usuario elige contable accede a la sección de contabilidad.
2. El usuario aplica un filtro y el sistema aplica el filtro y genera una lista.

- **Escenario alternativo**

No es importante en este apartado recalcar escenarios alternativos.

Caso de uso: Conciliar facturas

- **Descripción:** Conciliar factura

- **Actores:** usuario contable

- **Resumen:** El usuario contable accede a los informes de facturas enviados a contabilidad. Una vez ahí selecciona las facturas que desea conciliar y ejecuta la acción correspondiente.

- **Escenario principal**

1. El usuario elige contable accede a la sección de contabilidad.
2. El usuario aplica filtro, señala las facturas a conciliar y ejecuta la conciliación.

- **Escenario alternativo**

0. El usuario puede denegar la operación de confirmación.

Caso de uso: Generar reporte contable

- **Descripción:** Generar xls

- **Actores:** usuario contable

- **Resumen:** El usuario contable accede a los informes de facturas enviados a contabilidad. Una vez ahí y tras aplicar el filtro deseado, puede generar un documento hoja de cálculo con la tabla deseada.

➤ **Escenario principal**

1. El usuario elige contable accede a la sección de contabilidad.
2. El usuario aplica filtro, señala las facturas y ejecuta la acción.

➤ **Escenario alternativo**

1. El usuario puede denegar la operación de confirmación.

Caso de uso: Generar reporte contable2

➤ **Descripción:** Generar pendientes

➤ **Actores:** usuario contable

➤ **Resumen:** El usuario contable accede a los informes de pedidos pendientes de envío a contabilidad. Una vez ahí y tras aplicar el filtro deseado puede generar un documento hoja de cálculo con la tabla deseada.

➤ **Escenario principal**

1. El usuario elige contable accede a la sección de contabilidad.
2. La tabla visualizada la puede pasarla a un fichero de hoja de cálculo para su futuro tratamiento.

➤ **Escenario alternativo**

0. El usuario puede denegar la operación de confirmación.

Caso de uso: Modificar cuenta contable

➤ **Descripción:** Modificar Cuenta contable

➤ **Actores:** usuario contable

➤ **Resumen:** El usuario contable en la sección de contabilidad en la lista de facturas pasadas por el departamento de viajes puede acceder al detalle del pedido y modificar la cuenta contable asociada al gasto si cabe.

➤ **Escenario principal**

1. El usuario elige contable accede a la sección de contabilidad.
2. Accede al detalle del pedido pasado por factura. Ejecuta la acción de modificar cuenta contable. Si el usuario contable modifica este campo, el usuario propietario del viaje ya no podrá modificar este apartado.

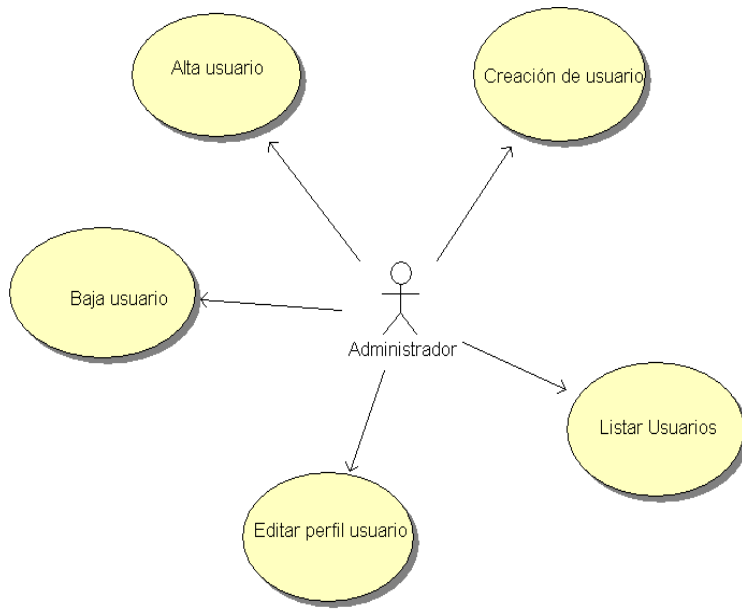
➤ **Escenario alternativo**

0. El usuario puede abortar la operación y no modificar nada

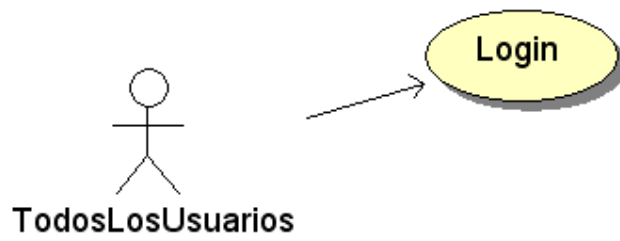
2.3.2 Especificaciones de casos de uso

A continuación, pasaremos a representar los diferentes diagramas de casos de uso de cada uno de los componentes que integran el sistema.

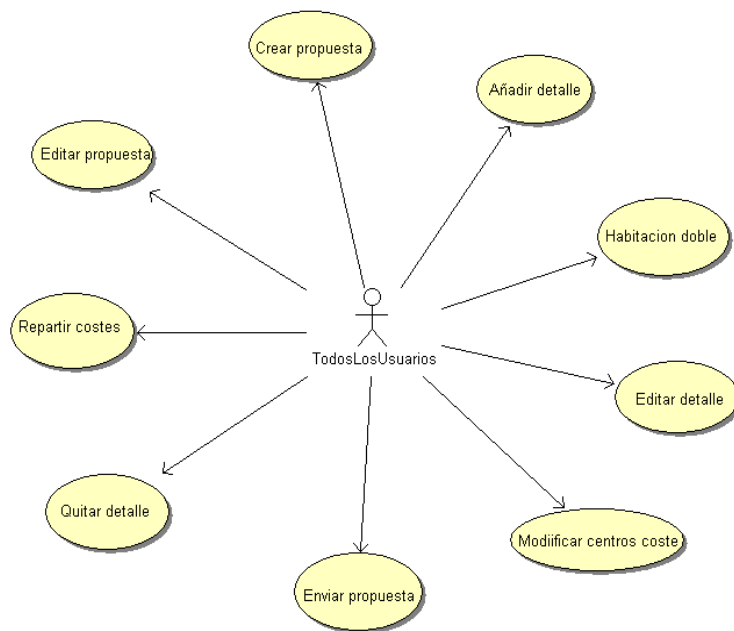
➤ **Diagrama de uso. Administración de usuarios**



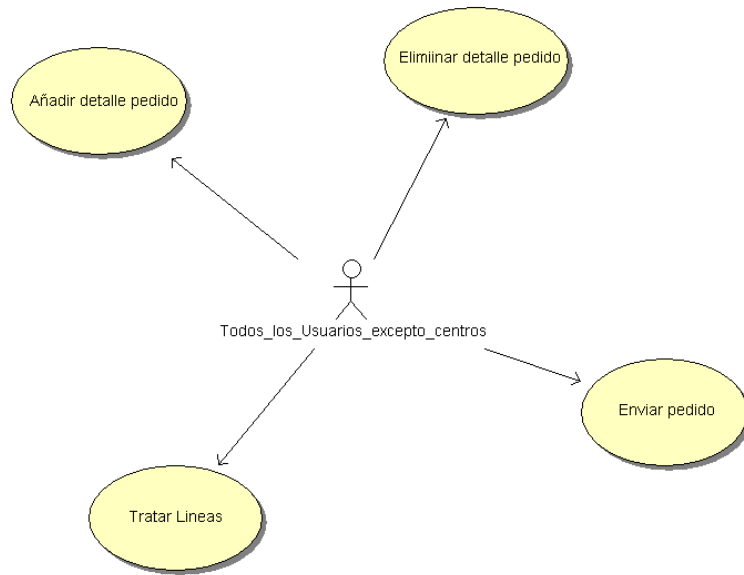
➤ Diagrama de uso del Login de Usuario



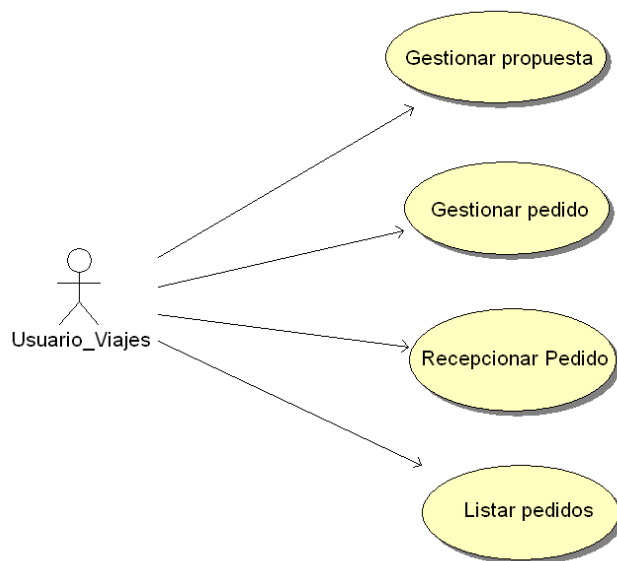
➤ Diagrama de uso Propuestas de viaje



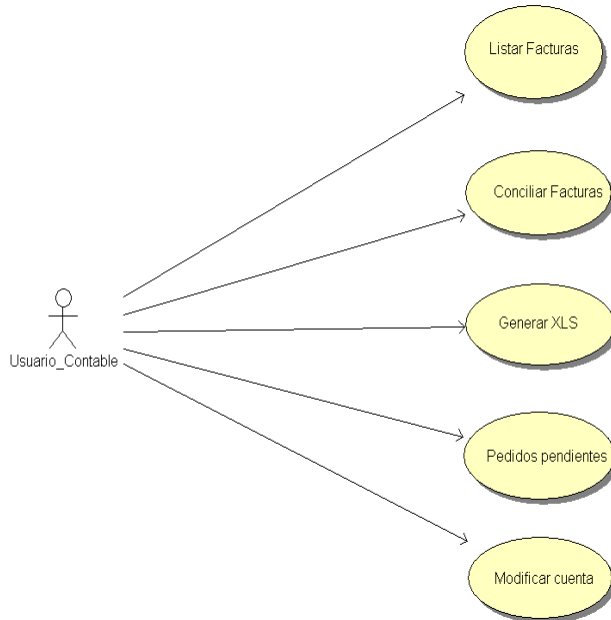
➤ **Diagrama de uso pedidos de viaje**



➤ **Diagrama de uso Departamento de Viajes**



➤ Diagrama de uso Departamento Contable



2.4 Desarrollo del calendario

En esta sección podemos visualizar la planificación temporal en cuanto al desarrollo del proyecto se refiere.

Esta se ve reflejada mediante un diagrama de Gantt.

2.4.1 Descripción de la herramienta

El diagrama de Gantt, gráfica de Gantt o carta Gantt es una popular herramienta gráfica cuyo objetivo es mostrar el tiempo de dedicación previsto para diferentes tareas o actividades a lo largo de un tiempo total determinado.

Desde su introducción los diagramas de Gantt se han convertido en una herramienta básica en la gestión de proyectos de todo tipo, con la finalidad de representar las diferentes fases, tareas y actividades programadas como parte de un proyecto o para mostrar una línea de tiempo en las diferentes actividades haciendo el método más eficiente.

Básicamente el diagrama esta compuesto por un eje vertical donde se establecen las actividades que constituyen el trabajo que se va a ejecutar, y un eje horizontal que muestra en un calendario la duración de cada una de ellas.

2.4.2 Descripción de las tareas y calendario

A continuación se indica un resumen de todas las tareas reflejadas en el gráfico:

- **Fase de inicio: [01/07/2010 ,15/08/2010] El objetivo de esta fase de inicio es la puesta en marcha del proyecto.**

Idea del proyecto: [01/07/2010, 04/07/2010]. En esta fase se contemplaron varias ideas de proyecto, hasta concretar en una sola. Hay que definir correctamente esa idea y ser capaz de exponerla a los tutores y en este caso también al departamento responsable del proyecto.

Solicitud formal al tutor del proyecto: [04/07/2010, 15/07/2010] En este periodo De tiempo realizamos la formalización con el tutor del proyecto, es el momento de exponerle las ideas. Cabe añadir que estas ideas deben estar debidamente argumentadas para que el tutor las acepte y de vía a libre al proyecto.

Entrevista con el responsable del proyecto: [04/07/2010, 15/07/2010] En las entrevistas con el jefe del proyecto, primero le indicamos que ofrecerle y que beneficios tendría la utilización del producto. Luego, es el propio responsable el que nos encarga el producto según las características de su empresa, aportándonos la funcionalidad que desea que integre.

Búsqueda de bibliografía: [15/07/2010, 25/07/2010] Durante esta etapa había que localizar la mayor cantidad de bibliografía. Esta información se podría localizar buscando en bibliotecas, navegando por Internet o recopilando apuntes de la titulación.

Selección de la bibliografía: [25/07/2010, 29/07/2010] De la bibliografía anterior tendría que existir una selección que verdaderamente nos interesara, desechando la que no procesa ya sea por que no recoge información útil, está incompleta o simplemente porque la fuente de donde procede no es segura.

Creación del prototipo: [29/07/2010 ,20/08/2010] Con ayuda de algún compañero del departamento se crea un prototipo de la interfaz que tendría la aplicación con el usuario.

Este prototipo debe de ser probado por el departamento, para ello se irán realizando las modificaciones oportunas hasta ajustarlo a su gusto personal, tampoco se requiere demasiada complicación, importa más la funcionalidad y la sencillez.

Planificación del proyecto: [20/08/2010, 23/08/2010] En esta etapa elaboramos toda la planificación del proyecto. Se calculan los tiempos necesarios para cada una de las tareas a realizar. Entre tarea y tarea dejaremos unos días de margen, ya que sino no se estarán cumpliendo los plazos.

▪ **Fase de ejecución:** [23/08/2010 , 13/11/2010] **Esta es la fase del desarrollo del proyecto en sí mismo.**

Análisis de requisitos: [23/08/2010 ,28/08/2010] En esta fase y con ayuda de toda la información obtenida tanto en reuniones con los tutores y con el departamento, obtenemos los requisitos de nuestra aplicación. Hay que distinguirlos bien ya que podemos encontrarnos con requisitos incompletos, ambiguos o contradictorios.

Especificación: [28/08/2010, 02/09/2010] Aquí se debe indicar el comportamiento que tendría el sistema una vez desarrollado. Hay que dedicar tiempo a localizar todos los posibles errores para que luego no nos encontremos la aplicación sin toda la funcionalidad deseada o simplemente que un proceso no nos de los resultados esperados.

Diseño y codificación: [02/09/2010, 07/11/2010] Es el momento de empezar a determinar como va a estar "montada la aplicación". En esta etapa mediante técnicas y principios adquiridos tendremos que ser capaces de permitir la interpretación del sistema así como la capacidad de su realización física. Después de esta etapa habría que llevar a código el diseño realizado. Además también tendremos que elegir el lenguaje de programación que vamos a utilizar, el gestor de la base de datos, en definitiva todos los elementos necesarios para crear físicamente la aplicación.

Pruebas: [07/11/2010, 13/11/2010] Esta es la última etapa del período de ejecución. En ella tendremos que comprobar que nuestra aplicación cumple con todo lo especificado en la etapa de especificación. Además será necesaria la ayuda de varios compañeros para que prueben el software generado.

▪ **Fase final:** [13/11/2010, 13/11/2010] **Esta es la etapa final del desarrollo del proyecto.**

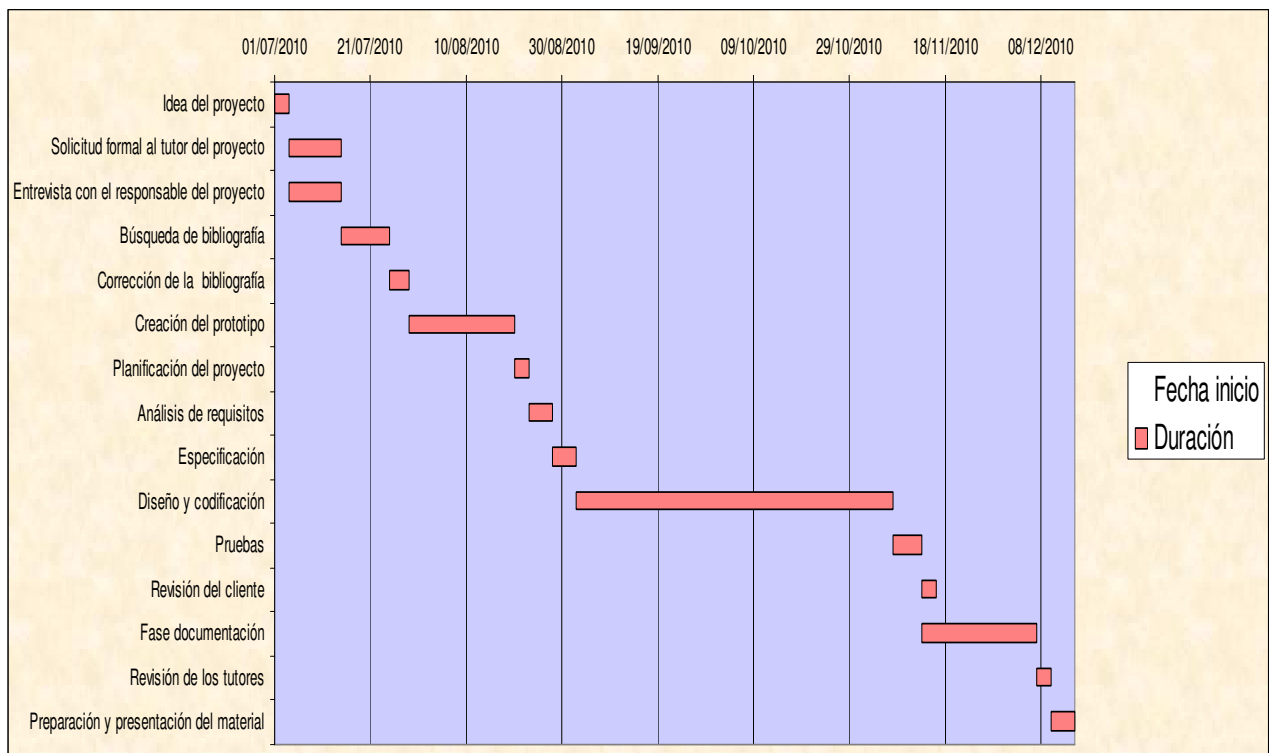
Revisión del cliente: [13/11/2010 ,16/11/2010] Se hace la entrega del proyecto a nuestra compañía. Durante esta tarea, el departamento tendrá que valorar todo nuestro trabajo, como los tutores, tendrá que comprobar que cumple las condiciones e ideas pactadas al inicio.

Fase documentación: [16/11/2010, 07/12/2010] En esta fase es donde se va a ir generando toda la documentación. Según en la tarea en la cual nos encontremos de la fase de ejecución, habrá que generar un tipo de documentación y unos diagramas o gráficos determinados. De la documentación se generará esta memoria.

Revisión del tutor: [07/12/2010, 10/12/2010] Nuestro tutor revisa todo nuestro trabajo, comprobará si todo nuestro proyecto cumple las condiciones y las ideas iniciales.

Preparación y presentación del material: [10/12/2010, 14/12/2010] En este momento, Finalmente nos dispondremos a entregar todo el material a la Universidad.

2.5 Diagrama de Gantt



Capítulo 3

Diseño

3.1 Diseño del sistema

3.1.1 Arquitectura del sistema

En la elección de la arquitectura software de la aplicación se han sopesado distintas opciones a favor de conseguir un sistema que fuese fácil de mantener, totalmente adaptable a la arquitectura de la plataforma software de la compañía, utilizando para ello las herramientas disponibles y las licencias existentes.

- **Capa de gestión de datos:** Se sitúa en el nivel más bajo de la abstracción, y es el responsable de interactuar con la base de datos del sistema o con el gestor de ficheros. El nuestro, debe interactuar con la base de datos a través de una conexión ODBC Microsoft Access, definida en el Servidor Web del sistema.
- **Capa de dominio:** Capa intermedia del sistema, encargada de implementar las funciones y el código necesario de la aplicación.
- **Capa de presentación:** Capa del nivel más alto de abstracción y cercana al usuario, es la encargada de la interacción con el usuario final del sistema.

3.1.1.2 Lenguajes y codificación

- **HTML**

El esqueleto de la aplicación está implementado en HTML. HTML, siglas de Hypertext Markup Language (lenguaje de marcas de hipertexto) es el lenguaje marcado predominante para la construcción de páginas Web.

A su vez HTML es un lenguaje muy sencillo que permite describir hipertexto con enlaces que conducen a otros documentos e informaciones relacionadas, con inserciones incluso de elementos multimedia (imágenes, gráficos, sonido).

- **ASP**

ASP, desarrollado por Microsoft, es un lenguaje desarrollado a finales de los años noventa para llenar el vacío que existía en los lenguajes de programación. Se puede utilizar a través del protocolo HTTP. Su nombre surge de las siglas en inglés “active Server pages” y fue introducido junto a la aplicación de servidor Web “Internet Information System” o IIS.

ASP es un lenguaje muy sencillo basado en lenguaje de programación Basic, con una estructura paralela a Visual Basic que permite a cualquier programador desarrollar aplicaciones utilizables mediante el sistema de servidores de Windows, que es el que utilizan en la compañía de este sistema.

