

Gestión de Concesionario

Noelia Pico Gayo

Enginyeria en Informàtica

Juan Martínez Bolaños

08/10/2012

Dedicatoria

Dedicatoria

Dedico este trabajo principalmente a mi familia, que se sienten muy orgullosos de que haya llegado este momento tan importante en mi formación profesional.

A mi madre Begoña, por ser el pilar más importante y por luchar siempre en beneficio de mi formación y bienestar.

A mi padre, que a pesar de nuestra distancia física, siento que estás conmigo siempre y aunque nos faltaron muchas cosas por vivir juntos, sé que este momento hubiera sido tan especial para ti como lo es para mí.

A mi tía Yolanda, a quien quiero como una madre, por compartir momentos significativos conmigo y por estar siempre dispuesta a escucharme y ayudarme en cualquier momento.

A mi tío Tito, por dar siempre buen ejemplo y demostrar que la constancia y el trabajo son el camino correcto.

A Andrés, por haber formado juntos una familia. Por ayudarme, cuidarme, apoyarme y quererme siempre.

Introducción

Este documento define el plan de proyecto que se llevará a cabo para la realización del Proyecto Final de Carrera de la rama Bases de Datos, que versa sobre la gestión de una base de datos para una cadena de concesionarios de coches.

El objetivo del documento es identificar todas las tareas necesarias que se tienen que llevar a cabo para la consecución del proyecto, ajustándolas al calendario predefinido por la asignatura.

Esta memoria se divide en capítulos, el primero donde se define el plan de trabajo y los siguientes acordes con las diferentes fases del proyecto: Análisis, Diseño e Implementación.

Se incluyen las explicaciones necesarias para comprender y hacer uso de la base de datos realizada y entregada. El producto entregado, cumple todos los requisitos presentados en el enunciado del proyecto y además profundiza en detalle en algunos campos. No obstante, la escalabilidad con la que se ha realiza la base de datos relacional, garantiza la facilidad para añadir nuevas funcionalidades (propuestas en el presente documento).

Índice

Capítulo1: Plan de Trabajo.....	5
Objetivos del Proyecto.....	5
Metodología.....	6
Listado de tareas.....	7
Plan de riesgos.....	11
Diagrama de Gantt.....	13
Capítulo2: Análisis.....	15
Estudio inicial.....	15
Identificación de roles.....	16
Listado de requerimientos.....	17
Casos de uso.....	18
Modelo lógico de datos.....	21
Data WareHouse.....	23
Modelo relacional.....	24
Capítulo 3: Diseño.....	26
Definición de estándares.....	26
Modelo físico de datos.....	28
Capítulo 4: Implementación.....	38
Introducción.....	38
Buenas prácticas.....	38
Decisiones de Diseño.....	40
Scripts de implementación.....	43
Capítulo 5: Valoración económica.....	44
Capítulo 6: Conclusiones.....	45
Bibliografía.....	47
Anexo A – Documentación.....	48

Capítulo1: Plan de Trabajo

Objetivos del Proyecto

Al término del proyecto se pretenden conseguir los siguientes objetivos:

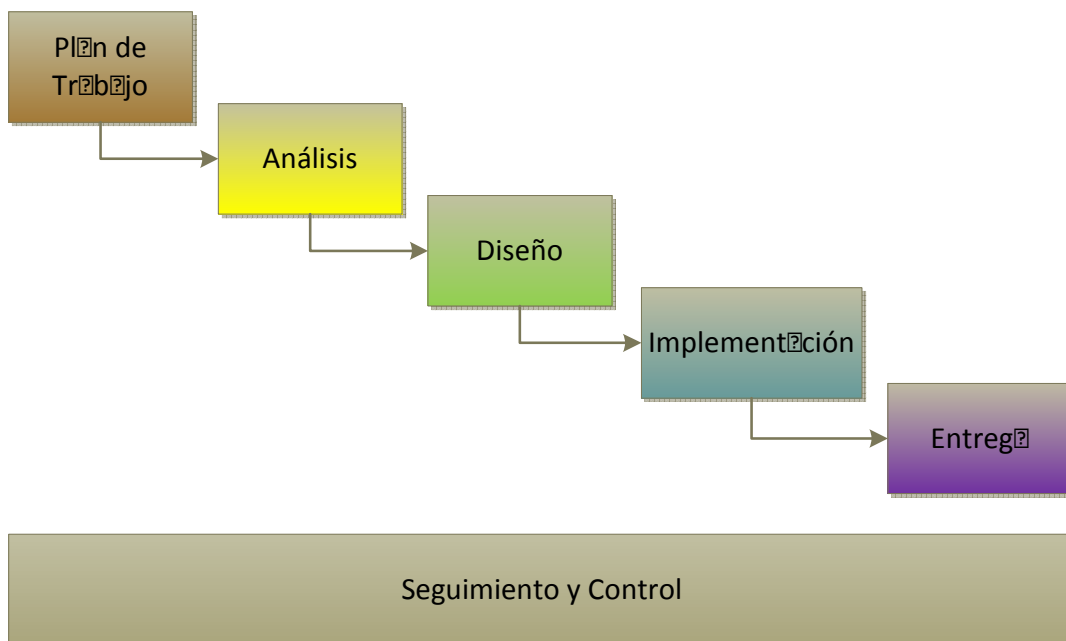
- ❖ Control de ventas:
 - Control sobre las ventas de vehículos nuevos y de segunda mano.
 - Información y estadísticas sobre los vendedores de la empresa. Esta información será muy importante a la hora de repartir primas por los beneficios de la empresa.
 - Inventario de los vehículos por concesionario y por tanto reducción del tiempo a la hora de buscar un vehículo.
 - Estadísticas de venta que permiten determinar que modelos son los más solicitados, que ofertas han sido las más exitosas, etc.
- ❖ Control del taller:
 - Control sobre las revisiones de vehículos. Aviso a los clientes de cuándo deben hacer las revisiones de sus vehículos.
 - Control sobre las reparaciones de vehículos.
 - Estadísticas de tiempo medio por tipo de avería, que permitirá planificar cada vez mejor el tiempo estimado de finalización de avería, con el fin de poder planificar mejor el tiempo del personal del taller y estimar el personal necesario en cada período.
 - Control sobre la productividad del personal del taller.
- ❖ Servicio postventa:
 - Almacén de información sobre los clientes que nos facilitará la comunicación de nuevas acciones comerciales.
 - Estadísticas sobre el tipo de cliente que usa nuestros concesionarios.

Metodología

La estructuración de las tareas del proyecto se basa en una metodología en **cascada**, donde es necesario finalizar una etapa para comenzar la siguiente. He escogido esta metodología porque creo que clave del éxito de este proyecto se basa en realizar un buen análisis de los datos necesarios y posteriormente generar un diseño adecuado para ellos. Cuando estas dos primeras etapas ya están finalizadas, podremos continuar con la implantación del proyecto, generando la estructura de BD y el código PL/SQL necesario.