Para que el servidor del sistema interprete el código es necesario poner como extensión de las páginas web la extensión .asp.

- **JavaScript**

“JavaScript es un lenguaje de programación interpretado, es decir, que no requiere compilación, utilizado también en nuestras páginas Web con una sintaxis semejante a la del lenguaje Java y el lenguaje C.

Al igual que Java, JavaScript es un lenguaje orientado a objetos propiamente dicho, ya que dispone de herencia, si bien esta se realiza siguiendo el paradigma de programación basada en prototipos, ya que las nuevas clases se generan clonando las clases base (prototipos) y extendiendo su funcionalidad.

Todos los navegadores modernos interpretan el código JavaScript integrado dentro de las páginas Web. Para interactuar con una página Web se provee al lenguaje JavaScript de una implementación del DOM.

El lenguaje fue inventado por Brendan Eich en la empresa Netscape Communications, que es la que desarrolló los primeros navegadores Web comerciales. Apareció por primera vez en el producto de Netscape llamado Netscape Navigator 2.0.

Tradicionalmente, se venía utilizando en páginas Web HTML, para realizar tareas y operaciones en el marco de la aplicación únicamente cliente, sin acceso a funciones del servidor. JavaScript se ejecuta en el agente de usuario que utilice nuestro sistema, al mismo tiempo que las sentencias van descargándose junto con el código HTML.

- **CSS**

“Las hojas de estilo en cascada (Cascading Style Sheets, CSS) son un lenguaje formal usado para definir la presentación de un documento estructurado escrito en HTML o XML (y por extensión en XHTML). El W3C (World Wide Web Consortium) es el encargado de formular la especificación de las hojas de estilo que servirán de estándar para los agentes de usuario o navegadores.

La idea que buscamos detrás del desarrollo de CSS es separar la estructura de un Documento, de su presentación.

La información de estilo puede ser adjuntada tanto como un documento separado o en el mismo documento HTML. En este último caso podrían definirse estilos generales en la cabecera del documento o en cada etiqueta particular mediante el atributo "style".

3.1.1.2 Herramientas para el desarrollo

- **StarUML**

Se ha utilizado como herramienta de diseño de los diagramas UML.

StarUML es una herramienta para el modelamiento de software basado en los estándares UML (Unified Modeling Language) y MDA (Model Driven Architecture), que en un principio era un producto comercial y que hace cerca de un año paso de ser un proyecto comercial (anteriormente llamado plastic) a uno de licencia abierta GNU/GPL.

En nuestro caso, hemos hecho uso del módulo “StarUML-WAECClass” para nuestro diseño de diagramas UML WAE.

- **Expresión Web y Microsoft FrontPage**

Utilizado como el editor de código de las páginas de la aplicación. Ambos son editores de programación con entorno de trabajo personalizables, con herramientas que permiten crear, editar, validar, previsualizar y publicar páginas Web.

Los editores comentados soportan los lenguajes utilizados y descritos previamente (HTML, XHTML, JavaScript, ASP, CSS, SQL, Java...) y son válidos así como suficientes para el diseño y desarrollo de nuestro sistema Web.

- **Microsoft Access 2003**

Utilizado para la gestión, creación y mantenimiento de nuestra base de datos.

Permite crear, eliminar bases de datos, crear, modificar y eliminar tablas. Ejecutar sentencias de SQL Standard, y se encuentra disponible bajo licencia oficial de la compañía.

- **Microsoft Excel 2003**

Utilizado para la elaboración del diagrama de Gantt, así como para exportar los informes de contabilidad y realizar diversos cálculos.

- **IIS. Internet Information System**

Hemos utilizado esta herramienta para el soporte servidor de la aplicación, bajo licencia Windows de la compañía.

Esta herramienta permite crear, manipular y gestionar fácilmente sitios Web con soporte integrado, que permite la ejecución de nuestras páginas con lenguaje asp.

3.1.2 UML Extensión WAE

Sistema Web

El servidor Web ofrece páginas Web y otros recursos (CSS, imágenes, jsp). Estos recursos se identifican de forma única mediante URL y/o URI. Para representar el diseño de nuestro proyecto aplicando un Standard de molado y representarlos mediante diagramas que relacionan los recursos del sistema, hemos seleccionado una extensión mas o menos estandarizada de UML. Este estándar se denomina WAE.

Es el único exclusivamente basado en UML. Ha sido desarrollado por Jim Conallen. . WAE como UML es recomendado usarlo en lenguajes orientados a objetos. Jim opta por ampliar UML sencillamente porque es más barato hacer un estándar ampliando que creándolo de cero. Lo primero que se plantea es que las aplicaciones Web presentan problemas que UML no contempla solución. UML no facilita la tarea de diferenciar código cliente (scripts) de código servidor. UML puede ser extendido para permitir una nueva semántica: estereotipos, listados de etiquetas (tags) y restricciones (constraints).

Metodología

Tipos de estereotipos empleados y entidades:

- **Server Page:** Son como dice el nombre, las páginas del lado del servidor, esto es, las páginas que contienen scripts o código ejecutado del lado del servidor (.php, .asp, .jsp, ...)
- **Client Page:** Páginas que están del lado del cliente, normalmente estáticas con código mayoritariamente HTML y scripts (java scripts).
- **Form:** Es la representación pura de un formulario. Básicamente es código html que contiene etiquetas del estilo: input, select, etc.
- **Frameset:** Páginas del lado del cliente o servidor, que contienen varias páginas divididas en distintas secciones o frames.
- **ObjectScript:** Páginas o mejor dicho, librerías con funciones java script que utilizan otras páginas para la ejecución de su código.

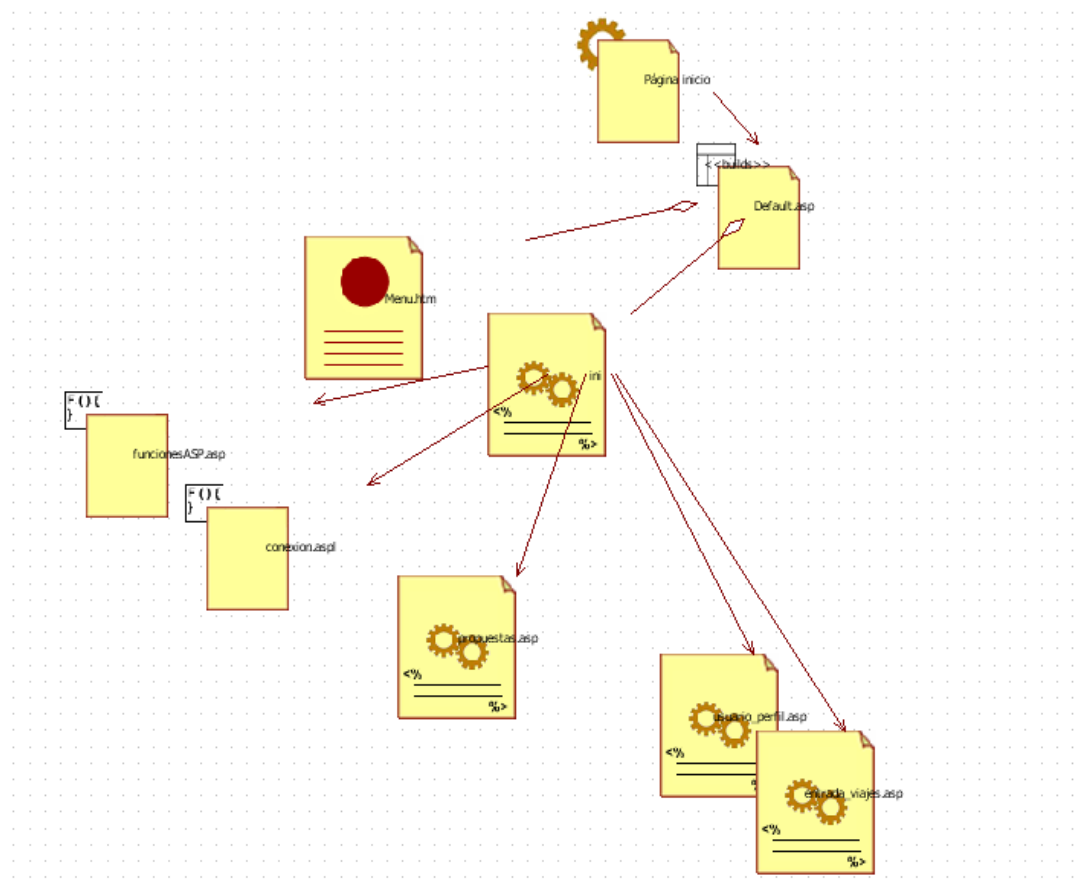
Tipos de relaciones empleadas entre los estereotipos y entidades:

- **<<buid>>** Una relación entre páginas que indica que la página servidor construye a otra página.
- **<<link>>** Simple relación entre dos páginas.
- **<<submit>>** Relación entre un formulario y otra página servidora del sistema Web.
- <<Redirect>>** Relación que indica la redirección de una página a otra.
- **<<uses>>** Relación que indica que una página utiliza código de otra página, es decir algo así como librería del sistema.

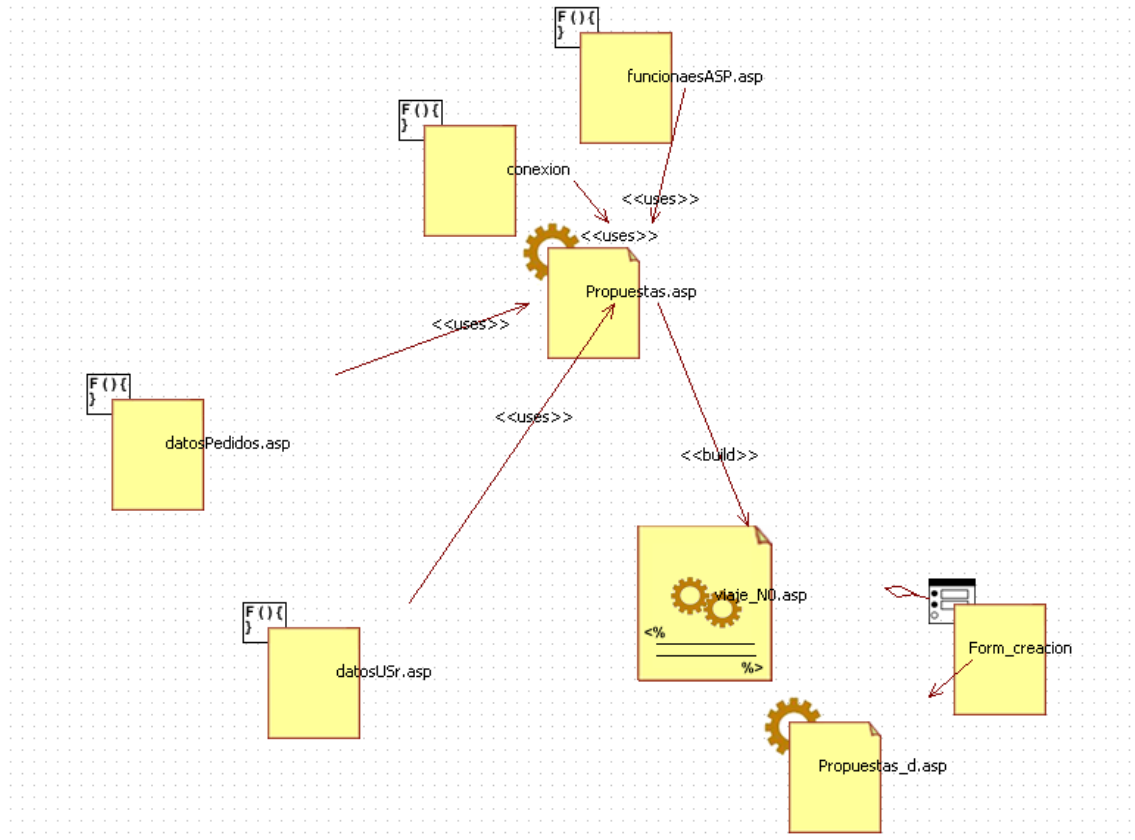
Diagramas WAE del sistema Web

Debido a la complejidad del sistema y puesto que hay muchas páginas y enlaces hemos mostrado aquí las principales vistas de diseño de las gestiones y acciones más importantes que los usuarios pueden realizar en el sistema, descritas ya anteriormente en los requisitos y casos de uso

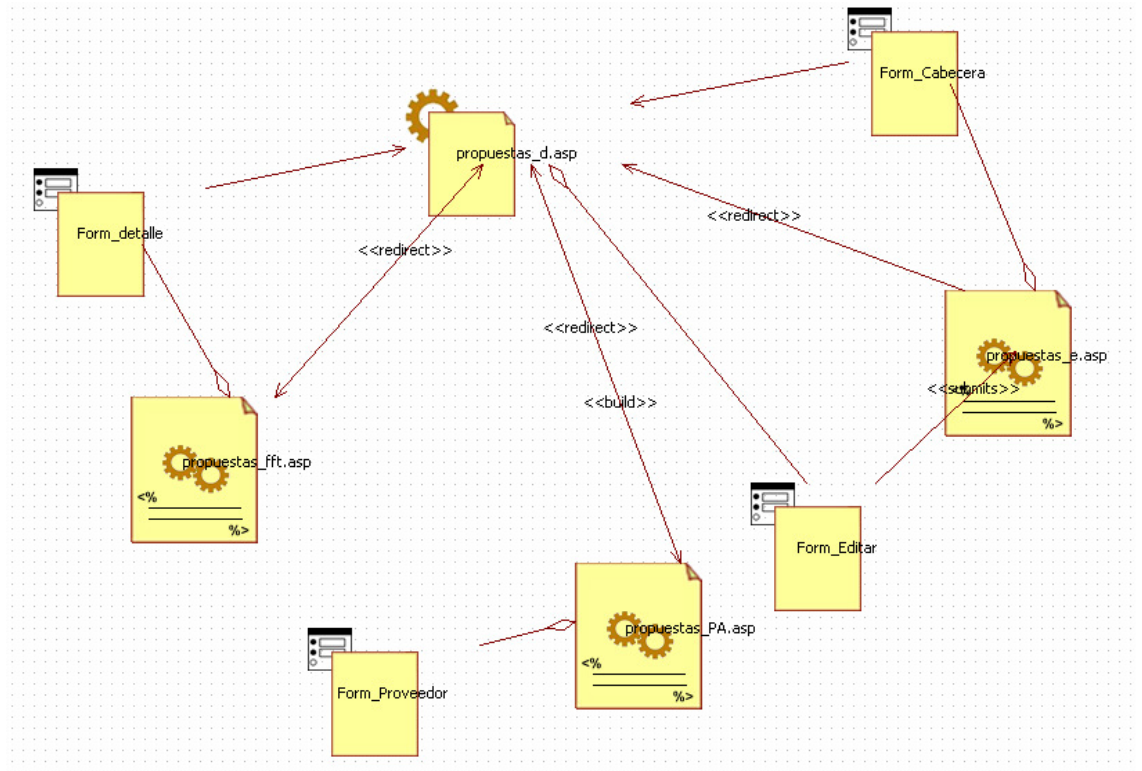
➤ Diagrama WAE para el login del usuario. Entrada al sistema



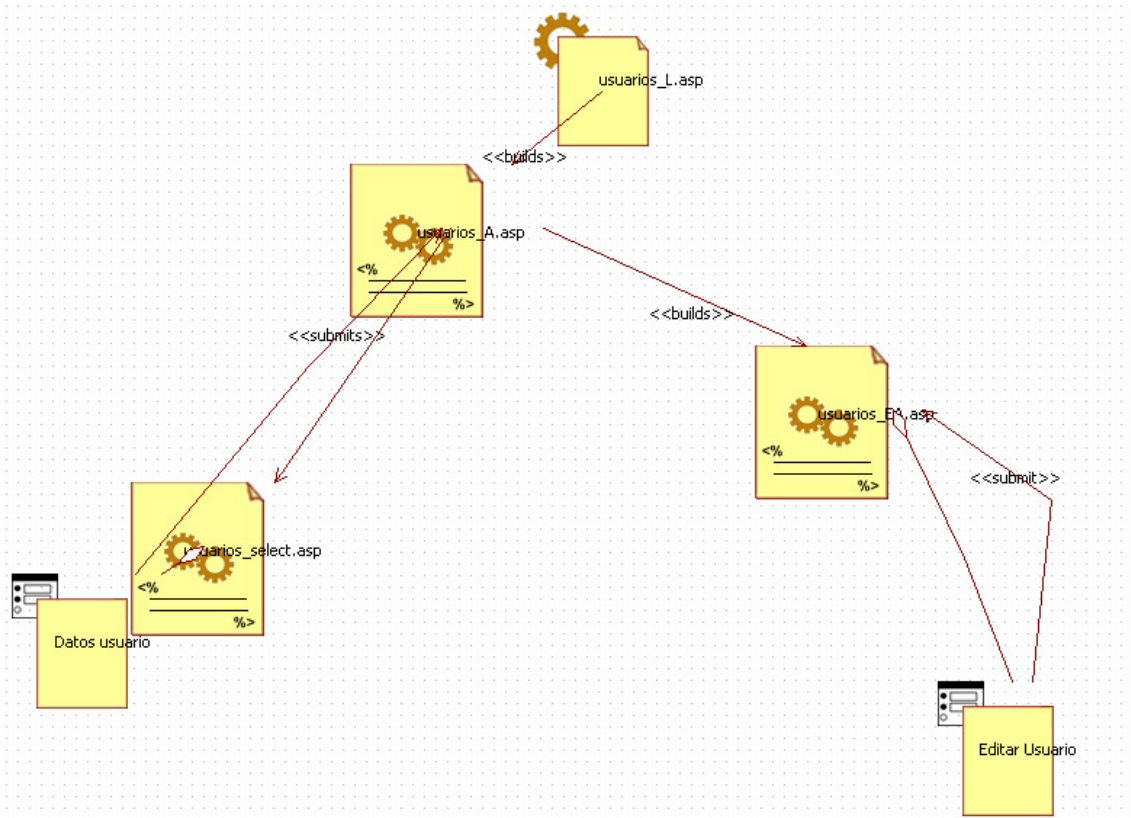
➤ Diagrama WAE para la creación manual de una propuesta de viaje



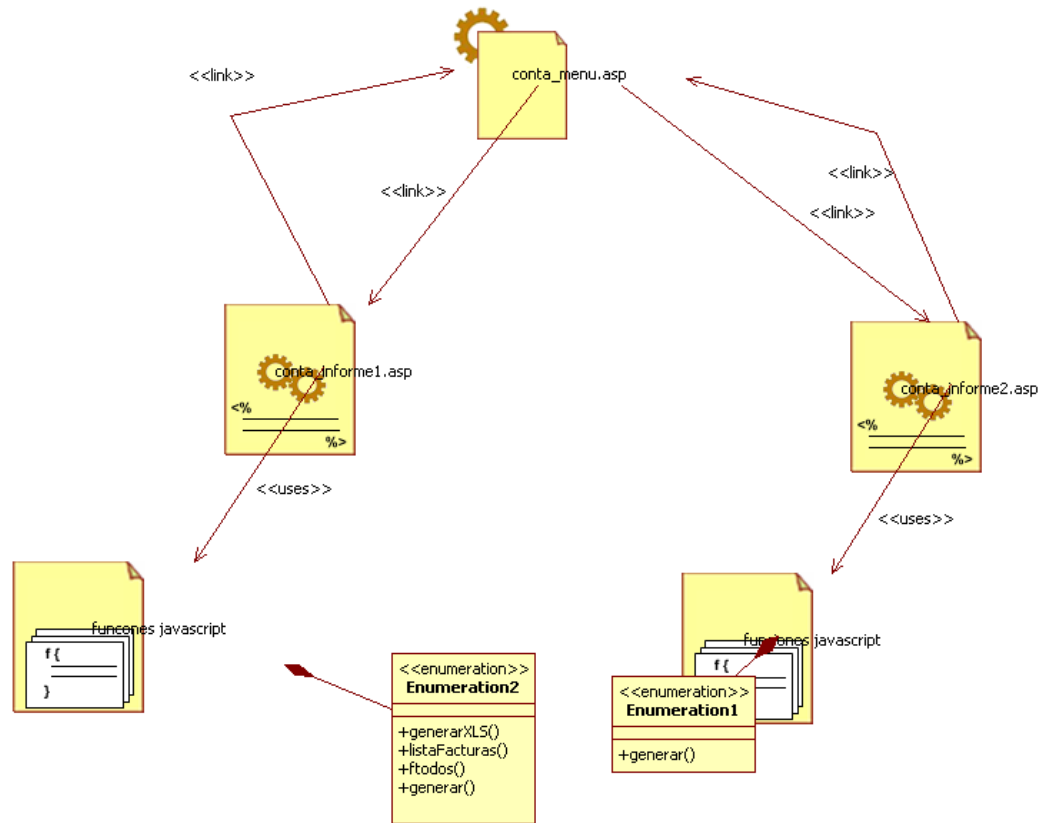
➤ Diagrama WAE para la transformación y edición de propuesta



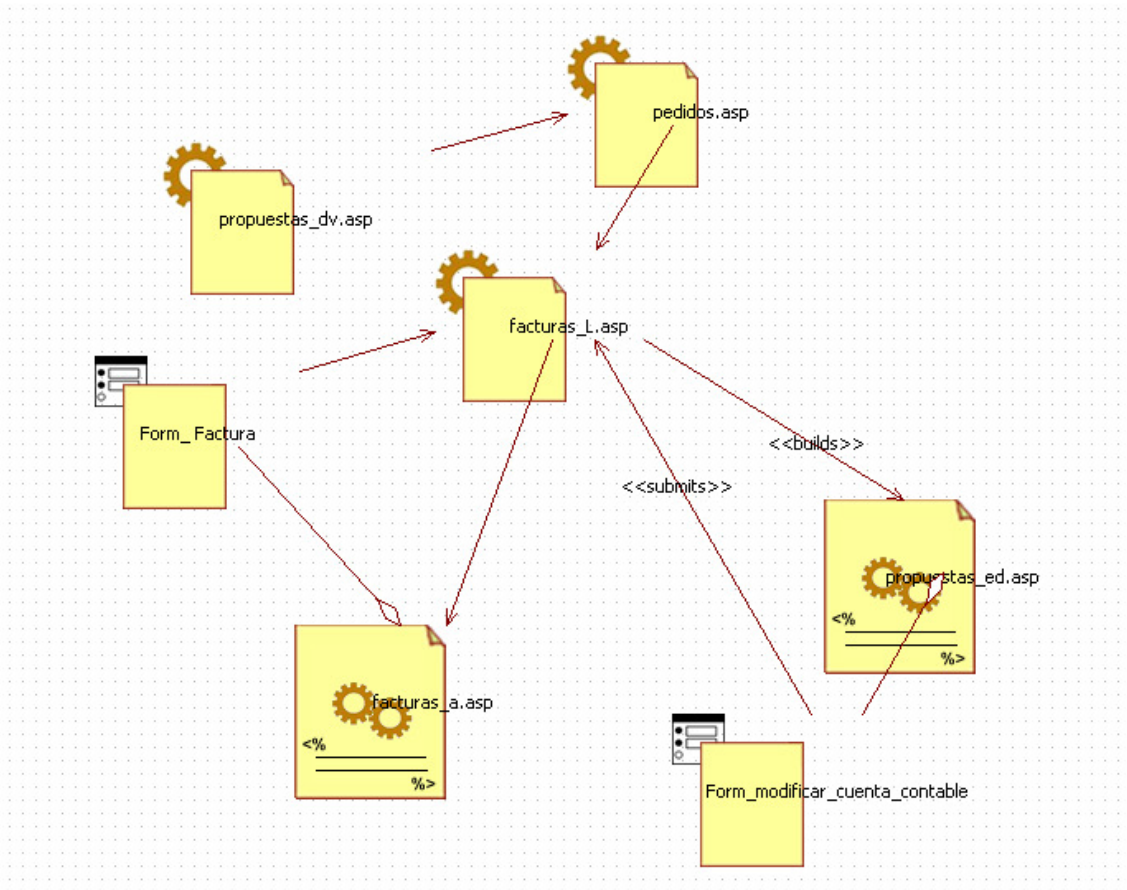
➤ Diagrama WAE para la gestión de usuarios



➤ Diagrama WAE para la gestión de facturas



➤ Diagrama WAE departamento de viajes



3.2 Diseño de la base de datos

3.2.1 Modelo relacional

A continuación mostramos las características principales del diseño de la base de datos con la que interactúa nuestro sistema Web:

3.2.1.1 Relaciones

ARTICULOSDESCRIPCION_CAMPOS

ARTICULOS		DESCRIPCION_CAMPOS
codart	1	∞ codart
Acción actualizar:		Forzado; Actualizaciones en cascada
Tipo de Relación definida:		Uno a varios

ARTICULOSPEDIDOS_DETALLE

ARTICULOS		PEDIDOS_DETALLE
codart	1	∞ codart
Acción actualizar:		Forzado; Actualizaciones en cascada
Tipo de Relación definida:		Uno a varios

ARTICULOSPROPUESTAS_DETALLE

ARTICULOS		PROPUESTAS_DETALLE
codart	1	∞ codart
Acción actualizar:		Forzado; Actualizaciones en cascada
Tipo de Relación definida:		Uno a varios

CENTROSCENTROS_CUENTAS

CENTROS		CENTROS_CUENTAS
codcen	1	∞ codcen
Acción actualizar:		Forzado; Actualizaciones en cascada
Tipo de Relación definida:		Uno a varios

CENTROSCENTROS_FACTURACION

CENTROS		CENTROS_FACTURACION
codcen	1	∞ codcen
Acción actualizar:		Forzado; Actualizaciones en cascada
Tipo de Relación definida:		Uno a varios

CENTROSPROPUESTAS_CENTROS

CENTROS		PROPUESTAS_CENTROS
codcen	1	∞ codcen
Acción actualizar:	Forzado; Actualizaciones en cascada	
Tipo de Relación definida:	Uno a varios	

CENTROSUSUARIOS

CENTROS		USUARIOS
codcen	1	∞ codcen
Acción actualizar:	Forzado; Actualizaciones en cascada	
Tipo de Relación definida:	Uno a varios	

CUENTASCENTROS_CUENTAS

CUENTAS		CENTROS_CUENTAS
codcue	1	∞ codcue
Acción actualizar:	Forzado; Actualizaciones en cascada	
Tipo de Relación definida:	Uno a varios	

CUENTASPROPUESTAS

CUENTAS		PROPUESTAS
codcue	1	∞ codcue
Acción actualizar:	Forzado; Actualizaciones en cascada	
Tipo de Relación definida:	Uno a varios	

FACTURASCENTROS_FACTURACION

FACTURAS		CENTROS_FACTURACION
codfac	1	∞ codfac
Acción actualizar:	Forzado; Actualizaciones en cascada	
Tipo de Relación definida:	Uno a varios	

PROPUESTAS_DETALLEPEDIDOS_DETALLE

PROPUESTAS_DETALLE			PEDIDOS_DETALLE
codpdt	1	∞	codpdt

Acción actualizar: Forzado; Actualizaciones en cascada
Tipo de Relación definida: Uno a varios

PROPUESTASPROPUESTAS_CENTROS

PROPUESTAS			PROPUESTAS_CENTROS
codpro	1	∞	codpro

Acción actualizar: Forzado; Actualizaciones en cascada
Tipo de Relación definida: Uno a varios

PROPUESTASPROPUESTAS_DETALLE

PROPUESTAS			PROPUESTAS_DETALLE
codpro	1	∞	codpro

Acción actualizar: Forzado; Actualizaciones en cascada
Tipo de Relación definida: Uno a varios

PROPUESTASPEDIDOS

PROPUESTAS			PEDIDOS
codpro	1	∞	codpro

Acción actualizar: Forzado; Actualizaciones en cascada
Tipo de Relación definida: Uno a varios

PROVEEDORESPEDIDOS

PROVEEDORES			PEDIDOS
codprov	1	∞	codprov

Acción actualizar: Forzado; Actualizaciones en cascada
Tipo de Relación definida: Uno a varios

USUARIOS_PERMISOSUSUARIOS

USUARIOS_PERMISOS		USUARIOS
codper	1	∞ permisos

Acción actualizar: Forzado; Actualizaciones en cascada
Tipo de Relación definida: Uno a varios

USUARIOSPROPUESTAS

USUARIOS		PROPUESTAS
codusr	1	∞ codusr

Acción actualizar: Forzado; Actualizaciones en cascada
Tipo de Relación definida: Uno a varios

PEDIDOSFACTURAS

PEDIDOS		FACTURAS
codped	1	∞ codped

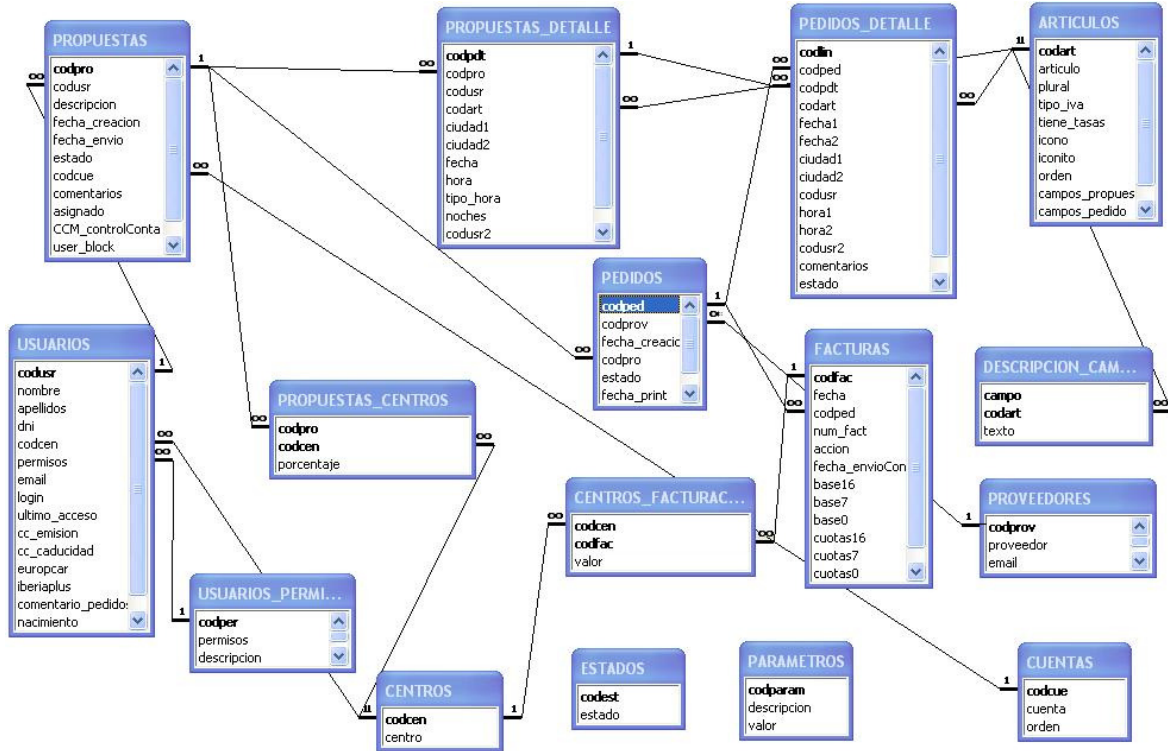
Acción actualizar: Forzado; Actualizaciones en cascada
Tipo de Relación definida: Uno a varios

PEDIDOSPEDIDOS_DETALLE

PEDIDOS		PEDIDOS_DETALLE
codped	1	∞ codped

Acción actualizar: Forzado; Actualizaciones en cascada
Tipo de Relación definida: Uno a varios

3.2.1.2 Mapa relacional



3.2.1.3 Descripción de la base de datos

TABLA PROPUUESTAS

Codpro: CP

Codusr: CA→usuarios que crea la propuesta

Descripción: descripción de la propuesta. Limitamos a 18 caracteres ya que esta información llega a contabilidad.

Fecha_creación:

Fecha_envío: fecha en que la propuesta es enviada al dpto de viajes

Estado: CA→estados. Estado de la propuesta

Codcue: CA→cuentas. Cuenta contable en la que se incluirá la propuesta.

Comentarios: comentarios de la propuesta. Estos comentarios no llegarán al proveedor.

Asignado: persona del dpto de viajes que tiene asignada la propuesta.

CCM_controlConta: Cuando este valor es TRUE, solo conta podrá cambiar los centros de coste o la cuenta.

User_block: usuario que está viendo actualmente la propuesta. Se usa para avisar al resto de usuarios quién la está viendo.

Prioridad: tipo de prioridad de la propuesta: horario, precio

TABLA CENTROS

Codcen: CP

Centro: nombre del centro

TABLA CENTROS

Codpro: CP + CA→propuestas

Codcen: CP + CA→centros

Porcentaje: porcentaje del total de la propuesta que se afectará al centro de coste.

TABLA CENTROS

Codcue: CP

Cuenta: nombre de la cuenta

Orden: campo numérico para ordenar las cuentas en los listados

TABLA CENTROS

Codpdt: CP

Codpro: CA→propuestas

Codusr: CA→usuarios: usuario que viaja

Codart: CA→articulos: articulo de la línea de la propuesta

Ciudad1: ciudad origen

Ciudad2: ciudad destino

Fecha: fecha del viaje

Hora: hora del viaje

Tipo_hora: hora salida u hora llegada

Noches: nº noches

Codusr: segundo usuario de la línea. Segunda cama, segundo conductor...

Comentario: comentario de la línea

TABLA CENTROS

Codart: CP

Articulo: nombre del articulo

Plural: nombre del articulo en plural

Tipo_iva: tipo de iva del artículo (en desuso)

Tiene_tasas: indica si tiene o no tiene tasas (en desuso)

Icono: imagen del icono normal

Iconito: imagen del icono mini

Orden: se usa para ordenar las líneas del pedido

Campos_propuesta: según el articulo se mostrarán unos campos u otros.

Campos_pedido: según el articulo se mostrarán unos campos u otros.

Tipo: transporte, alojamiento, otros

TABLA CENTROS

Según el articulo, los campos de la propuesta o pedidos tienen un nombre u otro.

Campo: CP: nombre del campo

Codart: CP + CA→articulos

Texto: texto a mostrar

TABLA CENTROS

Codped: CP

Codprov: CA→proveedores

Fecha_creacion

Codpro: CA→propuestas

Estado: CA→estados

Fecha_print: fecha en que el pedido es enviado al proveedor (vía mail, antes se imprimía)

Coment_pedido: comentario del pedido para la agencia

TABLA CENTROS

Codprov: CP

Proveedor: nombre del proveedor

Email: email del proveedor al que se enviarán los pedidos de forma automática

Telefono:

TABLA CENTROS

Codfac: CP

Fecha: fecha de la factura

Codped: CA→pedidos

Num_fact: nº de documento

Accion: acción formativa

Fecha_envioConta: fecha de creación de la factura

Base16: base imponible al 16%

Base7: base imponible al 7%

Base0: base imponible al 0%

Cuotas16: importe de las cuotas al 16%

Cuotas7: importe de las cuotas al 7%

Cuotas0: importe de las cuotas al 0%

Fecha_conciliada: si no está conciliada su valor es null. Cuando se concilia se actualiza con la fecha y hora.

TABLA CENTROS

Se crea un registro por cada factura y cada centro de coste que tenga la propuesta.

Codcen: CP + CA→centros

Codfac: CP + CA→facturas

Valor: importe a cargar al centro de coste

TABLA CENTROS

Aclaración: estados de las propuestas y los pedidos

Codest: CP

Estado: nombre del estado

TABLA CENTROS

Codusr: CP

Nombre_ nombre del usuario

Apellidos: apellidos del usuario

Dni: es el campo que se usa para evitar que hayan usuarios duplicados

Codcen: centro de coste por defecto

Permisos: CA→usuarios_permisos: tipo de usuario

Email: email del usuario para el envío de pedidos

Login: login para identificar el acceso del usuario a la aplicación

Ultimo_acceso: fecha del ultimo acceso del usuario

Cc_emision

Cc_caducidad

Europcar

Iberiaplus:

Comentario_pedidos: comentario que irá por defecto en los pedidos del usuario

Nacimiento: fecha nacimiento

Codusj: CA→usuarios : superior jerárquico

TABLA CENTROS

Codper: CP

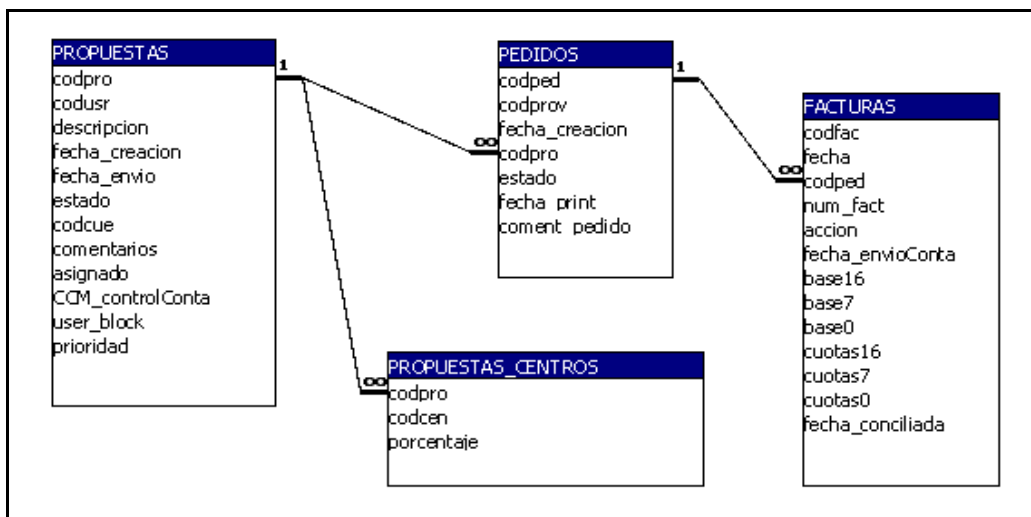
Permisos: tipo de usuario

Descripción: detalle de los sitios donde tiene acceso el usuario

Ayuda: link al fichero de ayuda para el tipo de usuario

3.2.1.3 Peculiaridades de la BD

Como podemos ver en el esquema a continuación, es la propuesta la que lleva asociados los centros de coste. Por tanto si queremos que cada pedido tenga un centro de coste distinto, tendremos que crear una propuesta distinta.



3.2.2 Consultas elaboradas en la base de datos

La herramienta se apoya en la base de datos para almacenar la información necesaria y obtener en cada momento los datos necesarios. Pero las tablas originales tienen limitaciones para obtener ciertas informaciones. Es por ello que en el diseño de la base de datos hemos elaborado una serie de consulta con el apoyo del gestor de la base de datos.

No vamos a detallar aquí el detalle de estas consultas, lo dejamos para la sección Apéndice C de la memoria.

Capítulo 4

Utilización del sistema

En el siguiente apartado detallaremos al completo el manual de usuario para poder dar una referencia o guía a los usuarios habituales del sistema. El sistema como es sabido, se trata de una aplicación Web, cualquier usuario familiarizado con un explorador Web no tendrá muchos problemas de adaptación.

4.1 Introducción al manual de usuario

La aplicación como bien es sabido, se trata de un sistema Web, el usuario para poder acceder a ella simplemente deberá ejecutar el explorador Web y a partir de ahí de buscar en el menú de aplicaciones del portal interno de la empresa la aplicación *Gevi*.

4.1.1 La Pantalla inicial. Login.

Es la pantalla que se encuentra el usuario al acceder a la aplicación. En esta pantalla cada usuario debe de entrar con su cuenta y su password correspondiente. Si el usuario detecta cualquier anomalía al intentar acceder al sistema, debe de ponerse en contacto con el administrador para que verifique su alta.

Si tras pulsar el botón enviar, la contraseña o login introducidos sean erróneos el usuario visualizará un mensaje de error en la pantalla.

Usuario o password incorrecta



The image shows a login form with a dark blue header containing the word "Login". Below the header, there are two input fields: "Usuario:" and "Password:". Below the "Password:" field is a button labeled "Enviar". Above the form, the text "Usuario o password incorrecta" is displayed, indicating an error message.

4.1.2 Pantalla principal

Tras acceder al sistema podemos ver la pantalla principal de la aplicación. Dependiendo nuestro rol de usuario, veremos ciertos componentes del menú principal. En la figura vemos el caso de un usuario administrador del sistema, que tiene acceso a todos los componentes del menú.



En la figura podemos observar que en la parte superior vemos el menú de secciones de la aplicación. Las secciones serán visibles dependiendo del tipo de usuario que seamos.

Vemos como aparece nuestro login arriba junto con el rol que tenemos. En la parte inferior, vemos la lista de propuestas del usuario que accede al sistema.

Tras presentar el sistema y ver como podemos acceder a él, procederemos a presentar un manual por tipo de usuario. Es decir un manual para un usuario de centros con la funcionalidad que le pertenece (creación de propuestas para recepción), otro para usuarios de central (creación de propuestas y pedidos), usuario recepcionista (gestión de pedidos, propuestas y facturación) y por último usuario contable (gestión de informes contables y conciliación de facturas).

4.2 Manual de usuario de central

Procedemos a presentar un detallado manual para el usuario de central, esto es, los usuarios con acceso a la aplicación que trabajan en servicios centrales.