He descartado otro tipo de metodologías como las metodologías ágiles o por iteraciones ya que considero que en estos casos el modelo de datos irá sufriendo modificaciones a lo largo del proyecto y muchas de estas modificaciones darán lugar a realizar cambios constantemente sobre el trabajo realizado. En este caso, y debido al corto margen de tiempo que tenemos para finalizar el proyecto creo que es más eficiente realizar el proyecto paso a paso, verificando al final de cada etapa la trazabilidad de funcionalidades con la etapa anterior.

Siguiendo esta metodología el proyecto se divide en las siguientes fases:



Listado de tareas

Según hemos visto en el apartado anterior, el proyecto se divide en diferentes fases, presentamos las tareas de cada fase con su planificación prevista:

- ❖ Fase de **Plan de Trabajo**, donde se definen las tareas que hemos llevado a cabo para generar este documento.
 - La tarea del estudio preliminar se basa en leer los documentos del PFC detalladamente para evaluar el alcance del proyecto y poder llevar a cabo este documento.
 - La tarea de la evaluación del material trata de recoger todo el software y la documentación necesaria para poder llevar a cabo el proyecto.
 - El resto de tareas están definidas en este documento.

El único hito de esta fase es la *Entrega de Plan de Trabajo*

Nombre de tarea	Duración	Comienzo	Fin
Plan de trabajo			
Estudio preliminar	2 horas	jue 20/09/12	jue 20/09/12
Objetivos del proyecto	5 horas	sáb 22/09/12	sáb 22/09/12
Definición del alcance	5 horas	sáb 22/09/12	dom 23/09/12
Listado de tareas	10 horas	dom 23/09/12	mié 26/09/12
Planificación	8 horas	jue 27/09/12	sáb 29/09/12
Gestión de riesgos	4 horas	sáb 29/09/12	dom 30/09/12
Evaluación material	2 horas	dom 30/09/12	dom 30/09/12
Revisión final Plan de Trabajo	3 horas	dom 30/09/12	dom 30/09/12
Entrega Plan de Trabajo	0 horas	dom 30/09/12	dom 30/09/12

❖ Fase de Análisis.

- En esta fase se llevarán a cabo las tareas propias del análisis: listado de requerimientos, *Modelo lógico de datos*, *Modelo Entidad – Relación* y creación del *Diccionario de Datos*.
- Una de las primeras tareas es la *Instalación del Software* (SW) para generación de los documentos de esta fase.
- Debido a la falta de experiencia en las tareas pertenecientes al área del Data WareHouse (DWH), se han creado las tareas de *Recopilación y análisis de documentación* y *Análisis de informes DWH*. Se hace énfasis en estas tareas ya que necesitarán una dedicación especial para poder llevarlas a cabo.
- La tarea de *Comprobación de trazabilidad de requerimientos* se ha creado para validar que los documentos de análisis creados en este

documento cumplen todas las funcionalidades descritas en el Documento de Requerimientos.

- La tarea de *Generación documentación de memoria* se basa en incluir los documentos resultantes de esta fase en la memoria del proyecto y revisar el contenido del mismo hasta el momento. Esta tarea se repetirá en las demás fases del proyecto.

Los hitos de esta fase son los documentos resultantes:

- ✓ Documento de Requerimientos
- ✓ Documento Modelo Entidad – Relación. Servirá de base para la fase de diseño.

Nombre de tarea	Duración	Comienzo	Fin
Análisis			
Listado de requerimientos	10 horas	lun 08/10/12	vie 12/10/12
Documento Requerimientos	0 horas	vie 12/10/12	vie 12/10/12
Instalación software	3 horas	vie 12/10/12	vie 12/10/12
Recopilación y análisis documentación DWH	10 horas	vie 12/10/12	sáb 13/10/12
Modelo lógico de datos	15 horas	dom 14/10/12	jue 18/10/12
Análisis informes DWH	5 horas	sáb 20/10/12	sáb 20/10/12
Modelo Entidad - Relación	20 horas	sáb 20/10/12	sáb 27/10/12
Documento MER	0 horas	sáb 27/10/12	sáb 27/10/12
Diccionario de datos	5 horas	sáb 27/10/12	dom 28/10/12
Comprobación trazabilidad requerimientos	4 horas	dom 28/10/12	dom 28/10/12
Generación documentación memoria	10 horas	dom 28/10/12	jue 01/11/12

❖ Fase de Diseño

- La tarea de *Instalación del Software* se pone en primer lugar para que en caso de errores podamos re-planificar y corregirlos cuanto antes. (Ver Plan de Riesgos).
- Dentro de esta fase, se contemplan las tareas propias del Diseño como: *Definición de estándares, Modelo físico de datos, Arquitectura de procesos*
- Igual que en la fase de análisis, se han creado tareas específicas para las tareas relacionadas con el DWH.
- Una vez realizadas todas las tareas de diseño y análisis, sólo habrá que implementar lo que marcan los documentos generados. La tarea de *Control de Cambios* tratará de realizar todos los cambios necesarios en los documentos creados hasta el momento, después de comprobar que todas las funcionalidades se contemplan correctamente.

El único hito de esta fase es el *Documento de Diseño de Base de Datos (BD)*

Nombre de tarea	Duración	Comienzo	Fin
Diseño			
Instalación software	4 horas	jue 01/11/12	jue 01/11/12
Definición estándares	2 horas	jue 01/11/12	jue 01/11/12
Modelo físico de datos	5 horas	vie 02/11/12	vie 02/11/12
Arquitectura procesos	2 horas	vie 02/11/12	vie 02/11/12
Modelo físico de datos DWH	3 horas	vie 02/11/12	sáb 03/11/12
Arquitectura procesos DWH	2 horas	sáb 03/11/12	sáb 03/11/12
Documento Diseño BD	0 horas	sáb 03/11/12	sáb 03/11/12
Control de cambios	5 horas	sáb 03/11/12	dom 04/11/12
Generación documentación memoria	15 horas	dom 04/11/12	sáb 10/11/12

❖ Fase de **Implementación**

- La primera tarea es la *Carga de Datos* necesaria para realizar las tareas consecutivas.
- La tarea más importante de esta fase y con la mayor duración del proyecto es la tarea de *Generación de código PL/SQL* que dará funcionalidad a la BD.
- La tarea de *Juegos de prueba unitarios* probarán cada módulo de la aplicación por separado para comprobar que todos funcionan correctamente y se realizarán las correcciones necesarias. Se estima poco tiempo durante esta tarea que al generar el código PL/SQL se irá probando cada función. Esta tarea se reserva para juegos de prueba unitarios de casos más complejos.
- La tarea de *Juegos de prueba finales* se encargarán de probar la integridad del sistema y se realizarán pruebas abarquen diferentes módulos.