Al entrar en la aplicación vemos el listado de los viajes que hemos creado. Si es la primera vez que entramos la lista estará vacía. Lo primero que tenemos que comprobar es que nuestros datos estén bien informados. Para eso pinchamos en el botón “MIS DATOS”. Llegaremos a la siguiente pantalla:

The screenshot shows a web application interface for user management. At the top, there is a blue navigation bar with a briefcase icon, the text "MENU LFORMENT (USUARIO CENTRAL)", and three buttons: "MIS VIAJES", "MIS DATOS", and "AYUDA". On the right side of the bar, it says "PROVEEDOR POR DEFECTO". Below the navigation bar, the main content area is titled "DATOS DE USUARIO" and contains a form with the following fields:

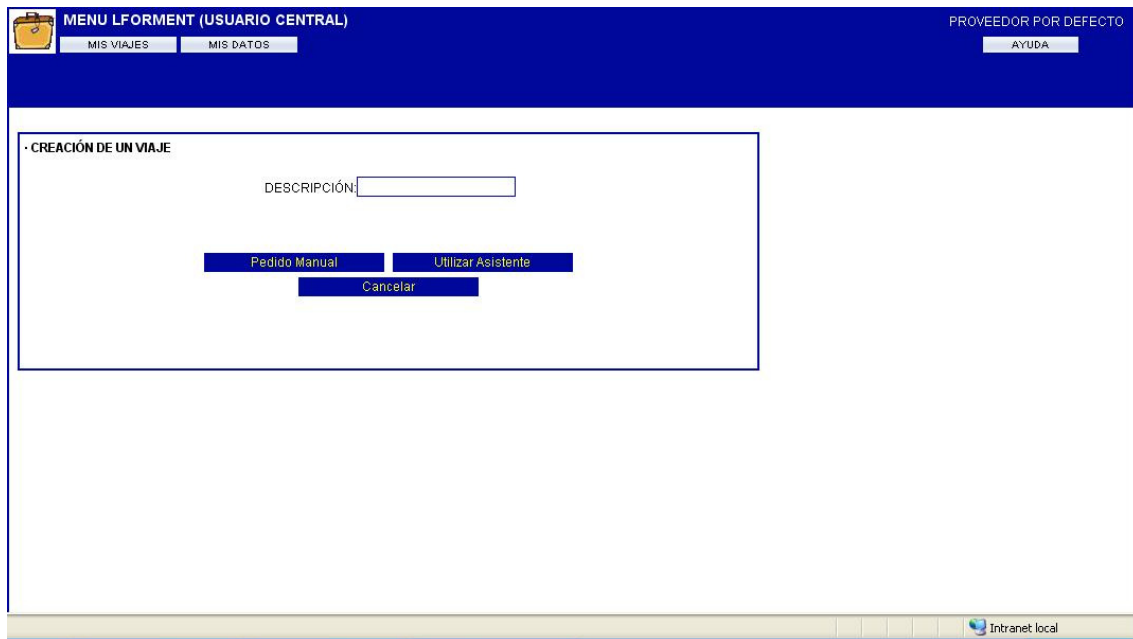
USUARIO:	LFORMENT	
NOMBRE:	<input type="text" value="LOLES"/>	
APELLIDOS:	<input type="text" value="FORMENT"/>	
DNI:	<input type="text" value="52632756Q"/>	
E-MAIL:	<input type="text" value="lforment@norauto.es"/>	
TJTA EUROPCAR:	<input type="text"/>	
TJTA IBERIA PLUS:	<input type="text"/>	
PERMISO DE CONDUCIR:	EMISIÓN: <input type="text"/>	CADUCIDAD: <input type="text"/>
CENTRO:	<input type="text" value="Recursos"/>	
TIPO USUARIO:	<input type="text" value="Usuario Central"/>	
COMENTARIO PEDIDOS POR DEFECTO:	<input type="text" value="pruebas..."/>	
SUPERIOR JERÁRQUICO:	<input type="text" value="JMFERRER"/>	

A "Guardar" button is located at the bottom right of the form. At the bottom of the page, there is a status bar with the text "Intranet local" and a small globe icon.

No todos los datos son obligatorios. Los importantes son el nombre y los apellidos, el e-mail y el centro.

4.2.1 Como crear un viaje

Al pinchar en “MIS VIAJES” volvemos al listado anteriormente mencionado y tenemos el botón NUEVO que nos llevará al formulario de creación de un viaje.



Como se puede ver tenemos la opción de crearlo manualmente o con el asistente.

El asistente es muy útil para viajes típicos de ida y vuelta con hotel para una o varias personas pero en ocasiones puede ser más rápido crearlo manualmente.

En todas las fases de creación podemos pulsar cancelar para cancelar la propuesta de viaje o siguiente para validar la fase y avanzar en el proceso. Una vez finalizado el asistente, la propuesta queda guardada en el sistema, será decisión nuestra si queremos crear un pedido del viaje o no.

El asistente consta de varios pasos. En el primero introducimos los datos básicos del viaje que podemos ver a continuación:

MENU LFORMENT (USUARIO CENTRAL) PROVEEDOR POR DEFECTO

MIS VIAJES MIS DATOS AYUDA

· PASO 1: DATOS DEL VIAJE

DESCRIPCIÓN:

ORIGEN:

DESTINO:

SALIDA:

REGRESO:


Nº PASAJEROS:

TIPO VIAJE: IDA Y VUELTA SOLO IDA

DETALLE DE LOS PASOS DEL ASISTENTE:

1. DATOS DEL VIAJE
2. DETALLE DE LOS TRAYECTOS
3. ALOJAMIENTOS
4. NOMBRES PASAJEROS

Intranet local

En el segundo paso tenemos que indicar los medios de transporte, las horas deseadas de salida y regreso y podemos insertar puntos intermedios con el icono 

MENU LFORMENT (USUARIO CENTRAL) PROVEEDOR POR DEFECTO

MIS VIAJES MIS DATOS AYUDA

· PASO 2: DETALLE DE LOS TRAYECTOS

DATOS DEL VIAJE

DESCRIPCIÓN: manualcentral

ORIGEN: Valencia SALIDA: 15/12/2010

DESTINO: Lille REGRESO: 17/12/2010

TIPO VIAJE: IDA Y VUELTA Nº PASAJEROS: 1

TRAYECTO Nº1

TRANSPORTE: ORIGEN:  DESTINO: FECHA: HORA:

TRAYECTO Nº2

TRANSPORTE: ORIGEN:  DESTINO: FECHA: HORA:

COMENTARIOS INTERNOS:

NOTAS:

Si seleccionas H.SALIDA (hora de salida) estás diciendo a partir de qué hora quieres salir.
 Si seleccionas H.LLEGADA (hora de llegada) estás diciendo la hora máxima a la cual quieres llegar.

Intranet local


El asistente propone automáticamente un hotel en función de la fechas de salida y regreso. Podemos eliminarlo pero no añadir otro. Para añadir otro hotel debemos finalizar el asistente y añadirlo posteriormente.


MENU LFORMENT (USUARIO CENTRAL) PROVEEDOR POR DEFECTO


MIS VIAJES MIS DATOS AYUDA

- PASO 3: DETALLE DEL ALOJAMIENTO

DESCRIPCIÓN: manualcentral
 ORIGEN: Valencia SALIDA: 15/12/2010
 DESTINO: Lille REGRESO: 17/12/2010
 TIPO VIAJE: IDA Y VUELTA N° PASAJEROS: 1

 **TRAYECTO N°1: AVIÓN** Valencia - Lille
 15/12/2010 H.LLEGADA: 20:00

 **HOTEL** Lille 15/12/2010 N°Noches: 2 X

 **TRAYECTO N°2: AVIÓN** Lille - Valencia
 17/12/2010 H.SALIDA: 20:00

COMENTARIOS INTERNOS:

[Atras](#) [Siguiete](#) [Cancelar](#)

NOTA: Para crear otra noche de Hotel hay que finalizar el asistente y añadirla posteriormente.

Intranet local

El último paso del asistente es la elección de los pasajeros. El pasajero nº1 es por defecto la persona que crea el viaje. Deberemos de añadir el resto de pasajeros y finalizar el asistente.

MENU LFORMENT (USUARIO CENTRAL) PROVEEDOR POR DEFECTO

MIS VIAJES MIS DATOS AYUDA

ORIGEN: Valencia SALIDA: 15/12/2010
 DESTINO: Lille REGRESO: 17/12/2010
 TIPO VIAJE: IDA Y VUELTA N° PASAJEROS: 1

 **TRAYECTO N°1: AVIÓN**
 Valencia - Lille
 15/12/2010 H.LLEGADA: 20:00

 **HOTEL Lille**
 15/12/2010 N°Noches: 2

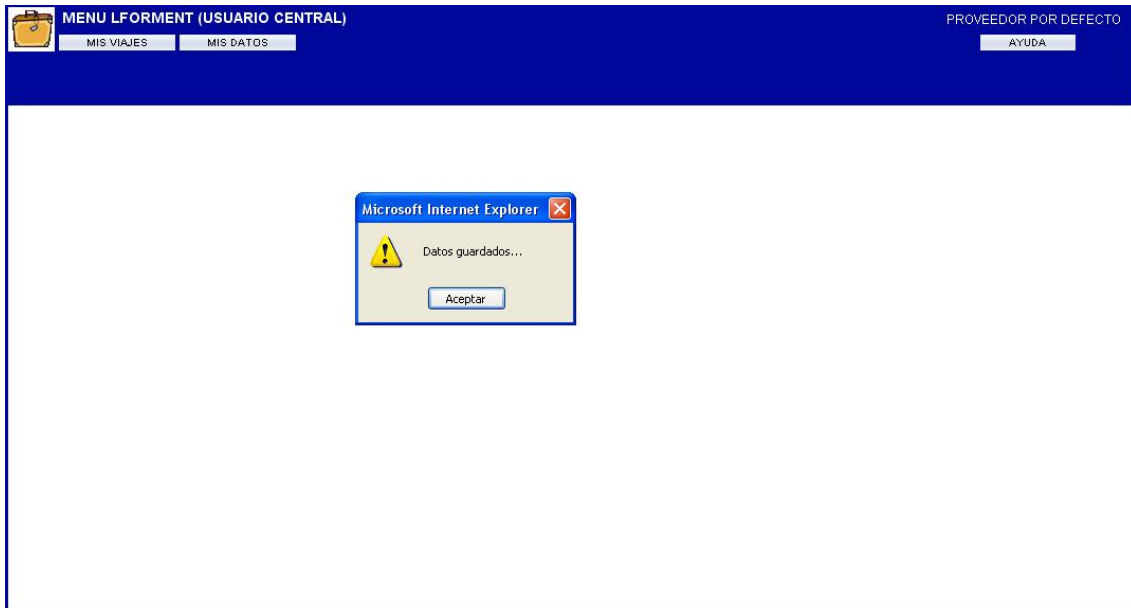
 **TRAYECTO N°2: AVIÓN**
 Lille - Valencia
 17/12/2010 H.SALIDA: 20:00

NOMBRES DE LOS PASAJEROS:

COMENTARIOS INTERNOS:

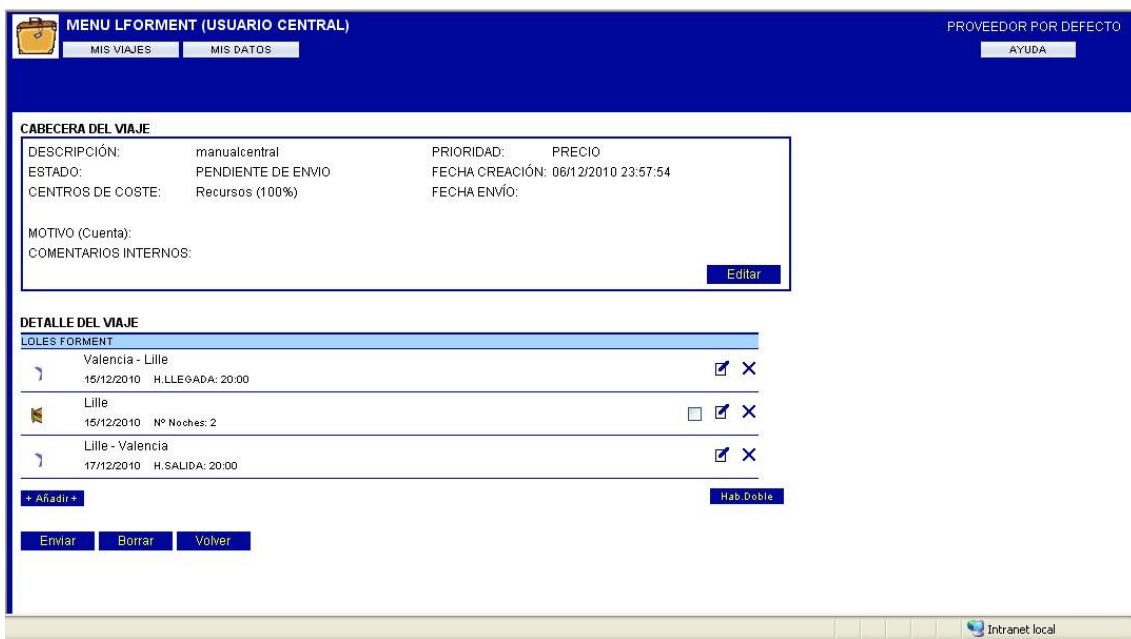
[Atras](#) [Siguiete](#) [Cancelar](#)



Hemos podido ver en todas las pantallas de asistente un campo "comentarios internos" que sirve para introducir cualquier comentario de forma interna (no llegará a la agencia de viajes). Cuando finaliza el proceso simplemente vemos como se han guardado los datos de la propuesta y deberemos pulsar en aceptar para ver el detalle de la propuesta de nuestro viaje.

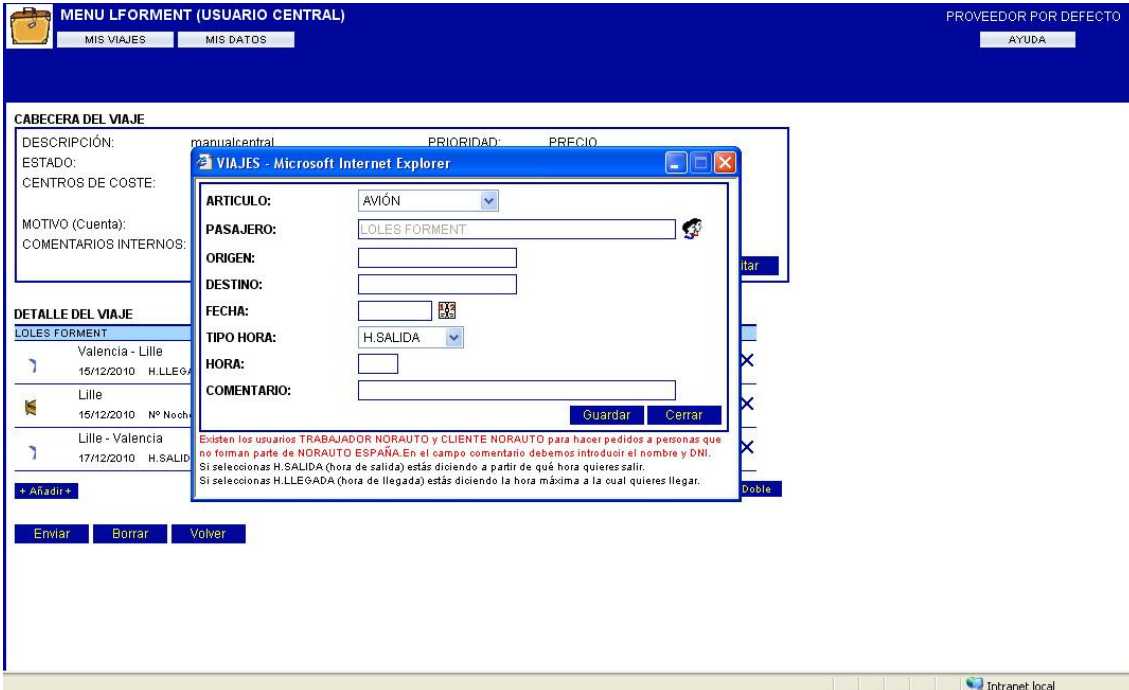


El asistente continúa y finaliza en la pantalla de introducción manual de líneas de detalle.




Si al principio de la creación del viaje hubiésemos decidido crear el viaje de forma manual, habríamos entrado en esta misma pantalla, pero las líneas que se han creado con el asistente tendríamos que haberlas creado una a una.



Con el botón AÑADIR podemos añadir más líneas que el asistente no nos ha podido crear y con los iconos  y  podemos editar o eliminar la línea.



DESCRIPCIÓN:	manualcentral	PRIORIDAD:	PRECIO:
ESTADO:			
CENTROS DE COSTE:			
MOTIVO (Cuenta):			
COMENTARIOS INTERNOS:			

DETALLE DEL VIAJE
LOLES FORMENT
 Valencia - Lille 15/12/2010 H.LLEGADA
 Lille 15/12/2010 Nº Noche
 Lille - Valencia 17/12/2010 H.SALIDA

ARTICULO:	AVIÓN
PASAJERO:	LOLES FORMENT
ORIGEN:	
DESTINO:	
FECHA:	
TIPO HORA:	H.SALIDA
HORA:	
COMENTARIO:	

Existen los usuarios TRABAJADOR NORAUTO y CLIENTE NORAUTO para hacer pedidos a personas que no forman parte de NORAUTO ESPAÑA. En el campo comentario debemos introducir el nombre y DNI.
Si seleccionas H.SALIDA (hora de salida) estás diciendo a partir de qué hora quieres salir.
Si seleccionas H.LLEGADA (hora de llegada) estás diciendo la hora máxima a la cual quieres llegar.

Aquí vemos como nos salta una ventana con un formulario para cumplimentar. Este formulario corresponde a la línea de detalle que queramos añadir o modificar. La fecha la hora y las ciudades, para este caso, serán obligatorios, en todo caso en el sistema, si introducimos algún valor erróneo o nos falta algún dato nos reportará un error a modo informativo.

Existe también un botón Hab.Doble que sirve para juntar a 2 personas en una misma habitación de hotel. Para ello pinchamos en 2 casillas y luego en dicho botón.

Una vez tengamos el detalle creado nos faltará añadir el motivo (cuenta contable) del viaje. Para ello accedemos a la edición de la cabecera del viaje con el botón editar.

Además del motivo podremos editar el centro de coste al que irá imputado el viaje. Sin estas dos características el sistema no nos dejará crear el pedido.

MENU LFORMENT (USUARIO CENTRAL) PROVEEDOR POR DEFECTO

MIS VIAJES MIS DATOS AYUDA

CABECERA DEL VIAJE

DESCRIPCIÓN:	manualcentral	PRIORIDAD:	PRECIO
ESTADO:	PENDIENTE DE ENVIO	FECHA CREACIÓN:	06/12/2010 23:57:54
CENTROS DE COSTE:	Recursos (100%)	FECHA ENVÍO:	

MOTIVO (Cuenta):
COMENTARIOS INTERNOS:

Microsoft Internet Explorer

Debes de seleccionar un motivo y verificar el/los centros de coste.

DETALLE DEL VIAJE

LOLES FORMENT

Valencia - Lille	15/12/2010 H.LLEGADA: 20:00	<input type="checkbox"/>	<input type="button" value="✎"/>	<input type="button" value="✕"/>
Lille	15/12/2010 Nº Noches: 2	<input type="checkbox"/>	<input type="button" value="✎"/>	<input type="button" value="✕"/>
Lille - Valencia	17/12/2010 H.SALIDA: 20:00	<input type="checkbox"/>	<input type="button" value="✎"/>	<input type="button" value="✕"/>

Intranet local

Aquí vemos la pantalla que muestra la cabecera del viaje.

MENU LFORMENT (USUARIO CENTRAL) PROVEEDOR POR DEFECTO

MIS VIAJES MIS DATOS AYUDA

CABECERA DE LA PROPUESTA DE PEDIDO

DESCRIPCIÓN:	<input type="text" value="manualcentral"/>
FECHA CREACIÓN:	06/12/2010 23:57:54
FECHA ENVÍO:	
PRIORIZAR:	<input type="text" value="PRECIO"/>
ESTADO:	TRAMITADO
CENTROS DE COSTE:	Recursos (100%)
[Añadir/Quitar] [Mod. %]	
MOTIVO (Cuenta):	<input type="text" value="Calendario Formación Interna"/>
COMENTARIOS INTERNOS:	<input type="text"/>

Intranet local

Si no seleccionamos el centro de coste asociado al viaje, el sistema nos informará con un error.

MENU LFORMENT (USUARIO CENTRAL) PROVEEDOR POR DEFECTO

MIS VIAJES MIS DATOS AYUDA

CABECERA DEL VIAJE

DESCRIPCIÓN:	manualcentral	PRIORIDAD:	PRECIO
ESTADO:	PENDIENTE DE ENVIO	FECHA CREACIÓN:	06/12/2010 23:57:54
CENTROS DE COSTE:	Recursos (100%)	FECHA ENVÍO:	

MOTIVO (Cuenta):
COMENTARIOS INTERNOS:

Microsoft Internet Explorer

Debes de seleccionar un motivo y verificar el/los centros de coste.

Aceptar

[Editar](#)

DETALLE DEL VIAJE

LOLES FORMENT

Valencia - Lille	15/12/2010	H.LLEGADA: 20:00	✎ ✕
Lille	15/12/2010	Nº Noches: 2	<input type="checkbox"/> ✎ ✕
Lille - Valencia	17/12/2010	H.SALIDA: 20:00	✎ ✕

[+ Añadir](#) [Hab. Doble](#)

[Enviar](#) [Borrar](#) [Volver](#)

Intranet local

Y por ultimo, debemos enviar el viaje al proveedor para que nos lo tramite y nos devuelva los billetes. Para ello utilizamos el botón ENVIAR en la parte inferior de la pantalla del detalle del viaje. Llegaremos a una pantalla como la que se muestra a continuación, donde podremos introducir un comentario al pedido, que esta vez si llegará al proveedor.

MENU LFORMENT (USUARIO CENTRAL) PROVEEDOR POR DEFECTO

MIS VIAJES MIS DATOS AYUDA

CABECERA DEL VIAJE

DESCRIPCIÓN:	manualcentral	PRIORIDAD:	PRECIO
ESTADO:	PENDIENTE DE ENVIO	FECHA CREACIÓN:	06/12/2010 23:57:54
CENTROS DE COSTE:	Recursos (100%)	FECHA ENVÍO:	

MOTIVO (Cuenta): **Calendario Formación Interna**
COMENTARIOS INTERNOS:

DETALLE DEL VIAJE

LOLES FORMENT

Valencia - Lille	15/12/2010	H.LLEGADA: 20:00
Lille	15/12/2010	Nº Noches: 2
Lille - Valencia	17/12/2010	H.SALIDA: 20:00

PROVEEDOR:

COMENTARIO PEDIDO:

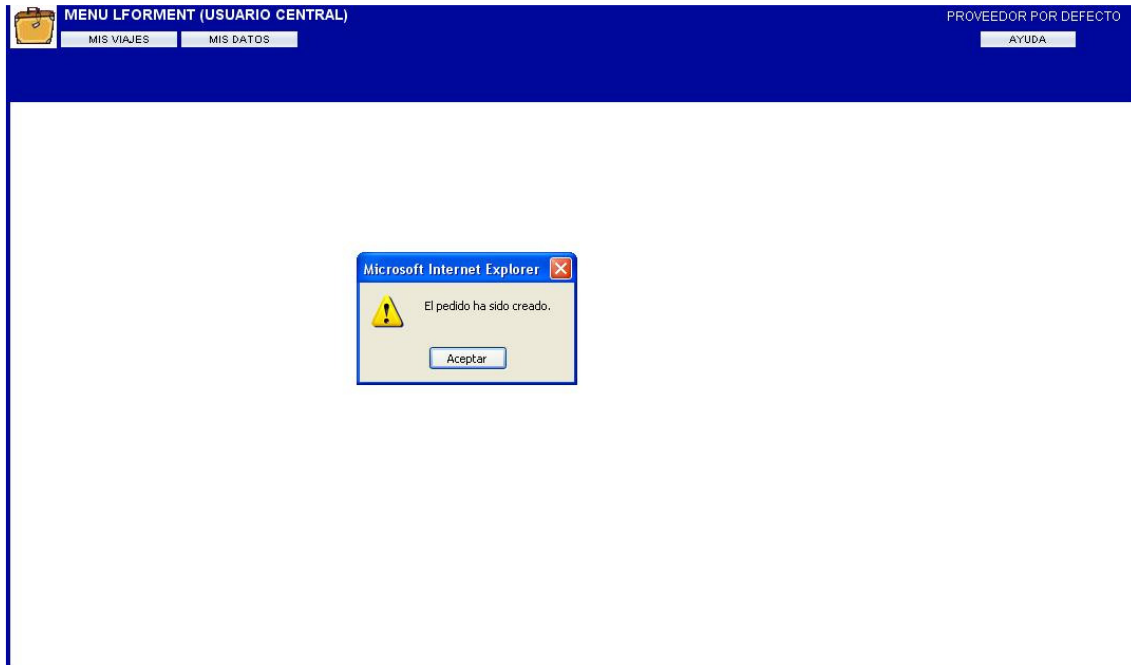
Se enviará un mail al proveedor con el detalle del pedido.

[Enviar](#) [Cancelar](#)

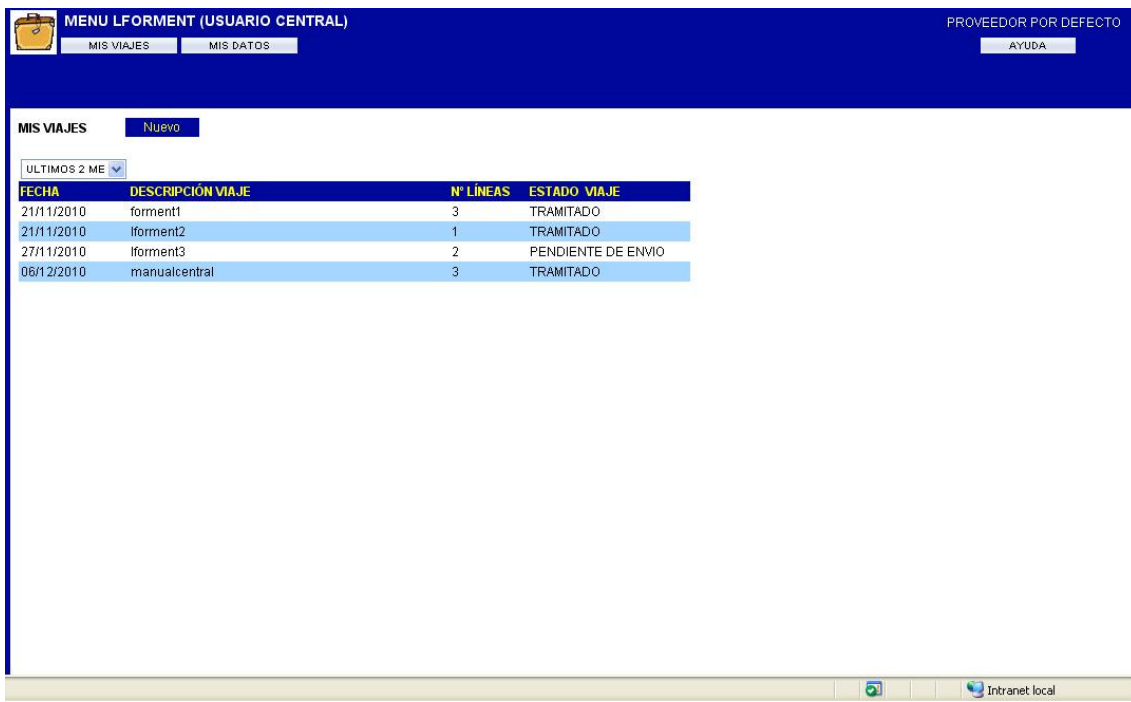
Intranet local

Es importante una vez llegados a este punto, saber que una vez enviado el pedido queremos informar algún cambio, podremos añadir o editar líneas del pedido actual o incluso

crear un nuevo pedido. Como ya sabemos una propuestas puede estar formada por muchos pedidos. También podremos volver a enviar un pedido ya enviado con el botón ENVIAR, nuevamente le llegará al proveedor nuestro pedido.



En este punto el pedido pasará a estar tramitado.



A partir del momento en que se envía el pedido, el proveedor nos enviará por e-mail una lista de posibilidades de viaje. Debemos contestarle la que nos interese pero todo esto se hará vía e-mail o teléfono.

4.3 Detalle usuario centros

Para el caso del usuario con rol centros, los pasos de creación del viaje son exactamente los mismos que para el usuario central. Lo único que difiere es en el paso de creación de pedido.

Tras seleccionar los centros de coste asociados y el motivo de cuenta si procedemos a darle al botón enviar, este pedido no llegará al proveedor, sino que llegará a la recepción de servicios centrales como una propuesta a tratar. Además el sistema le dejará un correo electrónico a recepción como modo de aviso, del usuario que ha propuesto el viaje.

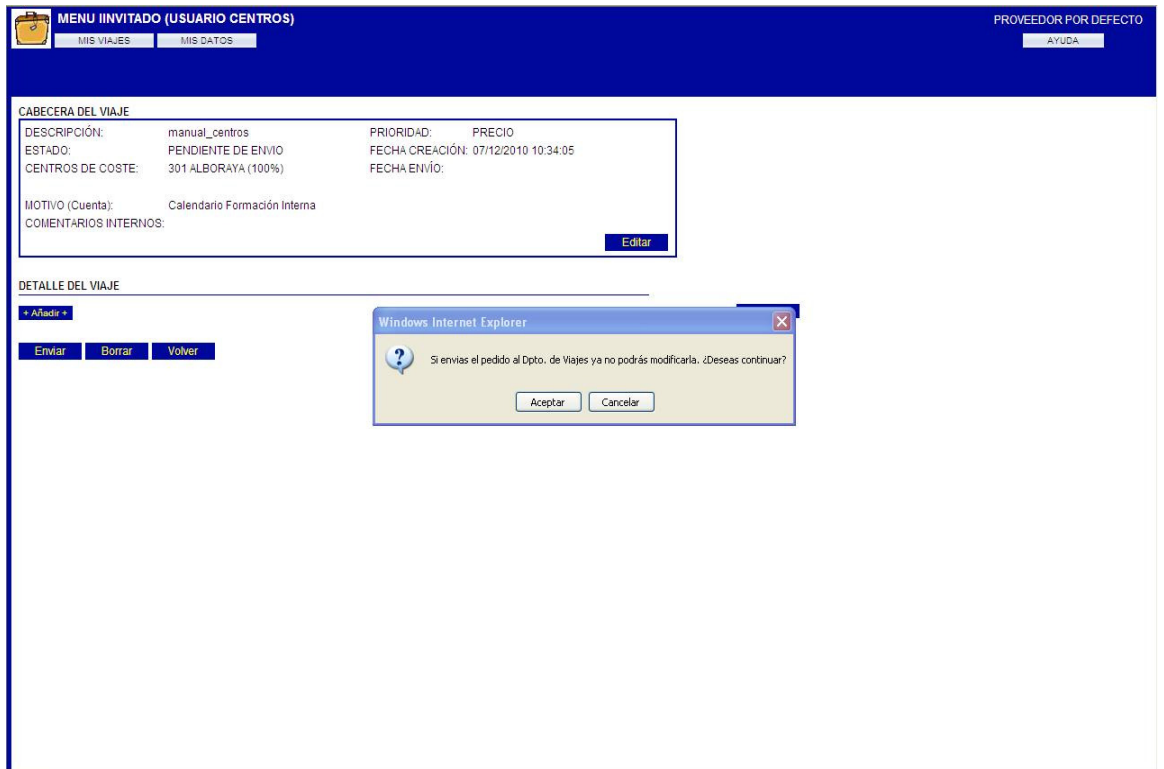
PROPUESTA DE VIAJE

Jose-Manuel FERRER

Para: Jose-Manuel FERRER

Te he hecho un nuevo pedido de viaje. Gracias.

Vemos en la imagen lo que ocurre cuando intentamos enviar el pedido.

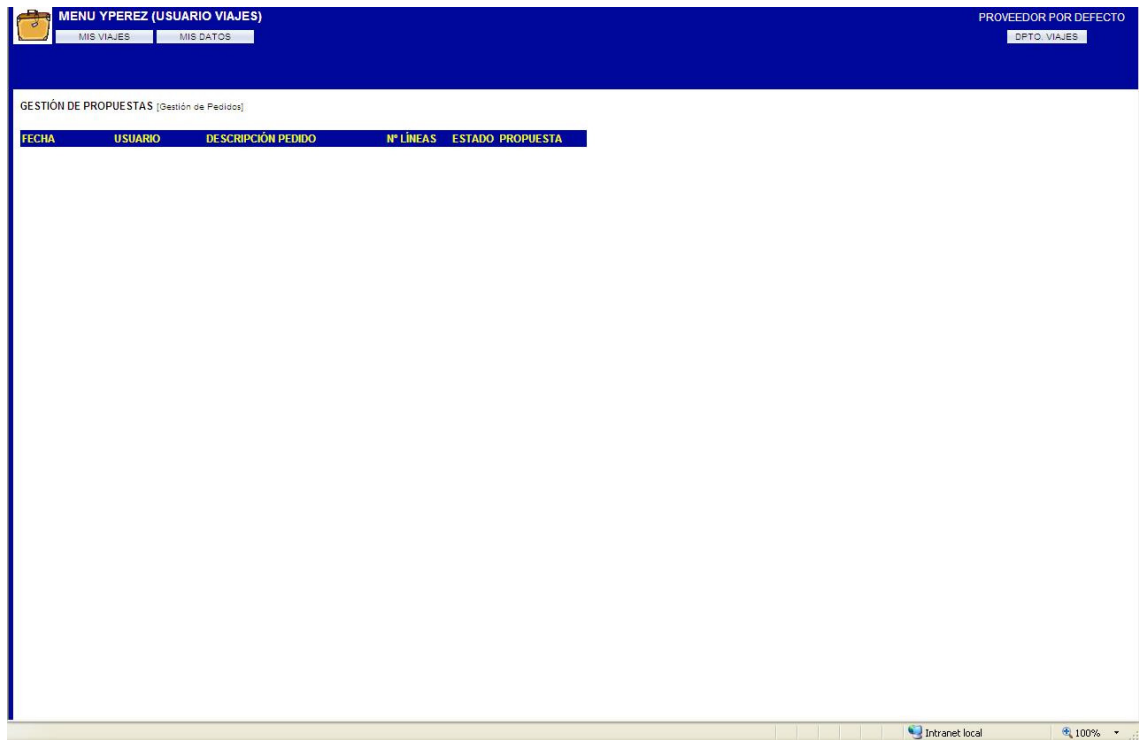


El departamento de viajes tramitará con la agencia y nos devolverá los billetes. Para ello utilizamos el botón ENVIAR en la parte inferior de la pantalla del detalle del viaje.

4.4 Manual usuario viajes. Gestión de los pedidos.

En este caso vamos a describir las tareas que deben seguir los usuarios del departamento de viajes a través de la herramienta claro está.

Quando un usuario del departamento de viajes accede a la aplicación observa la siguiente vista:



El usuario de viajes tiene dos tareas a hacer a través de la aplicación.

4.5 Revisión de propuestas de centros y creación de pedidos

El usuario de viajes cuando accede a la aplicación tiene dos vistas posibles: Gestión de propuestas, y gestión de pedidos.

En la vista de propuestas aparece una lista de todas las propuestas pasadas por los usuarios de centros y que están pendientes de tramitar. Para tramitar una propuesta, los pasos son muy sencillos. De entre la lista de propuestas pendientes de trámite de la vista propuestas, pueden seleccionar una de ellas pinchando en el enlace de la columna “Descripción de pedido”.

MENU YPEREZ (USUARIO VIAJES) PROVEEDOR POR DEFECTO

MIS VIAJES MIS DATOS DPTO. VIAJES

GESTIÓN DE PROPUESTAS [Gestión de Pedidos]

FECHA	USUARIO	DESCRIPCIÓN PEDIDO	Nº LÍNEAS	ESTADO PROPUESTA
07/12/2010	INVITADO	manual_viajes	2	ENVIADO A DEP.VIAJES

Intranet local 100%

Después de pinchar sobre una propuesta en concreto, el departamento de viajes deberá de seleccionar las líneas de propuesta, que pasarán al proveedor como pedido y pincharán en crear pedido.

MENU YPEREZ (USUARIO VIAJES) PROVEEDOR POR DEFECTO

MIS VIAJES MIS DATOS DPTO. VIAJES

PROPUESTA DE PEDIDO Asignar

DESCRIPCIÓN:	manual_viajes	PRIORIDAD:	PRECIO
USUARIO:	INVITADO INVITADO (301)	ESTADO:	ENVIADO A DEP.VIAJES
MOTIVO (Cuenta):	Calendario Formación Interna	FECHA CREACIÓN:	07/12/2010 12:54:41
CENTROS DE COSTE:	301 ALBORAYA (100%)	FECHA ENVÍO:	07/12/2010 13:26:38

[Editar](#)

[Ver Leyenda](#)

PROPUESTA ORIGINAL ▶

LÍNEAS DE LA PROPUESTA SIN TRATAR CREAR PEDIDO

INVITADO INVITADO

- valencia 22/12/2010
- valencia 22/12/2010

[Finalizar](#) [Volver](#)

Intranet local 100%

Finalmente en pantalla aparece un apartado donde el usuario de viajes deberá seleccionar el proveedor destino del pedido y un comentario interno que le llegará al mismo.

MENU YPEREZ (USUARIO VIAJES) MIS VIAJES MIS DATOS PROVEEDOR POR DEFECTO DPTO. VIAJES

CABECERA DEL VIAJE

DESCRIPCIÓN:	manual_viajes	PRIORIDAD:	PRECIO
ESTADO:	ENVIADO A DEP. VIAJES	FECHA CREACIÓN:	07/12/2010 12:54:41
CENTROS DE COSTE:	301 ALBORAYA (100%)	FECHA ENVÍO:	07/12/2010 13:26:38

MOTIVO (Cuenta): Calendario Formación Interna
COMENTARIOS INTERNOS:

DETALLE DEL VIAJE

INVITADO	INVITADO
valencia	22/12/2010
valencia	22/12/2010

PROVEEDOR: CARLSON
COMENTARIO PEDIDO: pruebas...
Se enviará un mail al proveedor con el detalle del pedido.

Enviar Cancelar

4.6 Recepción de pedidos.

Una vez tramitado el pedido, el proveedor responde con un correo automático llamado “bono” informativo como que se ha realizado el trámite. En el transcurso de pocas horas el proveedor envía a Recepción el albarán del pedido o factura electrónica. Con este albarán el usuario de viajes (repcionista) deberá crear una factura en el sistema Gevi que luego conciliará el usuario contable. Para ello, a través de la vista Gestión de pedidos, el usuario de viajes selecciona el pedido a recepcionar.

MENU YPEREZ (USUARIO VIAJES) PROVEEDOR POR DEFECTO
 MIS VIAJES MIS DATOS DPTO. VIAJES

GESTIÓN DE PEDIDOS [Gestión de Propuestas]

X T VP V

FECHA	USUARIO	Nº PEDIDO	PROVEEDOR	DESCRIPCIÓN PROPUESTA	Nº LÍNEAS	ESTADO PEDIDO	
01/12/2010	INVITADO	2	CARLSON	prueba2	1	TRAMITADO	
18/12/2010	INVITADO	1	CARLSON	prueba	1	TRAMITADO	

TOTAL LISTA: 2

NOTA: Los pedidos que se crearon hace más de 30 días y siguen sin estar visados aparecen en rojo.

Intranet local

Simplemente pinchando en el botoncito en forma de camión accede a la pantalla de recepción del pedido. Una vez ahí dentro le aparece un pequeño detalle del pedido con las líneas pendientes de recepción.

MENU YPEREZ (USUARIO VIAJES) PROVEEDOR POR DEFECTO
 MIS VIAJES MIS DATOS DPTO. VIAJES

RECEPCIÓN DEL PEDIDO

CARLSON PEDIDO: 2 [\[Editar CC + Cuenta\]](#)

LÍNEAS DEL PEDIDO SIN RECEPCIONAR

burjassot 01/12/2010
 INVITADO INVITADO (24242424L)

[Volver Listado](#) [Volver Pedido](#)

Intranet local

El usuario viajes deberá de recepcionar las líneas que le vengan en el albarán y pinchar en la flechita azul que aparece a la derecha “crear nuevo documento”. Tras esto aparece un formulario de creación de factura. La vemos a continuación:

MENU YPEREZ (USUARIO VIAJES) PROVEEDOR POR DEFECTO

MIS VIAJES MIS DATOS DPTO. VIAJES

RECEPCIÓN DE UN PEDIDO
CARLSON PEDIDO: 1

FECHA DOCUMENTO:
 Nº DOCUMENTO: Valencia 18/12/2010
 ACCIÓN: INVITADO INVITADO (24242424L)

Base 18%: 0 Cuotas 18%: 0
 Base 8%: 0 Cuotas 8%: 0
 Base 0%: 0 Cuotas 0%: 0
 TOTAL Base: 0 TOTAL Cuotas: 0
 TOTAL Documento: 0

Esta tarea es un poco más delicada, pues deberán crear la factura que formará a pasar parte del sistema y que contabilidad usará para integrar los gastos. Entonces, con albarán en mano, el usuario deberá introducir la fecha del día a la que está recepcionado el pedido, el número de documento, será el número de albarán (num_expte del albarán) y luego deberá indicar el importe en la sección correspondiente de IVA que corresponda, esta información también viene reflejada al final del albarán. El campo acción solo será un código numérico que se indicará en caso de formar parte de un viaje de formación interna de la empresa. Cada viaje de formación tendrá su identificación numérica. Así mismo deberá seleccionar el pedido que aparece a la derecha.

Tras esto simplemente deberá pinchar en el botón guardar se refrescará la pantalla con el detalle de la factura creada.

MENU YPEREZ (USUARIO VIAJES) PROVEEDOR POR DEFECTO

MIS VIAJES MIS DATOS DPTO. VIAJES

RECEPCIÓN DEL PEDIDO
CARLSON PEDIDO: 1

Nº DOCUMENTO: 2 **FECHA: 06/12/2010** **FECHA RECEPCIÓN: 08/12/2010 9:58:22**

valencia 18/12/2010
 INVITADO INVITADO (24242424L)

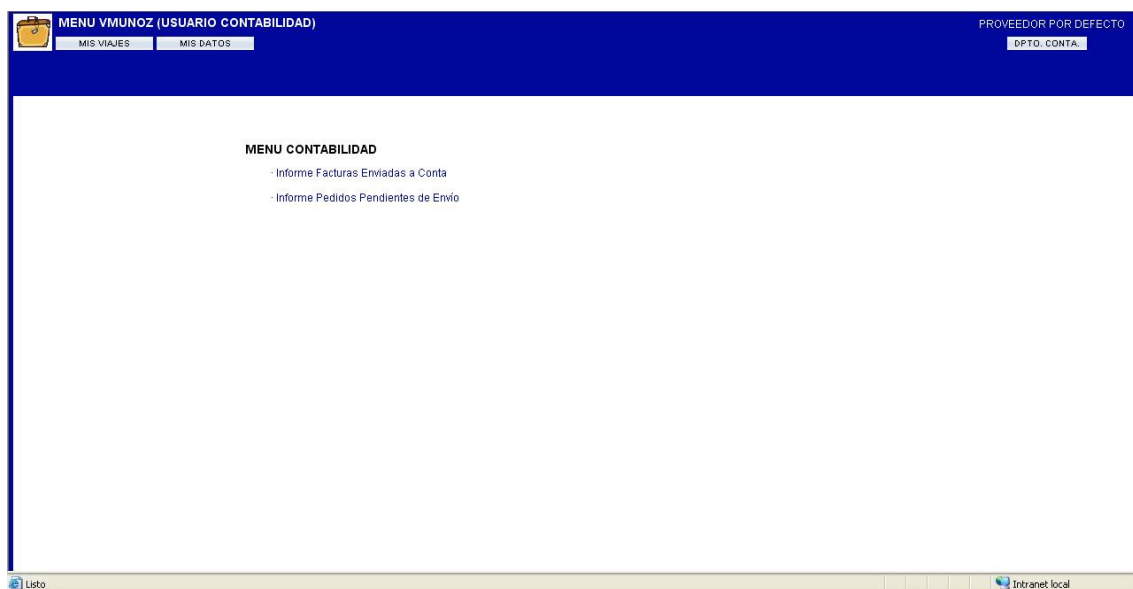
BASE 18%: 0,00 €	CUOTAS 18%: 0,00 €	
BASE 8%: 100,00 €	CUOTAS 8%: 8,00 €	
BASE 0%: 0,00 €	CUOTAS 0%: 0,00 €	
TOTAL BASE: 100,00 €	TOTAL CUOTAS: 8,00 €	TOTAL FACTURA: 108,00 €

LÍNEAS DEL PEDIDO SIN RECEPCIONAR

Si quedan más líneas de pedido por recepcionar, podrá crear una nueva factura o volver a los listados de propuestas para hacer alguna tarea más.