El único hito de esta fase es el *Documento de Pruebas Finales*

Nombre de tarea	Duración	Comienzo	Fin
Implantación			
Carga de datos	7 horas	sáb 10/11/12	lun 12/11/12
Generación código PL/SQL	85 horas	mar 13/11/12	mar 11/12/12
Juegos de prueba unitarios	10 horas	mié 12/12/12	sáb 15/12/12
Generación documentación memoria	18 horas	sáb 15/12/12	sáb 22/12/12
Juegos de prueba finales	12 horas	sáb 22/12/12	sáb 29/12/12
Documento de pruebas	0 horas	sáb 29/12/12	sáb 29/12/12

❖ Fase de **Entrega**

- En esta fase se generan todos los documentos necesarios para la entrega: archivos de BD, presentación y memoria.

Nombre de tarea	Duración	Comienzo	Fin
Entrega			
Generación archivos trabajo práctico	2 horas	sáb 29/12/12	sáb 29/12/12
Creación presentación	10 horas	sáb 29/12/12	dom 30/12/12
Revisión final memoria	30 horas	dom 30/12/12	jue 10/01/13

❖ Fase de **Seguimiento y Control**

- A lo largo del proyecto se realizarán tareas de seguimiento y control, que se llevarán a cabo cada 2 semanas. El objetivo es detectar errores y solucionarlos y re-planificar el proyecto y la gestión de riesgos si procede.

Nombre de tarea	Duración	Comienzo	Fin
Seguimiento y control			
Revisión planificación	45,25 días	sáb 22/09/12	sáb 12/01/13
Revisión planificación 1	1 hora	sáb 22/09/12	sáb 22/09/12
Revisión planificación 2	1 hora	sáb 06/10/12	sáb 06/10/12
Revisión planificación 3	1 hora	sáb 20/10/12	sáb 20/10/12
Revisión planificación 4	1 hora	sáb 03/11/12	sáb 03/11/12
Revisión planificación 5	1 hora	sáb 17/11/12	sáb 17/11/12
Revisión planificación 6	1 hora	sáb 01/12/12	sáb 01/12/12
Revisión planificación 7	1 hora	sáb 15/12/12	sáb 15/12/12
Revisión planificación 8	1 hora	sáb 29/12/12	sáb 29/12/12
Revisión planificación 9	1 hora	sáb 12/01/13	sáb 12/01/13
Gestión Riesgos	45,25 días	sáb 22/09/12	sáb 12/01/13
Gestión Riesgos 1	1 hora	sáb 22/09/12	sáb 22/09/12
Gestión Riesgos 2	1 hora	sáb 06/10/12	sáb 06/10/12
Gestión Riesgos 3	1 hora	sáb 20/10/12	sáb 20/10/12
Gestión Riesgos 4	1 hora	sáb 03/11/12	sáb 03/11/12
Gestión Riesgos 5	1 hora	sáb 17/11/12	sáb 17/11/12
Gestión Riesgos 6	1 hora	sáb 01/12/12	sáb 01/12/12
Gestión Riesgos 7	1 hora	sáb 15/12/12	sáb 15/12/12
Gestión Riesgos 8	1 hora	sáb 29/12/12	sáb 29/12/12
Gestión Riesgos 9	1 hora	sáb 12/01/13	sáb 12/01/13

Plan de riesgos

A continuación presentamos el plan de riesgos, que se compone de todos los riesgos detectados hasta el momento en el proyecto y de las medidas correctoras para mitigarlos.

ESTADO DE RIESGOS								
Código	Descripción	Causa	Consecuencia	Nvl inicial	Situación actual			
					Prob.	Imp.	Evol.	Nvl Actual
R01	Planificación demasiado optimista	Definición errónea del alcance de las tareas.	Retraso general en las tareas del proyecto	A	M	A	New	A
R02	Incumplimiento del calendario fijado por dedicación a otras asignaturas	Mala planificación de las actividades de otras asignaturas	Retraso en alguna de las tareas del proyecto	M	M	A	New	M
R03	Problemas con la instalación del SW	Problemas con la máquina o con otras versiones previamente instaladas	Retraso en las tareas de análisis o implantación	B	B	M	New	B
R04	Problemas con el uso del nuevo SW	Desconocimiento del nuevo SW instalado	Retraso en las tareas de análisis o implantación	M	M	M	New	M
R05	Pérdida del trabajo realizado en el proyecto por problemas con la máquina	Cualquier error de la máquina que provoque daños en el disco duro o en el sistema operativo	Pérdida del trabajo realizado	B	B	M	New	B
R06	No cubrir todos los requerimientos del proyecto	Error en la definición del alcance del proyecto y en la definición de la lista de requerimientos.	Proyecto incompleto si no se detecta o retraso elevado si se detecta demasiado tarde	M	M	A	New	M

ESTADO DE RIESGOS								
Código	Descripción	Causa	Consecuencia	Nvl inicial	Situación actual			
					Prob.	Imp.	Evol.	Nvl Actual
R07	Error en el análisis de algún requerimiento	Mal entendimiento del requerimiento o falta de revisión de relaciones entre requerimientos	Error en las tareas afectadas por el requerimiento	A	M	A	New	A
Nvl Inicial: Nivel inicial del riesgo [(A)lto, (M)edio, (B)ajo] Prob.: Probabilidad [(A)lto, (M)edio, (B)ajo] Imp.: Impacto [(A)lto, (M)edio, (B)ajo] Evol.: Evolución [▲ Se incrementa el riesgo. ▼ Disminuye el riesgo. ◀▶ Sin variaciones. New Nuevo riesgo] Nvl Actual: Nivel Actual [(A)lto, (M)edio, (B)ajo]								

MEDIDAS CORRECTIVAS			
Código	Acción	Tipo	Riesgo residual
R01	Añadir a la planificación del proyecto tareas de revisión periódicas para revisar el cumplimiento de la planificación	Mitigadora	Muy bajo
R02	Al detectar retraso en el proyecto debido al tiempo dedicado a otras asignaturas, re-planificar cuanto antes las tareas afectadas.	Mitigadora	Bajo
R03	Adelantar cuanto sea posible la instalación del SW necesario para saber cuanto antes si tendremos que adoptar otras acciones correctoras.	Mitigadora	Muy bajo
	Disponer de una máquina alternativa donde podamos llevar a cabo el proyecto	Correctora	Nulo
R04	Disponer de manuales de uso del SW instalado	Mitigadora	Medio
	Cambiar las aplicaciones de SW instaladas por otras aplicaciones ya conocidas	Correctora	Nulo
R05	Realizar copias de seguridad periódicas	Correctora	Muy bajo
R06	Revisar el documento de listado de requerimientos y comprobar que todos los aspectos solicitados en el enunciado están contemplados.	Mitigadora	Bajo
R07	Realizar controles de trazabilidad entre el documento de análisis y los documentos resultantes del modelo lógico de datos	Mitigadora	Bajo

Diagrama de Gantt

La planificación realizada se basa en un calendario propio para el proyecto, creado a partir de las horas disponibles para poder trabajar en las tareas del mismo.