4.7 Manual usuario contabilidad.

Aquí explicaremos brevemente lo que un usuario de contabilidad necesita para sus tareas en la aplicación. Cuando un usuario contable accede a la aplicación y pincha en el botón del menú principal “DPTO. CONTA” tiene dos vistas posibles.



La primera vista sirve para acceder al histórico de facturas creadas por los usuarios de viajes. La segunda sirve para ver el histórico de pedidos tramitados pendientes de crear factura. Por tanto si el usuario contable quiere conciliar una factura simplemente deberá acceder a la primera vista.

MENU VMUNOZ (USUARIO CONTABILIDAD) PROVEEDOR POR DEFECTO
DPTO. CONTA

MIS VIAJES MIS DATOS

INFORME FACTURAS ENVIADAS A CONTA

FECHA INICIO: FECHA FIN: Facturas sin conciliar

PROVEEDOR: TODOS

NOTA: Si dejas una fecha en blanco no se tendrá en cuenta.
Leyenda: Con fondo gris, ya conciliadas.

<input checked="" type="checkbox"/>	Cuenta	Centro	PED.	Descripción	Importe	N'Doc	Proveedor	Acción	Fecha Doc	Fecha Conc
TOTAL LÍNEAS: 0										

Listo Intranet local

En esta pantalla básicamente lo que se puede hacer es generar listados, hay un filtro en la parte superior de fácil uso. Pueden filtrar por fecha, proveedor y si están conciliadas o no. Para conciliar una factura simplemente deberá filtrar las facturas no conciliadas entre la fecha que quieran y marcar la casillas de las facturas que quieran conciliar. Si el usuario contable necesita extraer un listado a una hoja de datos, simplemente deberá pinchar en el botón “generar XLS” una vez generado el listado.

MENU VMUNOZ (USUARIO CONTABILIDAD) PROVEEDOR POR DEFECTO
DPTO. CONTA

MIS VIAJES MIS DATOS

INFORME FACTURAS ENVIADAS A CONTA

FECHA INICIO: FECHA FIN: Facturas sin conciliar

PROVEEDOR: TODOS

NOTA: Si dejas una fecha en blanco no se tendrá en cuenta.
Leyenda: Con fondo gris, ya conciliadas.

<input checked="" type="checkbox"/>	Cuenta	Centro	PED.	Descripción	Importe	N'Doc	Proveedor	Acción	Fecha Doc	Fecha Conc
<input checked="" type="checkbox"/>	629003	301	2	INVITADO 011210 prueba2	126,00	1	F05034		06/12/2010	
<input type="checkbox"/>	629003	301	1	INVITADO 181210 prueba	100,00	2	F05034		06/12/2010	

TOTAL LÍNEAS: 2

Listo Intranet local

Por otra parte, en la otra vista posible de la sección DEPT.CONTA. los usuarios contables pueden pasar a hoja de datos un listado de todas los pedidos pendientes de crear

factura, de esta forma ejercen un control sobre la carga de trabajo. Al igual que en la conciliación simplemente deberán pinchar en el botón "generarXLS".

Capítulo 5

Conclusiones

5.1 Elección del proyecto

Desde el punto de vista de la elección de este proyecto he de indicar las razones por las que he seleccionado este tipo de proyecto y la importancia del mismo en mi vida actual profesionalmente hablando.

Desde el punto de vista de los distintos tipos de software que se puede elegir para elaboración de un proyecto, una de las razones que me ha motivado a elegir este en concreto es la característica Web. Soy un alumno que ha cursado la especialización de “Tecnologías y servicios para Web” luego era lógico desde el punto de vista de mis conocimientos y como seguimiento lógico en mi tramo final de la carrera, elegir un proyecto des estas características.

Por otro lado el estar integrado dentro de la compañía ha facilitado que me involucrara rápidamente en la participación de proyectos internos de la compañía. La compañía me ha ofrecido su apoyo para mis estudios y ha depositado la confianza en mí para el desarrollo de esta importante herramienta.

5.2 Futuro y valoración del producto

El proyecto realizado abarca todos los aspectos del producto que necesitaba la compañía para mejorar la gestión administrativa de los viajes de forma informatizada.

Si hablamos de la importancia del proyecto cabe destacar la amplitud de usuarios que van a abarcar la utilización de la herramienta en un futuro. Hablamos de una herramienta que van a utilizar rutinariamente cientos de usuarios de una empresa importante a nivel nacional.

Cabe señalar que uno de mis objetivos prioritarios en la fase inicial del desarrollo era elaborar una herramienta lo más simple y directa posible, puesto que tenía que ser una herramienta destinada a muchos y distintos tipos de usuario. La sencillez de una herramienta software como aprendí en Ingeniería de software es un rasgo de calidad destacable.

Durante el desarrollo del producto he recibido mucho apoyo en el departamento de informática del que formo actualmente parte, aportándome conocimientos e ideas importantes para el desarrollo de la herramienta.

Este proyecto me ha servido a su vez como experiencia para afrontar futuros retos dentro de la compañía y poder seguir evolucionando profesionalmente.

5.3 Posibles mejoras y modificaciones

El producto está recientemente terminado y elaborado. De momento cabe esperar una temporada para ver como funciona en producción y observar sus posibles fallos o disfunciones que darían lugar a una posible puesta en marcha de mejora y modificación.

Por el momento lo que si está a la vista es un cambio ágil del sistema. Se trata de modificar la forma de acceder, para que sea más ágil y profesional.

Actualmente el sistema está diseñado con una pantalla inicial típica de login para poder acceder a la aplicación. Puesto que estamos dentro de una red interna y cada usuario tiene su propio login y perfil usuario en la red, hemos pensado que la aplicación en el proceso de entrada al sistema consulte mediante el servidor, el login del sistema que utiliza y lo consulte en su base de datos de usuarios de gevi. Simplemente el login del sistema y de gevi tienen que coincidir y ello evitaría tener que recordar contraseñas. Si un usuario no está registrado en Gevi cuando intente acceder al sistema, simplemente el sistema mostrará un error advirtiéndole que no tiene los accesos necesarios.

Se trata de una modificación fácil y ágil que no llevará mucho tiempo ni romperá ninguna estructura actual.

Capítulo 6

Referencias y bibliografía

6.1 Referencias en la red

1. Página Web Wikipedia. <http://www.wikipedia.org/>
2. Foros de Internet. <http://www.forosdelweb.com/>
3. <http://www.conocimientosweb.net> . Para diagramas.
4. <http://www.humbertocervantes.net/cursos/tutoriales> . Para tutoriales de UML.
5. <http://www.dcc.uchile.cl/~psalinas/uml/ejemplo/ejemplo.html>
6. Guíos de programación asp. <http://www.desarrolloweb.com/manuales>
7. Productos software: staruml.softonic.com/descargar
8. Editores de código y gráficos. <http://msdn30.e-academy.com>

6.2 Bibliografía

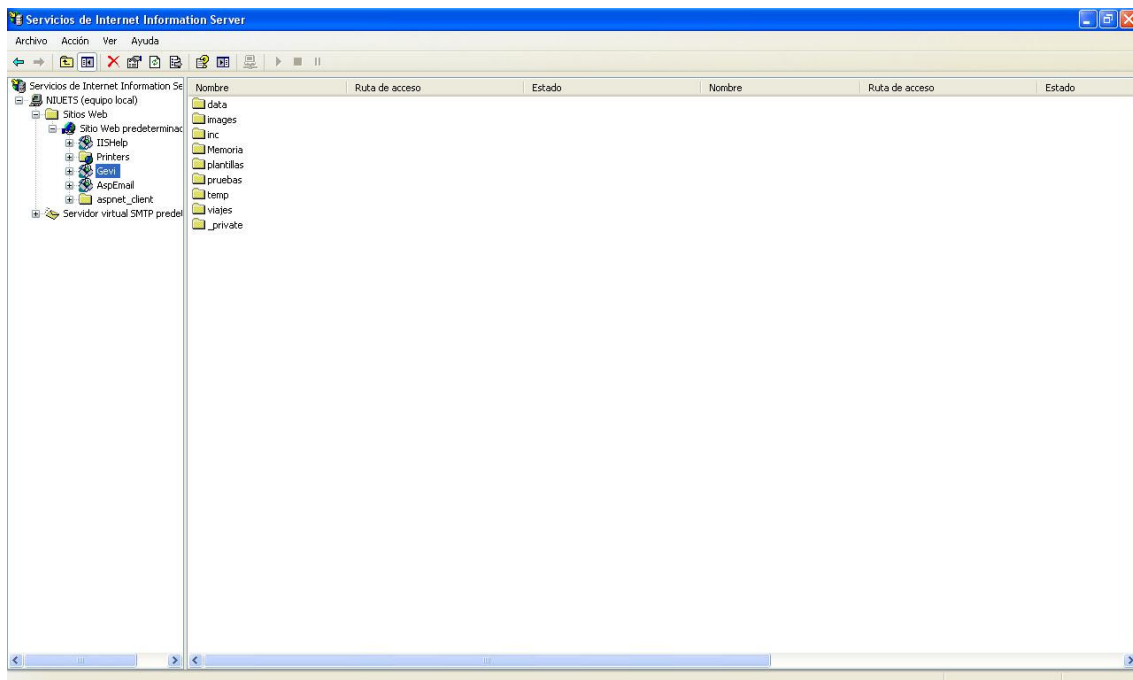
1. *Building web applications with UML*, Second Edition, Jim Conallen. Biblioteca de informática
2. Manual. *Desarrollo de una aplicación Web*.
3. *Consejos para el desarrollo de tu proyecto final de carrera*. Francisco José Abad García.
4. *Modelado de aplicaciones en Internet*. Álvaro Rendón Gallón

Apéndice A

Estructura de directorios del sistema

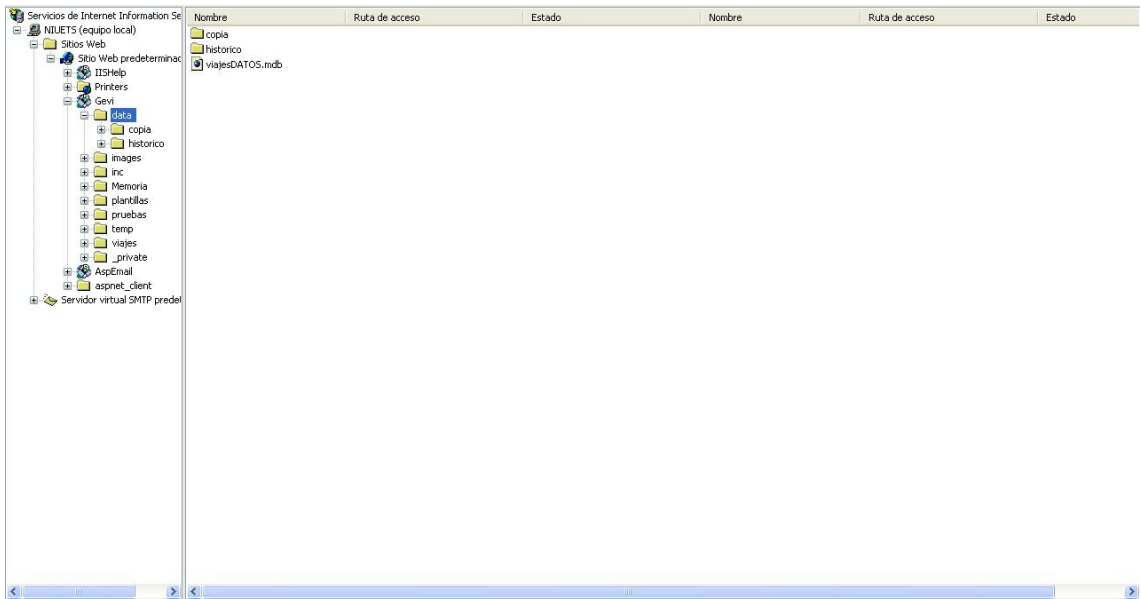
En este apéndice vamos a presentar la estructura de directorios del sitioweb del sistema, así como una breve descripción del contenido de cada uno de ellos.

En la siguiente figura observamos la perspectiva del sitio desde el punto de vista del servidor del servicio IIS.

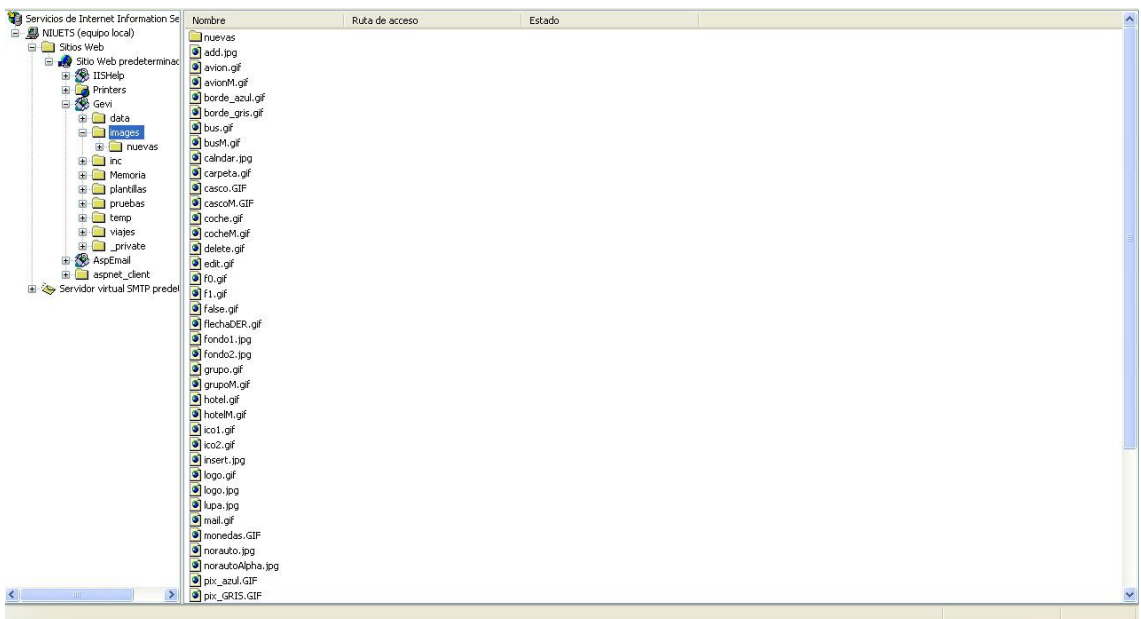


A.1 Descripción del directorio

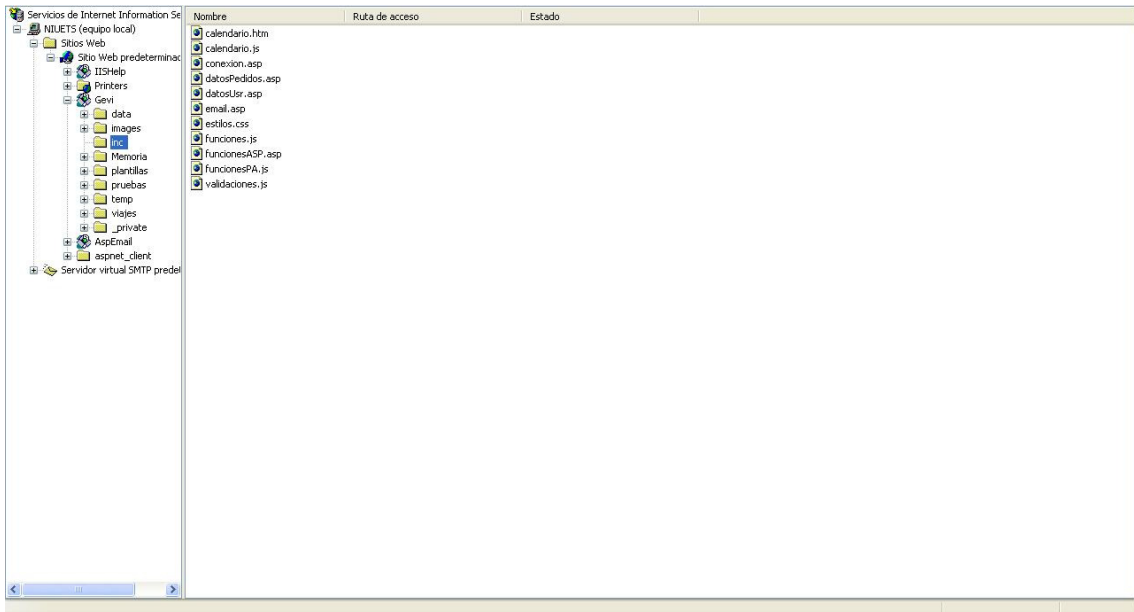
- **Directorio Data:** En este directorio encontramos la base de datos con la que interactúa el sistema así como una copia de seguridad de la misma. También encontramos los archivos de imagen de la presente memoria.



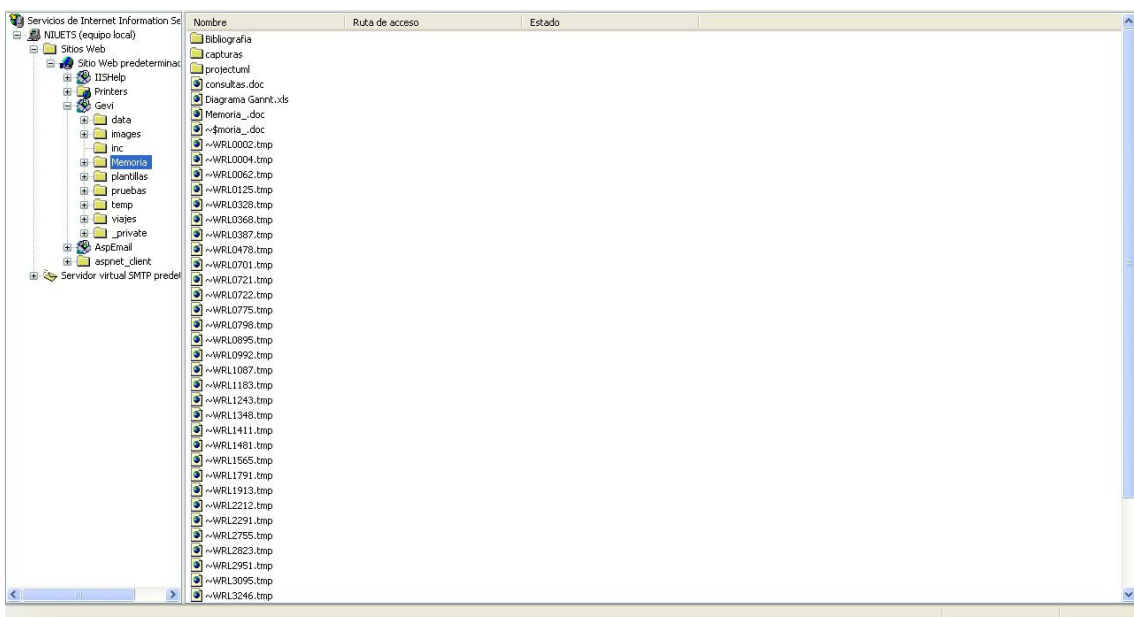
- **Directorio images:** En este directorio tenemos almacenadas todas las imágenes que tiene integradas la aplicación, en iconos, fondos, imágenes etc...