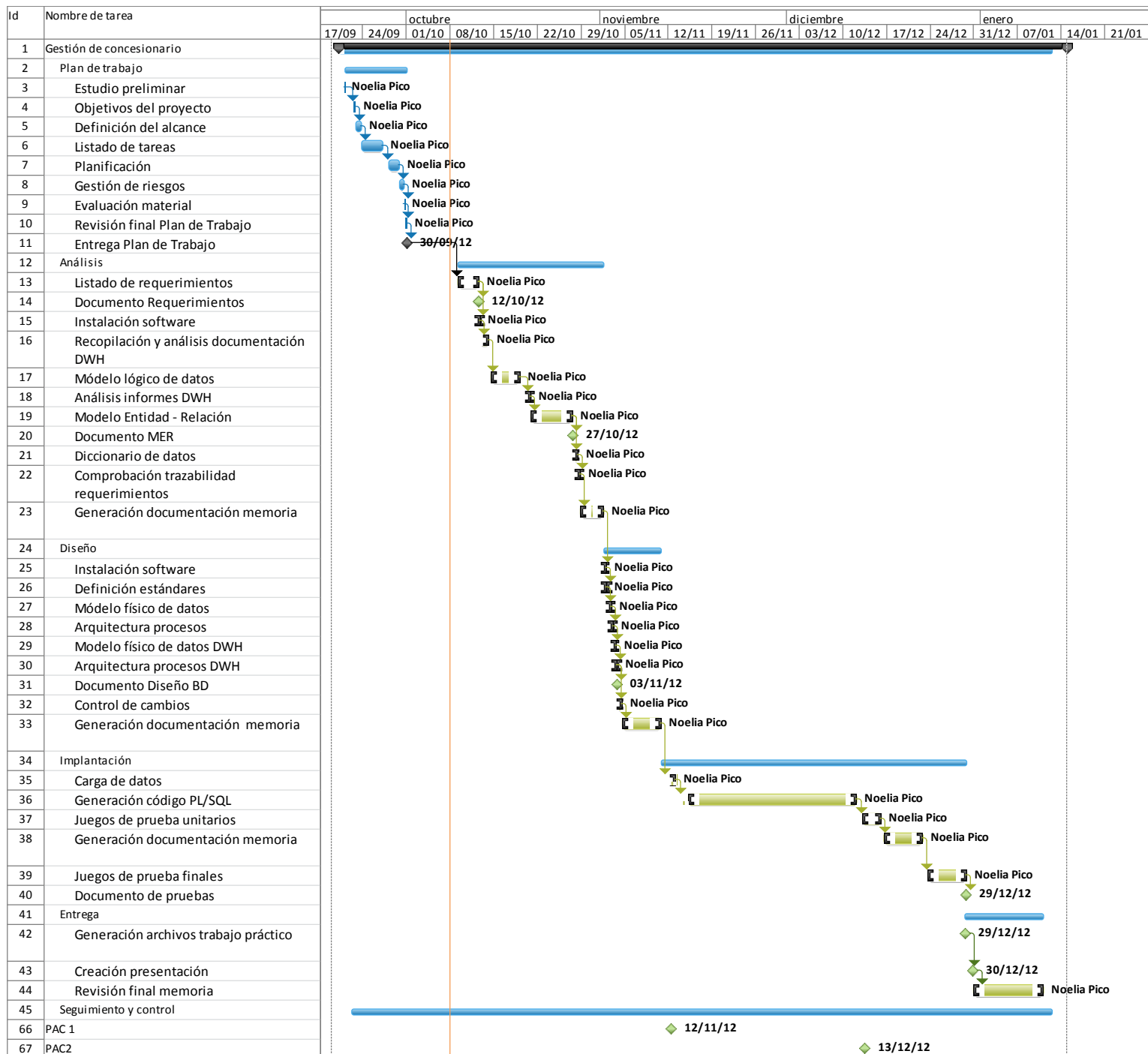
Este calendario se ha creado basándose en los siguientes puntos:

- Horario laboral, sólo se dedicarán dos horas de lunes a jueves, de 19:00h a 21:00h.
- Los sábados y domingos se marcan como laborables y se establece el horario de dedicación al proyecto de 10:00h a 1300h y de 16:00h a 21:00h.
- Viernes no laborable. Marcamos todos los viernes como no laborable ya que se dedicará este día a la asignatura Criptografía.
- Marcamos los días de entregas y prácticas de Criptografía como no laborables.
- Marcamos 3 días de vacaciones de Navidad como no laborables.
- Todos los festivos y puentes se marcan como laborables, con la misma dedicación que los fines de semana.
- Se marcan 5 días de vacaciones de lunes a viernes como laborables con la misma dedicación que los fines de semana.

En el diagrama de Gantt se muestran todas las tareas descritas anteriormente. Además se pueden ver dos hitos nuevos uno para cada entrega:

- PAC1: Coincide con el inicio de la tarea de programación, por tanto en la primera PAC estimo entregar la fase de análisis y diseño al completo.
- PAC2: Coincide con la tarea de programación al 80% por tanto estimo entregar las correcciones realizadas sobre la PAC1 y la BD con algunos módulos completamente operativos.

Para simplificar el diagrama de Gantt no se muestran en detalle las tareas de seguimiento y control.



Capítulo2: Análisis

Estudio inicial

La segunda fase del proyecto desarrolla el análisis técnico y funcional del proyecto. Comenzamos con una explicación detallada de las funcionalidades del proyecto, que amplía la información proporcionada en el enunciado. Normalmente el trabajo de síntesis se realiza a partir de las reuniones, entrevistas, cuestionarios, etc. que se mantienen con los usuarios que participan en el proyecto.

Se debe desarrollar una base de datos relacional para dar soporte al trabajo realizado en los concesionarios de vehículos de la marca *UOC*. Esta marca ficticia se dedica, a través de sus concesionarios a la venta y reparación de vehículos, tanto de primera como de segunda mano.

Todos los vehículos nuevos que se ofertan en los concesionarios *UOC* son coches de la propia marca, mientras que los coches de segunda mano pueden ser de cualquier marca. Pese a que la prioridad de venta en los concesionarios *UOC* es vender vehículos de la marca propia, el concesionario también se dedica a la venta de coches usados. Estos son adquiridos por el concesionario cuando se presenta alguna buena oferta o cuando se realiza una venta, que el comprador tiene la opción de entregar su vehículo antiguo para obtener un descuento en el nuevo vehículo. Disponemos de un inventario de coches disponibles, que incluyen los coches nuevos que vienen de fábrica y los coches usados que concesionario compra.

Los vehículos se identifican por el número de bastidor o por la matrícula. Los vehículos nuevos se matriculan en el momento de efectuar la venta.

Los comerciales de los concesionarios se dedican principalmente a la venta de vehículos. Cuando se realiza una venta se registra la marca, el modelo, las características del vehículo, los extras seleccionados y los datos del cliente y del vendedor. Las características básicas de un vehículo irá siempre en función del modelo del vehículo, en cambio las características extra varían en cada venta.

Al realizar una venta, es importante también registrar la fecha para poder avisar al cliente cuando deba realizar las revisiones de su vehículo.

Todos los concesionarios cuentan con talleres donde se realizan las revisiones periódicas y reparaciones de los vehículos. Los mecánicos llevan a cabo estas tareas, para cada una de ellas se debe registrar el vehículo implicado, el tipo de avería, la fecha y hora de entrada y salida del taller, los mecánicos que han participado en la revisión y el tiempo que le han dedicado y los datos del cliente si este no es ya conocido.

Cualquier vehículo puede ser reparado o revisado en el taller, tanto si es un vehículo previamente vendido por el concesionario como si no lo es. Para aquellos vehículos que hagan uso del servicio de taller y no sean vehículos el concesionario, deben registrarse los datos básicos para ser incorporado a la base de datos.

De los trabajadores del concesionario guardaremos los datos personales, la fecha de contratación en la empresa, la categoría profesional y el sueldo bruto anual. Esta información nos permitirá obtener información sobre una parte de los costes del concesionario.

Queda fuera del alcance de este proyecto la gestión de los recursos humanos tales como obtener el desglose detallado de su nómina, los períodos de bajas o ausencias, etc.

Deben definirse almacenes de datos que permitan obtener información estadística como: tiempo medio en reparación de un coche, número de ventas por vendedor, número de reparaciones o revisiones por mecánico, en que meses se hace un mayor número de ventas o revisiones, etc.

Todas las operaciones que se realizan en la base de datos deben quedar auditadas.

Identificación de roles

Los roles de este proyecto serán:

- Comercial. Realiza ventas de vehículos nuevos o de ocasión.
- Jefe Departamento Comercial. Es el encargado de realizar las compras de vehículos y de revisar el trabajo de los comerciales.
- Mecánico: Realiza las revisiones y reparaciones del taller
- Jefe de Mecánicos: Es el encargado de revisar las reparaciones y revisiones del taller, se encarga de gestionar los datos personales de los mecánicos.
- Operario de Marketing que estudia los informes estadísticos que obtiene del sistema para tomar las acciones de mercado oportunas.
- Cliente que compra o repara su vehículo.¹

¹ Este rol no participa directamente en el sistema pero sí se debe reflejar en los diagramas de flujo de información

Listado de requerimientos

Dividimos el proyecto en 3 grandes áreas funcionales y dentro de cada una de ellas definiremos sus propios requerimientos. En último lugar definimos los requerimientos técnicos comunes a todo el proyecto.²

Área comercial

El área comercial se ocupa principalmente a la venta de vehículos.

1. Se podrá añadir y modificar los datos de los empleados.
2. Para realizar una venta, es necesario haber registrado previamente al cliente.
3. Tanto los mecánicos como los comerciales pueden actualizar los datos de los clientes.
4. Los clientes podrán añadir cualquier característica extra, conociendo el precio de cada una de ellas.
5. El precio final de un vehículo, es la suma del precio del modelo más el precio de las características extra añadidas. A veces, al concesionario le interesa realizar algún tipo de descuento, en este caso el comercial que realiza la venta podrá añadir una línea adicional a la factura aplicando el descuento deseado.
6. Los comerciales disponen de un inventario de vehículos de segunda mano, donde podrán consultar todos los vehículos que están disponibles a la venta.
7. A la hora de añadir un vehículo al inventario, deben informarse el modelo y las características del mismo.

Área de taller

En esta área se centralizan las funcionalidades del taller.

8. Los clientes llevan el vehículo al taller para hacer la revisión, les atiende un mecánico, que es quien da de alta en el sistema la nueva revisión. El mecánico comprueba que el teléfono de contacto de cliente es correcto y si es necesario lo actualiza.
9. Una reparación o revisión del vehículo puede ser atendida por uno o más mecánicos. Cada mecánico debe registrar el número de horas que ha dedicado a cada reparación.
10. Cuando un cliente, cuyo coche no fue adquirido en el concesionario lleva el coche al taller para realizar una reparación, el mecánico que lo atiende da de alta una nueva reparación en el sistema y guarda los datos de contacto del cliente: nombre y teléfono y la matrícula del vehículo.
11. Una vez finalizada la reparación o revisión el mecánico debe registrar la fecha y hora.

² El documento de requerimientos se realiza a partir del contenido de este apartado.

Área Marketing

El área de Marketing se ocupa de estudiar los datos estadísticos que le proporcionará el Data WareHouse para realizar los siguientes estudios:

12. Uno de los análisis más importantes es el Estudio de coste/beneficio. Se calcula el beneficio total de uno o varios concesionarios en un período determinado. Para calcular el importe total de ventas, se suma el número de ventas realizadas y el importe total de las reparaciones. Para calcular el coste total se suman los sueldos de todos los empleados, el coste de las compras de vehículos y el coste de piezas y repuestos de las intervenciones llevadas a cabo en el taller. El beneficio es el total de ventas – el total de costes.
13. La productividad de los comerciales se mide gracias al informe de Total de ventas por empleado que puede obtenerse para cualquier rango de fechas
14. La productividad de los mecánicos se mide gracias al informe de Tiempo medio de reparación por tipo de reparación.
15. Anualmente se revisan cuales son los períodos en los que se realizan más revisiones y más ventas.
16. Se realiza un estudio de las ventas de los vehículos nuevos, mediante el Informe de ventas por tipo de vehículo