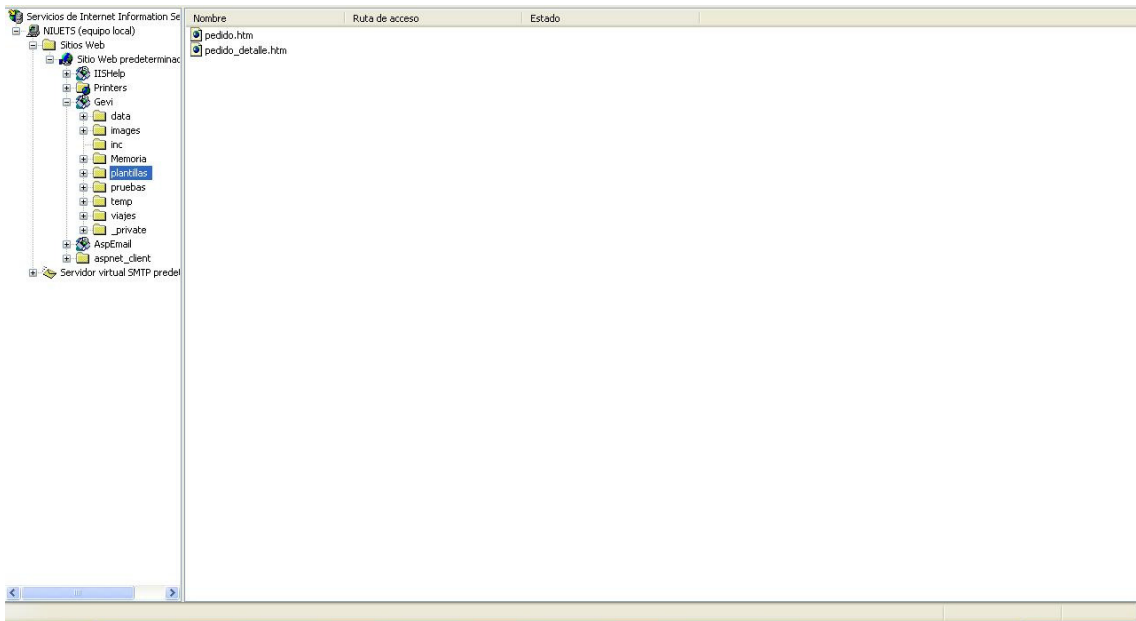
- **Directorio Inc.:** Este es el directorio donde tenemos almacenados las librerías del sistema, esto es, todos aquellos archivos fuente con código asp y java script, que el sistema utiliza como librerías en sus páginas.



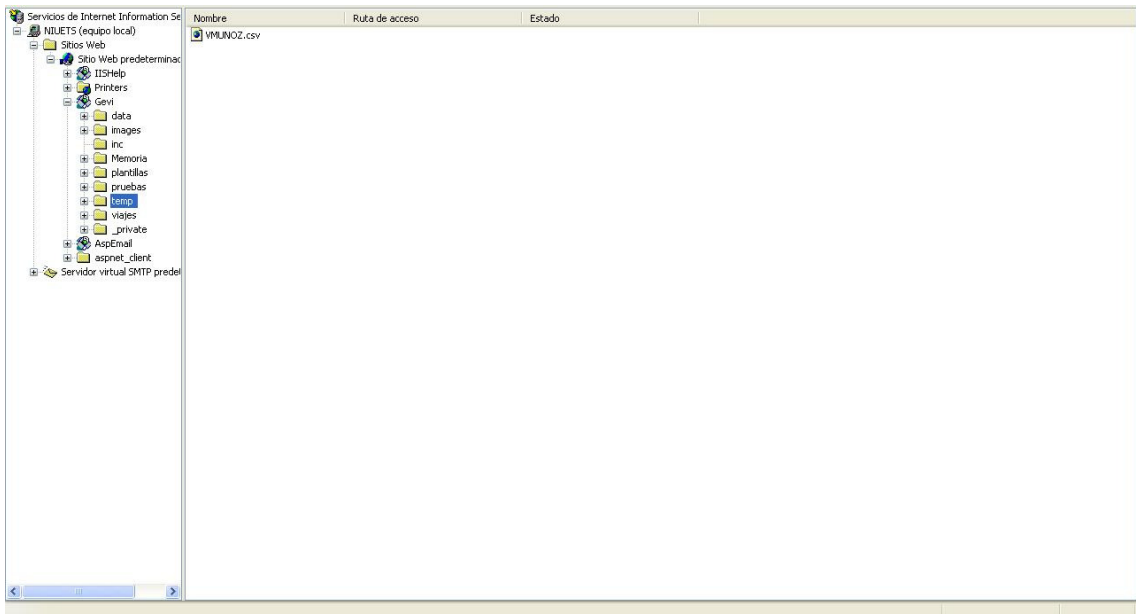
- **Directorio Memoria:** En este directorio tenemos almacenado los archivos de documentación del proyecto. La memoria, diagramas, informes de la base de datos, bibliografía, capturas...



- **Directorio plantillas:** En este directorio tenemos dos simples archivos. Estos archivos son las plantillas que utiliza la función de enviar correo para enviar los correos electrónicos a los proveedores.



- **Directorio Temp:** Este directorio es donde se almacena un histórico de los listados en forma de hojas de datos, generados por contabilidad.



- **Directorio viajes:** Por último presentamos el directorio viajes, el más importante si cabe, ya que alberga todos las páginas y el grueso de código del que está constituido el sistema.

Nombre	Ruta de acceso	Estado
copias		
Prueba		
borde.htm		
conta_cc.asp		
conta_cc_ec.asp		
conta_cc_eq.asp		
conta_inF.asp		
conta_informe1.asp		
conta_informe1.C.asp		
conta_informe1_070126.asp		
conta_informe2.asp		
conta_menu.asp		
datos.asp		
default.asp		
entrada_viajes.asp		
envioMail.asp		
error.asp		
excel.asp		
Facturas_A.asp		
Facturas_A1_old.asp		
Facturas_A2.asp		
Facturas_E.asp		
Facturas_L.asp		
Facturas_1060930.asp		
login.asp		
menu.asp		
menu2.asp		
pedidos.asp		
pedidos_C.asp		
pedidos_dv.asp		
pedidos_dv_a.asp		
pedidos_dv_a2.asp		
pedidos_dv_ag.asp		
pedidos_dv_ag.asp		
pedidos_dv_e.asp		
pedidos_dv_I.asp		
pedidos_dv_leyenda.asp		
pedidos_I.asp		
pedidos_N.asp		

Apéndice B

Librerías. Funciones a destacar

En este Apéndice vamos a describir un poco las librerías que hemos presentado en el Apéndice A de la memoria. Se trata de dar una visión global de la función que cumplen las librerías y destacar algunas de las funciones importantes del sistema y como interactúan con la base de datos. No vamos a incluir demasiado código aquí, puesto que todo está disponible en el directorio del sitioweb. Simplemente en esta sección incluiremos algún trozo de código interesante o alguna descripción de flujo interesante.

Las librerías principales a destacar del directorio *inc* son: FuncionesASP.asp, calendario, conexión.asp, datosPedidos.asp, datosUsr.asp, estilos, email.asp

Del calendario destacar que tenemos un archivo **calendario.js** que contiene código JavaScript mediante el cuál dibujamos un calendario. Este calendario será utilizado para realizar filtros de fechas, elección de fechas de vuelos, de hotel, etc. Hemos pensado que implementar un calendario dinámico era útil y aportaba complejidad y profesionalidad al sistema.

Este script trabaja junto con el documento .htm para dibujar el calendario.

```
//SCRIPT Calendario
```

```
function dev_titulo()
{
    var mes = document.form1.mes.value;
    var dia = document.form1.dia.value;
    var anyo =document.form1.anyo.value;

    return dev_nombreMes(mes) + " " + anyo;
}
/////////////////////////////////////////////////////////////////
/////////////////////////////////////////////////////////////////
function codigoCalendario()
{
```

```

var mes = document.form1.mes.value;
var dia = document.form1.dia.value;
var anyo =document.form1.anyo.value;

var calendario = "";

var dia1 = new Date(anyo,mes-1,1);
var diasDelMes=dev_diasMes(mes,anyo);
var primerDiaMes = dia1.getDay();
if (primerDiaMes == 0) primerDiaMes = 7;

calendario += "<table border=0 cellpadding=2 cellspacing=0>";

calendario += "<tr class='cTit'>";
calendario += "<td colspan=7>";
calendario += "<input class=boton type='button' name='b1' onclick='cambioMes(-1,0)'
value='<<'>&nbsp;";
calendario += "<input class=boton type='button' name='b2' onclick='cambioMes(0,-1)'
value='<'>&nbsp;";
calendario += "&nbsp;" + dev_titulo() + "&nbsp;&nbsp;";
calendario += "<input class=boton type='button' name='b3' onclick='cambioMes(0,1)'
value='>'>&nbsp;";
calendario += "<input class=boton type='button' name='b3' onclick='cambioMes(1,0)'
value='>>'>";
calendario += "</td></tr>";

calendario += "<tr class='cTit'>";
calendario += "<td>L</td>";
calendario += "<td>M</td>";
calendario += "<td>X</td>";
calendario += "<td>J</td>";
calendario += "<td>V</td>";
calendario += "<td>S</td>";

```

```

calendario += "<td>D</td>";
calendario += "</tr>";

//primera semana
diaCont=1;
calendario += "<tr>";
for (q=1;q<primerDiaMes;q++) calendario += celdaGris();
for (diaCont=1;diaCont+q-1<=7;diaCont++) calendario += celdaDia(diaCont);
calendario += "</tr>";

//resto de dias
q=1
while(diaCont<=diasDelMes)
{
    if (q==1) calendario += "<tr>";
    calendario += celdaDia(diaCont);
    if (q==7)
    {
        calendario += "</tr>";
        q=0;
    }
    q++;
    diaCont++;
}

//completamos ultima linea
for (q=q-1;q<7;q++) calendario += celdaGris()

calendario += "</table>";
return calendario;
}

////////////////////////////////////
////////////////////////////////////
function iniciar()
{

    var hoy = new Date();
    var mes = hoy.getMonth() + 1;

```



```

var dia = hoy.getDate();
var anyo = hoy.getFullYear();

document.form1.dia.value = dia;
document.form1.mes.value = mes;
document.form1.anyo.value = anyo;

var campo = valParam(window.location.href, "c");
document.form1.campo.value = campo

dibujar();
}
////////////////////////////////////
////////////////////////////////////
function dibujar()
{
    micalendario.innerHTML = codigoCalendario();
    //alert(dev_titulo());
    //titulo.innerHTML = dev_titulo();
}

////////////////////////////////////
////////////////////////////////////
function dev_diasMes(xmes,xanyo)
{
    switch(xmes + "")
    {
        case '1': return 31;break;
        case '2': if (xanyo%4==0) return 29;
                    else return 28;
                    break;

        case '3': return 31;break;
        case '4': return 30;break;
        case '5': return 31;break;
        case '6': return 30;break;
        case '7': return 31;break;
        case '8': return 31;break;
        case '9': return 30;break;
        case '10': return 31;break;
        case '11': return 30;break;
    }
}

```

```

        case '12': return 31;break;
    }

}

////////////////////////////////////
////////////////////////////////////

function cambioMes(a2,m2)
{

    m1 = document.form1.mes.value;
    a1 = document.form1.anyo.value;

    aN = parseInt(a1) + parseInt(a2);

    mN = parseInt(m1) + parseInt(m2);

    if (mN==0)
    {
        mN = 12;
        aN--;
    }

    if (mN==13)
    {
        mN = 1;
        aN++;
    }

    //var url = pagina + "?a=" + aN + "&m=" + mN;

    //location.href=url;
    //alert(url)
    document.form1.mes.value = mN;
    document.form1.anyo.value = aN;
}

```

```

document.form1.dia.value = 1;
dibujar();

}
////////////////////////////////////////////////////////////////
////////////////////////////////////////////////////////////////

function celdaDia(d)
{
    return "<td class='cDia'><a href='#' onclick=seleccionaDia(\" + d + ") class='enlace'>" + d
+ "</a></td>";
}
////////////////////////////////////////////////////////////////
////////////////////////////////////////////////////////////////
function celdaGris()
{
    return "<td class='cGris'>&nbsp;</td>";
}
////////////////////////////////////////////////////////////////
////////////////////////////////////////////////////////////////

function valParam(url, prmName)
{
    var remplaza = "/+gi";

    var cadena = unescape(url);
    cadena = cadena.replace(remplaza, " ");
    var param = prmName;
    var pos = cadena.indexOf("?");
    if (pos>=0)
    {
        cadena = cadena.substring(pos+1,cadena.length)
        vParams = cadena.split("&");

        for(var q=0;q<vParams.length;q++)
        {
            vValores = vParams[q].split("=");
            if (vValores[0]==param) return vValores[1];
        }
    }
}

```

```

        }
    }
    else return "";
}

/////////////////////////////////////////////////////////////////
/////////////////////////////////////////////////////////////////
function dev_nombreMes(m)
{
switch(m + "")
{
    case '1': return 'ENE';break;
    case '2': return 'FEB';break;
    case '3': return 'MAR';break;
    case '4': return 'ABR';break;
    case '5': return 'MAY';break;
    case '6': return 'JUN';break;
    case '7': return 'JUL';break;
    case '8': return 'AGO';break;
    case '9': return 'SEP';break;
    case '10': return 'OCT';break;
    case '11': return 'NOV';break;
    case '12': return 'DIC';break;
    default: return '-';break;
}
}

/////////////////////////////////////////////////////////////////
/////////////////////////////////////////////////////////////////
function seleccionaDia(x)
{
    var mes = document.form1.mes.value;
    var anyo = document.form1.anyo.value;
    var campo = document.form1.campo.value;
    var dev = ""
    dev = ponCero(x) + "/" + ponCero(mes) + "/" + anyo;

    cadena = "window.opener." + campo + ".value=" + dev + "";
    eval(cadena);
}

```

```

        window.close();
    }
    ///////////////////////////////////////////////////////////////////
    ///////////////////////////////////////////////////////////////////
    function ponCero(num)
    {
        if (parseInt(num) < 10) return "0" + num;
        else return num;
    }

```

Por otro lado todas las operaciones comunes y repetitivas que se utilizan en la gestión de los pedidos están incluidas en los archivos fuente *.asp : funcionesASP, datosUSR.asp, y datos.asp. A continuación vamos a destacar algunas características que consideramos importantes.

Funciones de interés del sistema

- **actualizaEstadoPedido:** actualiza el estado del pedido.
- **calcula_FacturacionPorCentro:** actualiza la tabla centros_facturación en función del total de cada factura y los centros de coste que esta tenga.
- **actualizaEstadoPropuesta:** actualiza el estado de la propuesta
- **CCM_controlConta=true:** este update bloquea el cambio de centros de coste para usuarios que no sean de contabilidad

Cuando se llama a las funciones anteriores

- Al crear una factura (recepción de un pedido):
 - **actualizaEstadoPedido**
 - **calcula_FacturacionPorCentro**
- Al cambiar los centros de coste de una propuesta:
 - Para cada factura de la propuesta: calcula_FacturacionPorCentro
 - Si el usuario es un usuario de contabilidad, CCM_controlConta=true
- Al crear un pedido
 - ActualizaEstadoPropuesta
- Al conciliar una factura:
 - CCM_controlConta=true
- Al añadir una línea de detalle a un pedido:
 - actualizaEstadoPropuesta
- Al quitar una línea de detalle a un pedido:
 - actualizaEstadoPropuesta

Apéndice C

Consultas de la base de datos

Como ya dijimos en el capítulo de diseño, pasamos a enumerar en este apéndice el detalle de las consultas que integra la base de datos viajesDATOS

Listado de consultas

- **Consulta: c_propuestas_cuentas**

SQL

```
SELECT PROPUESTAS_CENTROS.codpro, CUENTAS.codcue, CUENTAS.cuenta,
Count(*) AS n_reggs
FROM CUENTAS INNER JOIN (CENTROS_CUENTAS INNER JOIN
PROPUESTAS_CENTROS ON
CENTROS_CUENTAS.codcen=PROPUESTAS_CENTROS.codcen) ON
CUENTAS.codcue=CENTROS_CUENTAS.codcue
GROUP BY PROPUESTAS_CENTROS.codpro, CUENTAS.codcue,
CUENTAS.cuenta;
```

Columnas

Nombre	Tipo	Tamaño
codpro	Entero largo	4
codcue	Texto	15
cuenta	Texto	50
n_reggs	Entero largo	4

- **Consulta: facturas_cab**

SQL

```
SELECT FACTURAS.codfac, FACTURAS.fecha, FACTURAS.num_fact,
FACTURAS.accion,
PROVEEDORES.codprov, PROVEEDORES.proveedor, PEDIDOS.codpro,
PEDIDOS.codped,
```

```

    FACTURAS.fecha_envioConta,    FACTURAS.base16,    FACTURAS.base7,
FACTURAS.base0,
    FACTURAS.cuotas16,    FACTURAS.cuotas7,    FACTURAS.cuotas0,
[base16]+[base7]+[base0] AS totalB,
    [cuotas16]+[cuotas7]+[cuotas0] AS totalC, [totalB]+[totalC] AS totalF
FROM PROVEEDORES INNER JOIN (PEDIDOS INNER JOIN FACTURAS ON
PEDIDOS.codped =
    FACTURAS.codped) ON PROVEEDORES.codprov = PEDIDOS.codprov;

```

Columnas

Nombre	Tipo	Tamaño
codfac	Entero largo	4
fecha	Fecha/Hora	8
num_fact	Texto	50
accion	Texto	10
codprov	Texto	15
proveedor	Texto	50
codpro	Entero largo	4
codped	Entero largo	4
fecha_envioConta	Fecha/Hora	8
base16	: Consulta de actualización	16
base7	: Consulta de actualización	16
base0	: Consulta de actualización	16
cuotas16	: Consulta de actualización	16
cuotas7	: Consulta de actualización	16
cuotas0	: Consulta de actualización	16
totalB	: Consulta de actualización	16
totalC	: Consulta de actualización	16
totalF	: Consulta de actualización	16

- **Consulta: facturas_det**

SQL

```

SELECT    PEDIDOS_DETALLE.codlin,    PEDIDOS_DETALLE.codfac,
PEDIDOS_DETALLE.codped,
    FACTURAS.fecha_envioConta, ARTICULOS.tipo_iva, ARTICULOS.codart
FROM ARTICULOS INNER JOIN (FACTURAS RIGHT JOIN PEDIDOS_DETALLE ON
    FACTURAS.codfac=PEDIDOS_DETALLE.codfac) ON
    ARTICULOS.codart=PEDIDOS_DETALLE.codart;

```

Columnas

Nombre	Tipo	Tamaño
codlin	Entero largo	4
codfac	Entero largo	4
codped	Entero largo	4
fecha_envioConta	Fecha/Hora	8
tipo_iva	: Consulta de actualización	16
codart	Entero largo	4

▪ Consulta: facturas_enviadas

SQL

```
SELECT FACTURAS.*, PEDIDOS.codpro
FROM PEDIDOS INNER JOIN FACTURAS ON
PEDIDOS.codped=FACTURAS.codped
WHERE (((FACTURAS.fecha_envioConta) Is Not Null));
```

Columnas

Nombre	Tipo	Tamaño
codfac	Entero largo	4
fecha	Fecha/Hora	8
codped	Entero largo	4
num_fact	Texto	50
accion	Texto	10
fecha_envioConta	Fecha/Hora	8
base16	: Consulta de actualización	16
base7	: Consulta de actualización	16
base0	: Consulta de actualización	16
cuotas16	: Consulta de actualización	16
cuotas7	: Consulta de actualización	16
cuotas0	: Consulta de actualización	16
fecha_conciliada	Fecha/Hora	8
codpro	Entero largo	4

- **Consulta: facturas_por_pedido**

SQL

```
SELECT PEDIDOS.codped, Count(FACTURAS.codfac) AS nfacs
FROM PEDIDOS LEFT JOIN FACTURAS ON PEDIDOS.codped=FACTURAS.codped
GROUP BY PEDIDOS.codped;
```

Columnas

Nombre	Tipo	Tamaño
codped	Entero largo	4
nfacs	Entero largo	4

- **Consulta: facturas_proveedor**

SQL

```
SELECT PEDIDOS.codprov, FACTURAS.codfac, FACTURAS.num_fact,
FACTURAS.codped
FROM PEDIDOS INNER JOIN FACTURAS ON PEDIDOS.codped =
FACTURAS.codped;
```

Columnas

Nombre	Tipo	Tamaño
codprov	Texto	15
codfac	Entero largo	4
num_fact	Texto	50
codped	Entero largo	4

- **Consulta: informeConta1**

SQL

```
SELECT pedidos_cab.codprov, FX.num_fact, FX.codfac, FX.fecha AS fechaFactura,
CENTROS_FACTURACION.codcen,          CENTROS_FACTURACION.valor,
PROPUESTAS.codcue, FX.accion,
FX.fecha_envioConta, PROPUESTAS.codpro, Format([fechaFactura], "yyyymmdd") AS
FEC,
FX.fecha_conciliada,          pedidos_cab.fmin,          PROPUESTAS.descripcion,
viajantes_por_propuesta.viajantes,
viajantes_por_propuesta.maxLogin, pedidos_cab.codped
FROM ((FACTURAS_ENVIADAS AS FX INNER JOIN CENTROS_FACTURACION ON
FX.codfac=CENTROS_FACTURACION.codfac) INNER JOIN (PROPUESTAS INNER
JOIN pedidos_cab ON
PROPUESTAS.codpro=pedidos_cab.codpro) ON FX.codped=pedidos_cab.codped)
INNER JOIN
viajantes_por_propuesta ON PROPUESTAS.codpro=viajantes_por_propuesta.codpro
ORDER BY FX.num_fact, FX.codfac, FX.fecha, CENTROS_FACTURACION.codcen;
```