Requerimientos no funcionales

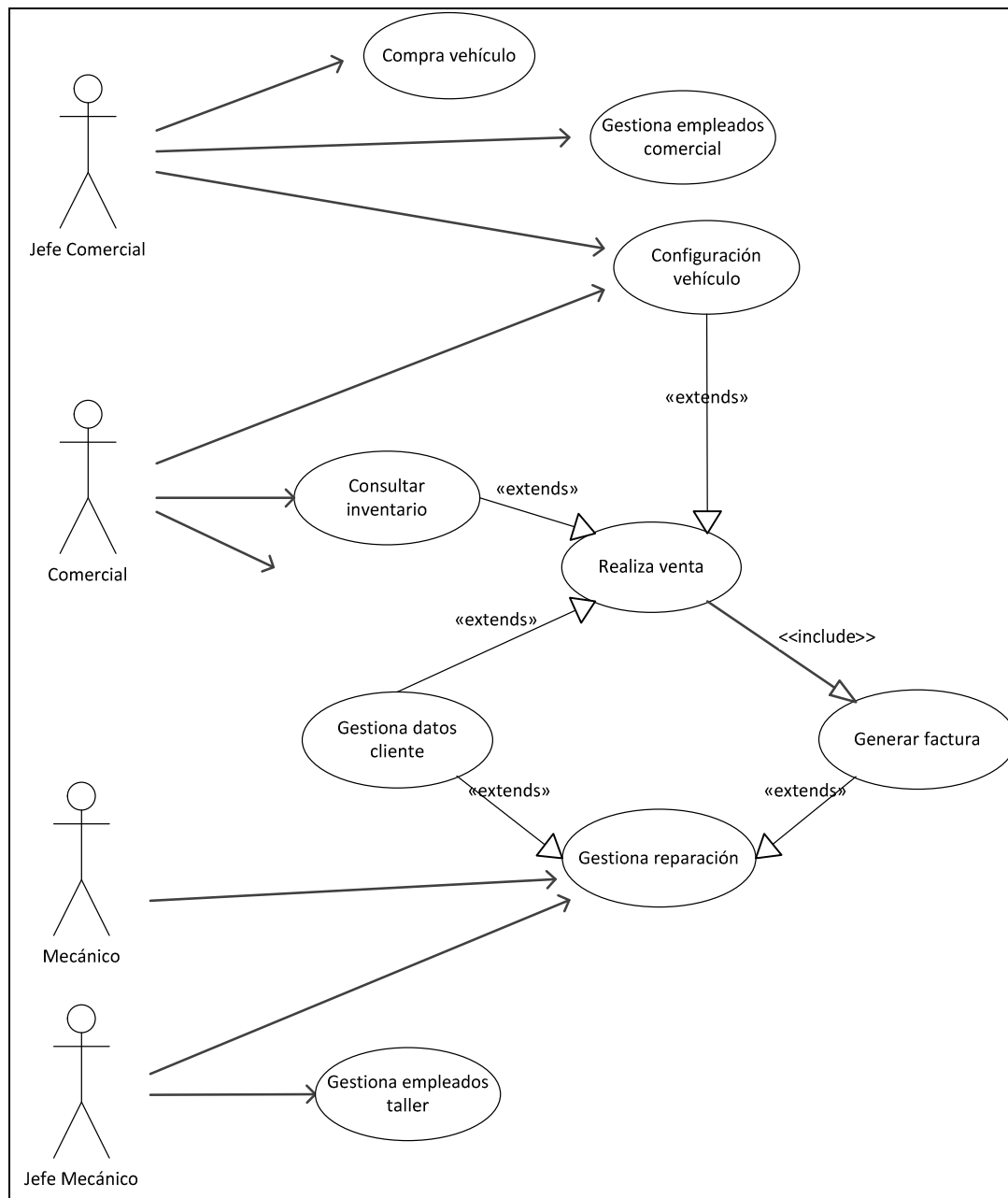
17. La base de datos debe desarrollarse en Oracle.
18. El sistema debe ser escalable
19. El sistema debe automatizar procesos que faciliten la usabilidad del sistema por ejemplo: en el taller es importante la búsqueda de vehículo por matrícula o la búsqueda de vehículo

Casos de uso

El diagrama de los casos de uso muestra las funcionalidades del sistema.

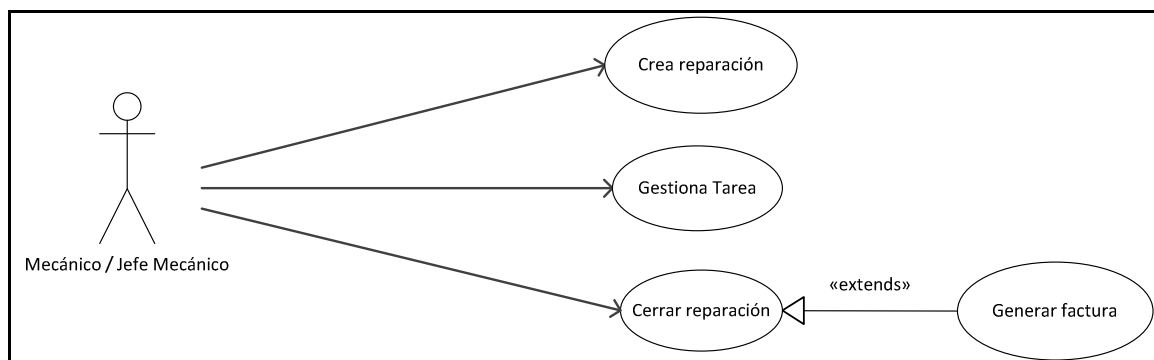
- El jefe del departamento comercial se encarga de dar de alta en el sistema a sus empleados, así como a realizar todos los cambios oportunos sobre sus datos.
- Al realizar una venta, los comerciales dan de alta en el sistema los datos del cliente, creando una nueva ficha de cliente. Si el cliente ya está registrado en nuestro sistema, se actualizarán sus datos si es oportuno.
- La prioridad de venta del personal del área comercial se centra en los coches de la propia marca UOC. A la hora de realizar una venta los comerciales disponen de todos los modelos disponibles. Para cada modelo, pueden consultar cuales son las características básicas que incluye.
- El jefe de comercial se encarga de realizar la compra de vehículos cuando se le presenta alguna oferta. En este momento debe añadir el vehículo al inventario informando la marca, el modelo, el precio, etc.

- Al realizar una venta, los comerciales evalúan la posibilidad de adquirir el vehículo del actual del cliente, si creen que pueden venderlo como coche de ocasión a un mayor precio. A cambio, le ofrecen al cliente un descuento en la compra de su nuevo vehículo marca UOC. En este caso, el jefe de comercial aprueba la compra y añade el vehículo al inventario.
- El jefe de mecánicos añade al sistema la información de sus mecánicos y la actualiza cuando es oportuno.



El caso de uso de Gestiona reparación se desglosa del siguiente modo:

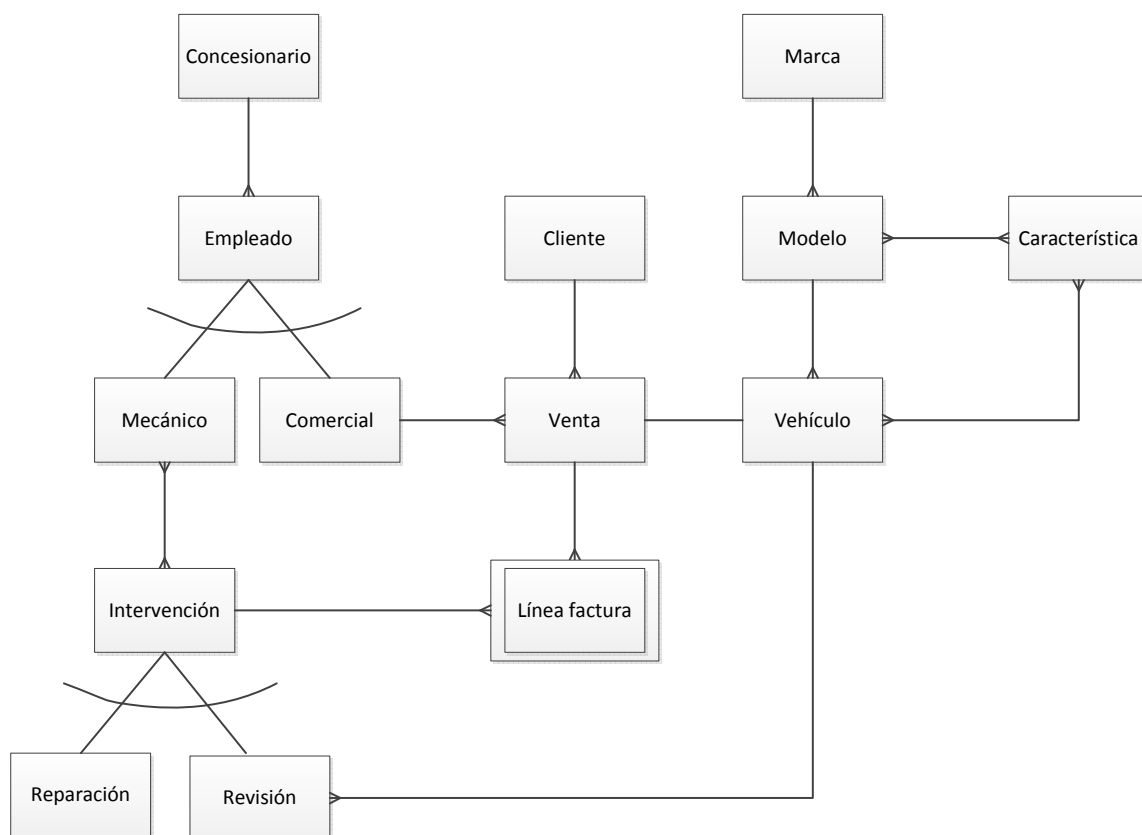
- Al crear una nueva reparación los mecánicos introducen los datos de contacto del cliente y el vehículo. También indican si es una revisión o una reparación.
- Una vez creada la reparación pueden ir añadiendo tareas indicando si es facturable o no.
- Cuando el trabajo ha finalizado, deben cerrar la reparación. En el caso de que exista alguna tarea facturable, se generará la factura.



Modelo lógico de datos

A partir de los requerimientos, creamos dos modelos lógicos de datos, el primero representa el sistema de información de los procesos que se llevan a cabo en los concesionarios. El segundo se centra en la información de Data WareHouse que se utilizará para sacar informes.

En el siguiente modelo lógico de datos, se representa la información que debemos modelar desde el punto de vista del negocio del concesionario.

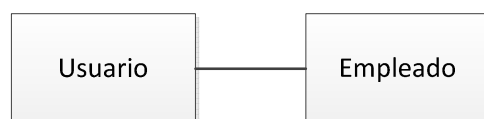


Describimos las entidades:

- **Concesionario:** Cada concesionario tiene sus propios empleados y por extensión sus vehículos vendidos y su taller propio.
- **Empleado:** Un empleado, puede ser o un *Mecánico* o un *Comercial*. Un atributo destacado es *jefe* que indica si el empleado del que estamos hablando es jefe o no.
- **Marca:** Una marca de vehículo tiene varios modelos. Se gestiona la marca UOC y otras marcas que se vendan como vehículos de ocasión.

- *Modelo*: Un modelo incluye un conjunto de características básicas. Además a un modelo se le pueden añadir características extra, cada una de ellas tendrá un precio diferente en cada modelo.
- *Vehículo*: Un vehículo pertenece a un modelo, por tanto incluye sus características básicas, pero además, un vehículo puede tener sus propias características, llamadas extra.
- *Cliente*: Modela los datos personales y de contacto de los clientes.
- *Venta*: Una venta de vehículo la realizará siempre un comercial. Puede darse el caso de que un vehículo se venda varias veces si el concesionario volviera a comprarlo.
- *Intervención*: Hace referencia a un trabajo realizado en el taller, que puede ser o bien una *revisión* de un vehículo o bien una *reparación*. Varios mecánicos puede trabajar en una misma intervención, deberán registrar el número de horas que dediquen a cada tarea y el coste y precio de la misma.
- *Revisión*: Los vehículos de marca UOC que se venden en el concesionario pueden pasar la revisión en nuestros concesionarios.
- *Reparación*: Se puede realizar una reparación de un vehículo que no se haya vendido en nuestro concesionario. Al departamento de Marketing no le interesa saber cuántas reparaciones se han hecho sobre un vehículo marca UOC.
- *Líneas de factura*: Esta entidad es una entidad débil porque depende de las entidades de *Venta* e *Intervención*. El conjunto de unas líneas de factura de una intervención o de una venta forman la factura final.

Los empleados interactúan con el sistema y son responsables de la información del mismo. Debemos implementar mecanismos para auditar todos los movimientos que se produzcan en la BD. Para ello creamos la entidad usuario, que estará ligada al empleado. Todos los empleados accederán al sistema usando su propio usuario y la auditoría mostrará las acciones realizadas por estos.



- *Usuario*: Usuario del sistema. Para la seguridad de acceso se guarda una contraseña.
- *Auditoría*: Log de las acciones realizadas.

A todas las tablas se añade la información de auditoría: usuario de alta y fecha de registro y de última modificación.

Data WareHouse

En cuanto al modelo sobre el Data WareHouse, se basará en dos tipos de almacenes de datos:

- Tabla de *Hechos*. Una tabla donde se obtiene toda la información detallada del sistema o de una parte del sistema, que se explota para obtener los datos analíticos.
- Tablas de *Dimensiones*. Tablas auxiliares con maestros de información. Son la base de información para la tabla de *Hechos*.

En este proyecto, debido a la pequeña envergadura, generaremos las siguientes tablas para la gestión de los datos de DWH

Se crean las siguientes entidades, todas ellas pueden considerarse entidades de dimensiones ya que a partir de todas se obtiene la información necesaria para la obtención de informes.

- Vehículo: Información de los vehículos y sus características.
- Cliente: Información general de los clientes.
- Empleado: Información general de empleados.
- Catálogo: Información general de la configuración de los vehículos. Catálogo de vehículos UOC.
- Comercial: Movimientos comerciales de compra/venta
- Facturación: Información de la facturación
- Reparaciones: Información sobre las tareas de las reparaciones.