Columnas

Nombre	Tipo	Tamaño
codprov	Texto	15
num_fact	Texto	50
codfac	Entero largo	4
fechaFactura	Fecha/Hora	8
codcen	Texto	50
valor	: Consulta de actualización	16
codcue	Texto	15
accion	Texto	10
fecha_envioConta	Fecha/Hora	8
codpro	Entero largo	4
FEC	Texto	0
fecha_conciliada	Fecha/Hora	8
fmin	Fecha/Hora	8
descripcion	Texto	255
viajantes	Entero largo	4
maxLogin	Texto	0
codped	Entero largo	4

- **Consulta: informeConta2**

SQL

```

SELECT PEDIDOS.codped, PROPUESTAS.codpro, PROPUESTAS.descripcion,
PROPUESTAS.codcue,
    USUARIOS.login, PEDIDOS.estado, PEDIDOS.codprov, ESTADOS.estado,
pedidos_fechaMAX.fmax,
    pedidos_fechaMIN.fmin, facturas_por_pedido.nfacs
FROM (USUARIOS INNER JOIN (PROPUESTAS INNER JOIN (((ESTADOS INNER
JOIN PEDIDOS ON
    ESTADOS.codest = PEDIDOS.estado) INNER JOIN pedidos_fechaMAX ON
PEDIDOS.codped =
    pedidos_fechaMAX.codped) INNER JOIN pedidos_fechaMIN ON PEDIDOS.codped =
pedidos_fechaMIN.codped)
    ON PROPUESTAS.codpro = PEDIDOS.codpro) ON USUARIOS.codusr =
PROPUESTAS.codusr) INNER JOIN
    facturas_por_pedido ON PEDIDOS.codped = facturas_por_pedido.codped
WHERE (((PEDIDOS.estado)<110));

```

Columnas

Nombre	Tipo	Tamaño
codped	Entero largo	4
codpro	Entero largo	4
descripcion	Texto	255
codcue	Texto	15
login	Texto	50
PEDIDOS.estado	Entero largo	4
codprov	Texto	15
ESTADOS.estado	Texto	50
fmax	Fecha/Hora	8
fmin	Fecha/Hora	8
nfacs	Entero largo	4

- **Consulta: pedidos_cab**

SQL

```

SELECT  PEDIDOS.codped,    PEDIDOS.codprov,    PEDIDOS.fecha_creacion,
PEDIDOS.codpro,
        PROVEEDORES.proveedor,    pedidos_fechaMIN.fmin,    pedidos_fechaMAX.fmax,
Count(PEDIDOS_DETALLE.codlin)
        AS  lineas,    ESTADOS.codest,    ESTADOS.estado,    facturas_por_pedido.nfacs,
PROPUESTAS.descripcion AS
        descProp,    PROPUESTAS.asignado    AS    propAsig,    PEDIDOS.fecha_print,
PROPUESTAS.user_block,
        PEDIDOS.coment_pedido,    USUARIOS.login
FROM  USUARIOS INNER JOIN ((PROVEEDORES INNER JOIN (PROPUESTAS
INNER JOIN (((PEDIDOS LEFT JOIN
        pedidos_fechaMAX ON PEDIDOS.codped=pedidos_fechaMAX.codped) INNER JOIN
ESTADOS ON
        PEDIDOS.estado=ESTADOS.codest) LEFT JOIN pedidos_fechaMIN ON
        PEDIDOS.codped=pedidos_fechaMIN.codped) INNER JOIN facturas_por_pedido ON
        PEDIDOS.codped=facturas_por_pedido.codped) ON
PROPUESTAS.codpro=PEDIDOS.codpro) ON
        PROVEEDORES.codprov=PEDIDOS.codprov) LEFT JOIN PEDIDOS_DETALLE ON
        PEDIDOS.codped=PEDIDOS_DETALLE.codped) ON
USUARIOS.codusr=PROPUESTAS.codusr
GROUP BY  PEDIDOS.codped,    PEDIDOS.codprov,    PEDIDOS.fecha_creacion,
PEDIDOS.codpro,
        PROVEEDORES.proveedor,    pedidos_fechaMIN.fmin,    pedidos_fechaMAX.fmax,
ESTADOS.codest,
        ESTADOS.estado,    facturas_por_pedido.nfacs,    PROPUESTAS.descripcion,
PROPUESTAS.asignado,
        PEDIDOS.fecha_print,    PROPUESTAS.user_block,    PEDIDOS.coment_pedido,
USUARIOS.login;

```

Columnas

Nombre	Tipo	Tamaño
codped	Entero largo	4
codprov	Texto	15
fecha_creacion	Fecha/Hora	8
codpro	Entero largo	4
proveedor	Texto	50
fmin	Fecha/Hora	8

fmax	Fecha/Hora	8
lineas	Entero largo	4
codest	Entero	2
estado	Texto	50
nfacs	Entero largo	4
descProp	Texto	255
propAsig	Entero largo	4
fecha_print	Fecha/Hora	8
user_block	Entero largo	4
coment_pedido	Texto	255
login	Texto	50

- **Consulta: pedidos_det**

SQL

```

SELECT  PEDIDOS_DETALLE.codlin,  PEDIDOS.codpro,  USUARIOS.nombre,
USUARIOS.apellidos,
        PEDIDOS_DETALLE.codped,                                PEDIDOS_DETALLE.codpdt,
PEDIDOS_DETALLE.codart,
        PEDIDOS_DETALLE.fecha1,                                PEDIDOS_DETALLE.fecha2,
PEDIDOS_DETALLE.ciudad1,
        PEDIDOS_DETALLE.ciudad2,                                PEDIDOS_DETALLE.codusr,
PEDIDOS_DETALLE.hora1, PEDIDOS_DETALLE.hora2,
        USUARIOS.dni,  ARTICULOS.articulo,  ARTICULOS.icono,  ARTICULOS.tipo AS
tipo_articulo, PEDIDOS.codprov,
        PEDIDOS_DETALLE.codusr2,                                PEDIDOS_DETALLE.comentarios,
pedidos_det_nlineas_pdt.n_linea_pdt,
        PEDIDOS_DETALLE.estado, PEDIDOS_DETALLE.codfac, USUARIOS.codcen
FROM PEDIDOS INNER JOIN (ARTICULOS INNER JOIN (pedidos_det_nlineas_pdt
RIGHT JOIN
        (PEDIDOS_DETALLE INNER JOIN USUARIOS ON PEDIDOS_DETALLE.codusr =
USUARIOS.codusr) ON
        pedidos_det_nlineas_pdt.codpdt      =      PEDIDOS_DETALLE.codpdt)      ON
ARTICULOS.codart =
        PEDIDOS_DETALLE.codart) ON PEDIDOS.codped = PEDIDOS_DETALLE.codped
ORDER BY USUARIOS.nombre, USUARIOS.apellidos, PEDIDOS_DETALLE.fecha1,
PEDIDOS_DETALLE.fecha2,
        ARTICULOS.orden, PEDIDOS_DETALLE.hora1;

```

Columnas

Nombre	Tipo	Tamaño
codlin	Entero largo	4
codpro	Entero largo	4
nombre	Texto	100
apellidos	Texto	150
codped	Entero largo	4
codpdt	Entero largo	4
codart	Entero largo	4
fecha1	Fecha/Hora	8
fecha2	Fecha/Hora	8
ciudad1	Texto	255
ciudad2	Texto	255
codusr	Entero largo	4
hora1	Texto	5
hora2	Texto	5
dni	Texto	25
articulo	Texto	50
icono	Texto	50
tipo_articulo	Texto	50
codprov	Texto	15
codusr2	Entero largo	4
comentarios	Texto	250
n_linea_pdt	Entero largo	4
estado	Texto	1
codfac	Entero largo	4
codcen	Texto	15

- **Consulta: pedidos_det_nlineas_pdt**

SQL

```
SELECT  PEDIDOS_DETALLE.codpdt, Count(PEDIDOS_DETALLE.codlin) AS
```

```
n_linea_pdt
FROM PEDIDOS_DETALLE
GROUP BY PEDIDOS_DETALLE.codpdt;
```

Columnas

Nombre	Tipo	Tamaño
codpdt	Entero largo	4
n_linea_pdt	Entero largo	4

- **Consulta: pedidos_det_temp**

SQL

```
SELECT PD.codlin, PD.codped, PD.codpdt, PD.codart, PD.fecha2, PD.ciudad1,
PD.ciudad2, PD.codusr,
PD.hora1, PD.hora2, USUARIOS.nombre, USUARIOS.apellidos, PD.fecha1,
USUARIOS.dni,
ARTICULOS.articulo, ARTICULOS.icono, ARTICULOS.tipo AS tipo_articulo,
PEDIDOS.codprov, PEDIDOS.codpro,
PD.usuario, PD.codusr2
FROM PEDIDOS INNER JOIN (ARTICULOS INNER JOIN (PEDIDOS_TEMP AS PD
INNER JOIN USUARIOS ON
PD.codusr = USUARIOS.codusr) ON ARTICULOS.codart = PD.codart) ON
PEDIDOS.codped = PD.codped
ORDER BY USUARIOS.nombre, USUARIOS.apellidos, PD.fecha1,
ARTICULOS.orden, PD.hora1;
```

- **Consulta: pedidos_fecha1MAX**

SQL

```
SELECT PEDIDOS_DETALLE.codped, Max(PEDIDOS_DETALLE.fecha1) AS f1
FROM PEDIDOS_DETALLE
GROUP BY PEDIDOS_DETALLE.codped;
```

Columnas

Nombre	Tipo	Tamaño
codped	Entero largo	4
f1	Fecha/Hora	8

- **Consulta: pedidos_fecha1MIN**

SQL

```
SELECT PEDIDOS_DETALLE.codped, Min(PEDIDOS_DETALLE.fecha1) AS f1
FROM PEDIDOS_DETALLE
GROUP BY PEDIDOS_DETALLE.codped;
```

Columnas

Nombre	Tipo	Tamaño
codped	Entero largo	4
f1	Fecha/Hora	8

- **Consulta: pedidos_fecha2MAX**

SQL

```
SELECT PEDIDOS_DETALLE.codped, Max(PEDIDOS_DETALLE.fecha2) AS f2
FROM PEDIDOS_DETALLE
GROUP BY PEDIDOS_DETALLE.codped;
```

Columnas

Nombre	Tipo	Tamaño
codped	Entero largo	4
f2	Fecha/Hora	8

- **Consulta: pedidos_fecha2MIN**

SQL

```
SELECT PEDIDOS_DETALLE.codped, Min(PEDIDOS_DETALLE.fecha2) AS f2
FROM PEDIDOS_DETALLE
GROUP BY PEDIDOS_DETALLE.codped;
```

Columnas

Nombre	Tipo	Tamaño
codped	Entero largo	4
f2	Fecha/Hora	8

- **Consulta: pedidos_fechaMAX**

SQL

```
SELECT          pedidos_fecha2MAX.codped,          pedidos_fecha1MAX.f1,
pedidos_fecha2MAX.f2, If([f2]>[f1],[f2],[f1])
AS fmax, PEDIDOS.codpro
FROM  PEDIDOS  INNER JOIN  (pedidos_fecha2MAX  RIGHT JOIN
pedidos_fecha1MAX ON
pedidos_fecha2MAX.codped=pedidos_fecha1MAX.codped) ON
PEDIDOS.codped=pedidos_fecha1MAX.codped;
```

Columnas

Nombre	Tipo	Tamaño
codped	Entero largo	4
f1	Fecha/Hora	8
f2	Fecha/Hora	8
fmax	Fecha/Hora	8
codpro	Entero largo	4

- **Consulta: pedidos_fechaMIN**

SQL

```
SELECT PEDIDOS.codped, PEDIDOS.codpro, IIf([f2]<[f1],[f2],[f1]) AS fmin,
pedidos_fecha1MIN.f1,
pedidos_fecha2MIN.f2
FROM (PEDIDOS INNER JOIN pedidos_fecha1MIN ON
PEDIDOS.codped=pedidos_fecha1MIN.codped) INNER
JOIN pedidos_fecha2MIN ON PEDIDOS.codped=pedidos_fecha2MIN.codped;
```

Columnas

Nombre	Tipo	Tamaño
codped	Entero largo	4
codpro	Entero largo	4
fmin	Fecha/Hora	8
f1	Fecha/Hora	8
f2	Fecha/Hora	8

▪ Consulta: propuestas_cab

SQL

```
SELECT PROPUESTAS.codpro, PROPUESTAS.codusr, PROPUESTAS.descripcion,
PROPUESTAS.fecha_creacion,
PROPUESTAS.fecha_envio, ESTADOS.codest, ESTADOS.estado,
propuestas_lineas.lineas, CUENTAS.cuenta,
CUENTAS.codcue, USUARIOS.login, PROPUESTAS.comentarios,
PROPUESTAS.asignado,
propuestasFechaMAX.finViaje, PROPUESTAS.user_block, PROPUESTAS.prioridad,
DateDiff("d",[fecha_creacion],Date()) AS ndias
FROM USUARIOS INNER JOIN (CUENTAS RIGHT JOIN (((PROPUESTAS INNER
JOIN ESTADOS ON
PROPUESTAS.estado=ESTADOS.codest) LEFT JOIN propuestas_lineas ON
PROPUESTAS.codpro=propuestas_lineas.codpro) LEFT JOIN propuestasFechaMAX
ON
PROPUESTAS.codpro=propuestasFechaMAX.codpro) ON
CUENTAS.codcue=PROPUESTAS.codcue) ON
USUARIOS.codusr=PROPUESTAS.codusr
ORDER BY PROPUESTAS.fecha_creacion;
```

Columnas

Nombre	Tipo	Tamaño
codpro	Entero largo	4
codusr	Entero largo	4
descripcion	Texto	255
fecha_creacion	Fecha/Hora	8
fecha_envio	Fecha/Hora	8
codest	Entero	2
estado	Texto	50
lineas	Entero largo	4
cuenta	Texto	50
codcue	Texto	15
login	Texto	50
comentarios	Texto	255
asignado	Entero largo	4
finViaje	Fecha/Hora	8
user_block	Entero largo	4
prioridad	Texto	1
ndias	Entero largo	4

▪ Consulta: propuestas_det

SQL

```
SELECT      USUARIOS.nombre,      USUARIOS.apellidos,      USUARIOS.dni,
PROPUESTAS_DETALLE.*,
ARTICULOS.articulo, ARTICULOS.icono, ARTICULOS.tipo AS tipo_articulo
FROM ARTICULOS INNER JOIN (USUARIOS INNER JOIN PROPUESTAS_DETALLE
ON
USUARIOS.codusr=PROPUESTAS_DETALLE.codusr) ON
ARTICULOS.codart=PROPUESTAS_DETALLE.codart
ORDER      BY      USUARIOS.nombre,      USUARIOS.apellidos,
PROPUESTAS_DETALLE.fecha, ARTICULOS.orden,
PROPUESTAS_DETALLE.hora;
```

Columnas

Nombre	Tipo	Tamaño
nombre	Texto	100

apellidos	Texto	150
dni	Texto	25
codpdt	Entero largo	4
codpro	Entero largo	4
codusr	Entero	2
codart	Entero largo	4
ciudad1	Texto	255
ciudad2	Texto	255
fecha	Fecha/Hora	8
hora	Texto	7
tipo_hora	Texto	5
noches	Entero	2
codusr2	Entero largo	4
comentario	Texto	200
articulo	Texto	50
icono	Texto	50
tipo_articulo	Texto	50

- **Consulta: propuestas_det_SIN_PEDIDO**

SQL

```

SELECT propuestas_det.*, PEDIDOS_DETALLE.codlin
FROM (propuestas_det LEFT JOIN PEDIDOS_DETALLE ON propuestas_det.codpdt =
PEDIDOS_DETALLE.codpdt) INNER JOIN ARTICULOS ON propuestas_det.codart =
ARTICULOS.codart
WHERE (((PEDIDOS_DETALLE.codlin) Is Null))
ORDER BY propuestas_det.nombre, propuestas_det.apellidos, propuestas_det.fecha,
ARTICULOS.orden;

```

Columnas

Nombre	Tipo	Tamaño
nombre	Texto	100
apellidos	Texto	150
dni	Texto	25
codpdt	Entero largo	4
codpro	Entero largo	4
codusr	Entero	2
codart	Entero largo	4

ciudad1	Texto	255
ciudad2	Texto	255
fecha	Fecha/Hora	8
hora	Texto	7
tipo_hora	Texto	5
noches	Entero	2
codusr2	Entero largo	4
comentario	Texto	200
articulo	Texto	50
icono	Texto	50
tipo_articulo	Texto	50
codlin	Entero largo	4

- **Consulta: propuestas_det_temp**

SQL

```

SELECT PT.*, ARTICULOS.articulo, ARTICULOS.icono, ARTICULOS.tipo AS
tipo_articulo
FROM ARTICULOS INNER JOIN PROPUESTAS_TEMP AS PT ON
ARTICULOS.codart = PT.codart
ORDER BY PT.fecha, ARTICULOS.orden, PT.hora;

```

Columnas

Nombre	Tipo	Tamaño
codlin	Entero largo	4
usuario	Entero largo	4
codart	Entero largo	4
ciudad1	Texto	255
ciudad2	Texto	255
fecha	Fecha/Hora	8
hora	Texto	7
tipo_hora	Texto	5
noches	Entero	2
articulo	Texto	50
icono	Texto	50
tipo_articulo	Texto	50

- **Consulta: propuestas_lineas**

SQL

```
SELECT PROPUESTAS.codpro, Count(PROPUESTAS_DETALLE.codpdt) AS lineas
FROM PROPUESTAS LEFT JOIN PROPUESTAS_DETALLE ON
PROPUESTAS.codpro=PROPUESTAS_DETALLE.codpro
GROUP BY PROPUESTAS.codpro;
```

Columnas

Nombre	Tipo	Tamaño
codpro	Entero largo	4
lineas	Entero largo	4

- **Consulta: propuestas_pedidos_facturas**

SQL

```
SELECT PROPUESTAS.codpro, PEDIDOS.codped, FACTURAS.codfac,
PROPUESTAS.CCM_controlConta
FROM PROPUESTAS INNER JOIN (PEDIDOS INNER JOIN FACTURAS ON
PEDIDOS.codped = FACTURAS.codped)
ON PROPUESTAS.codpro = PEDIDOS.codpro;
```

Columnas

Nombre	Tipo	Tamaño
codpro	Entero largo	4
codped	Entero largo	4
codfac	Entero largo	4
CCM_controlConta	Sí/No	1

- **Consulta: propuestas_peorEstadoPedido**

SQL

```
SELECT PEDIDOS.codpro, Min(PEDIDOS.estado) AS peorEstado
```

```
FROM PEDIDOS
GROUP BY PEDIDOS.codpro;
```

Columnas

Nombre	Tipo	Tamaño
codpro	Entero largo	4
peorEstado	Entero largo	4

- **Consulta: propuestasFechaMAX**

SQL

```
SELECT pedidos_fechaMAX.codpro, Max(pedidos_fechaMAX.fmax) AS finViaje
FROM pedidos_fechaMAX
GROUP BY pedidos_fechaMAX.codpro;
```

Columnas

Nombre	Tipo	Tamaño
codpro	Entero largo	4
finViaje	Fecha/Hora	8

- **Consulta: usuarios_comentarios_defecto**

SQL

```
SELECT          USUARIOS.codusr,          USUARIOS.comentario_pedidos,
Len([comentario_pedidos] & "") AS n_chars
FROM USUARIOS;
```

Columnas

Nombre	Tipo	Tamaño
codusr	Entero largo	4
comentario_pedidos	Texto	255
n_chars	Entero largo	4

- **Consulta: usuarios_pedido_detalle**

SQL

```
SELECT DISTINCT PEDIDOS_DETALLE.codped, PEDIDOS_DETALLE.codusr
FROM PEDIDOS_DETALLE;
```

Columnas

Nombre	Tipo	Tamaño
codped	Entero largo	4
codusr	Entero largo	4

- **Consulta: usuariosLogin**

SQL

```
SELECT      USUARIOS.codusr,      USUARIOS.apellidos,      USUARIOS.nombre,
USUARIOS.login,
      USUARIOS_PERMISOS.codper,      USUARIOS_PERMISOS.permisos,
USUARIOS_PERMISOS.descripcion,
      USUARIOS.email, CENTROS.codcen, CENTROS.centro, USUARIOS.codusj
FROM USUARIOS_PERMISOS INNER JOIN (CENTROS INNER JOIN USUARIOS
ON CENTROS.codcen =
      USUARIOS.codcen) ON USUARIOS_PERMISOS.codper = USUARIOS.permisos
ORDER BY USUARIOS.apellidos, USUARIOS.nombre;
```

Columnas

Nombre	Tipo	Tamaño
codusr	Entero largo	4
apellidos	Texto	150
nombre	Texto	100
login	Texto	50
codper	Entero	2
permisos	Texto	50

descripcion	Texto	200
email	Texto	50
codcen	Texto	15
centro	Texto	50
codusj	Entero largo	4

- **Consulta: viajeros_por_propuesta**

SQL

```

SELECT PEDIDOS.codpro, Count(usuarios_pedido_detalle.codusr) AS viajeros,
Max(USUARIOS.login) AS
maxLogin
FROM USUARIOS INNER JOIN (PEDIDOS INNER JOIN usuarios_pedido_detalle ON
PEDIDOS.codped=usuarios_pedido_detalle.codped) ON
USUARIOS.codusr=usuarios_pedido_detalle.codusr
GROUP BY PEDIDOS.codpro;

```

Columnas

Nombre	Tipo	Tamaño
codpro	Entero largo	4
viajantes	Entero largo	4
maxLogin	Texto	0