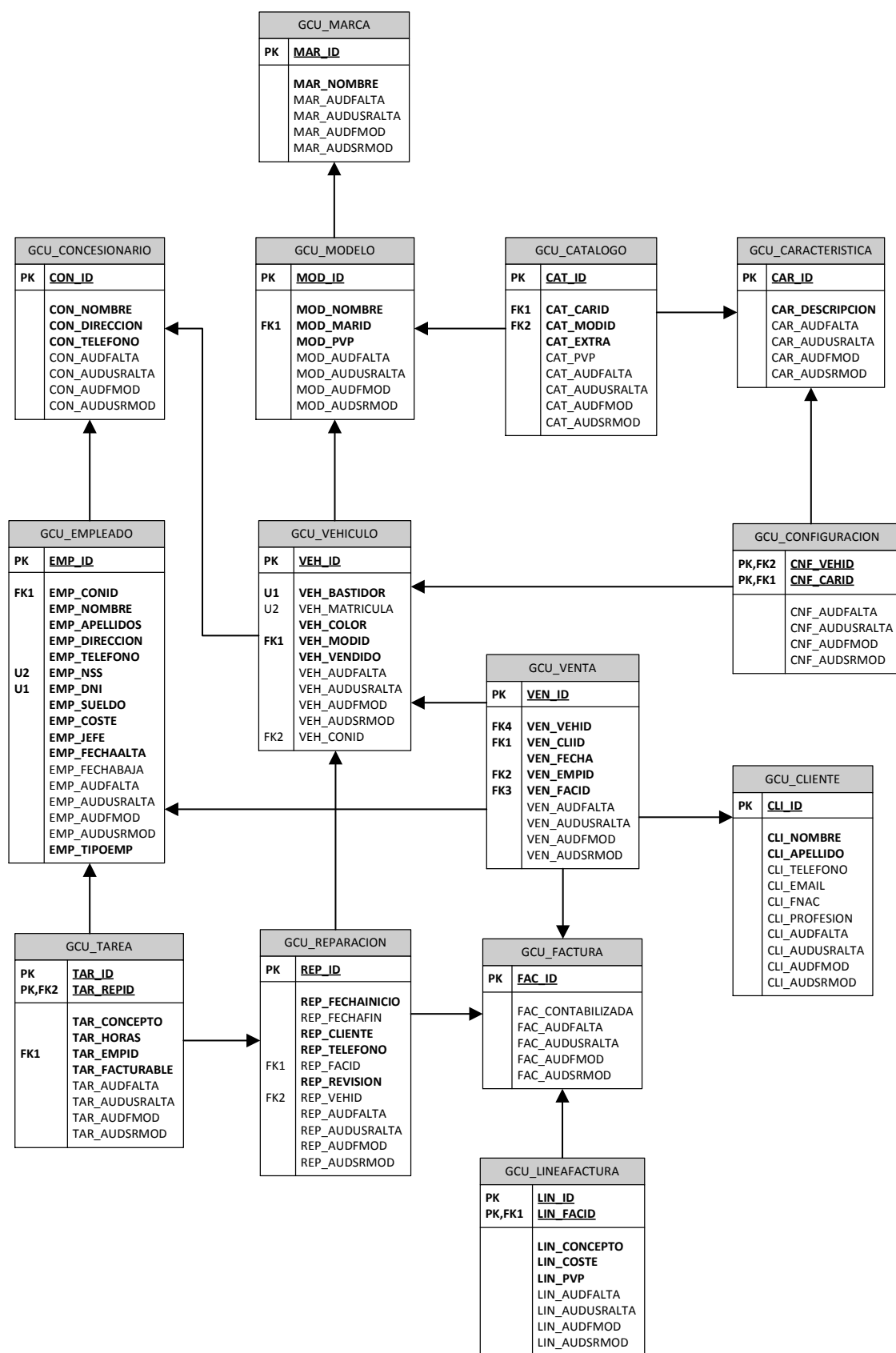
En base a todas estas tablas podemos realizar los siguientes informes:

- Estadísticas de clientes. Qué perfiles compran qué modelos de coche. Perfiles por edad o profesión. Apto para generación de campañas de marketing.
- Catálogo de modelos disponibles para realizar búsquedas.
- Estudio de ventas y compras de la empresa.
- Estudio de número de horas dedicadas por los empleados del taller
- Análisis coste/beneficio entre el coste que supone un empleado y el beneficio aportado (horas de reparaciones facturables y ventas)
- Información estadística de empleados para previsiones de futuras primas.
- Información del inventario de vehículos disponibles.
- Estudio de facturación.

Modelo relacional

Después de normalizar el modelo lógico de datos, obtenemos el modelo relacional, que será el que importaremos en la base de datos. Mostramos el modelo relacional eliminando las tablas de control de auditoría (usuario y auditoría) para mejorar la legibilidad del diagrama.

En el Anexo A – Documentos, la [ilustración A.1](#), muestra el modelo relacional completo.



Capítulo 3: Diseño

Definición de estándares

Para facilitar las tareas de implementación, se definen estándares para crear la definición de los objetos.

Para construir la base de datos, aplicaremos las siguientes nomenclaturas.

1. Nombre de las tablas: *Área* + [*DHW*] + *Nombre*
 - *Área*: Puede tomar los siguientes valores:
 - 'GCU' (Gestión Concesionario UOC),
 - 'DWH' (Data WareHouse)
 - *DWH*: Si la tabla es de DWH debe identificar el tipo de tabla:
 - 'CON': Consolidada. Tabla de hechos.
 - 'DIM': Tabla de dimensiones.
 - *Nombre*: Breve descripción de tabla
2. Nombre de las vistas 'VW' + *Área* + *Nombre*
 - *Área*: Con los siguientes valores:
 - 'GCU' (Gestión Concesionario UOC),
 - 'DWH' (Data WareHouse)
 - *Nombre*: Breve descripción de la vista
3. Nombre de las secuencias: 'SEQ' + *Nombre corto de tabla* + *Descripción*
 - *Nombre corto de tabla*: Abreviatura tabla (3 caracteres)
 - *Descripción*: Descripción que identifique la secuencia
4. Nombre de disparadores: 'TRG' + '*Nombre corto de tabla*' + *Nombre*
 - *Nombre corto de tabla*: Abreviatura de la tabla (3 caracteres)
 - *Nombre*: Descripción que defina que operación realiza el trigger
5. Nombre de primary keys: *Nombre corto de tabla* + 'PK'
 - *Nombre corto de tabla*: Abreviatura de la tabla (3 caracteres)
6. Nombre de constraints: *Tipo constraint* + *Nombre corto de tabla* + [*Descripción*]
 - *Tipo constraint*: Puede tomar los siguientes valores:
 - 'FK': Foreign Key
 - 'CH': Check Constraint
 - UQ: Unique Key
 - *Nombre corto de tabla*: Abreviatura tabla (3 caracteres)
 - *Descripción*: Optativo
7. Nombre de packages: 'PKG' + *Área* + *Descripción*
 - *Área*: Puede tomar los siguientes valores:
 - 'GCU' (Gestión Concesionario UOC),
 - 'DWH' (Data WareHouse)
 - *Descripción*: Nombre del package
8. Nombre de funciones: 'FUN' + *Área* + *Descripción*
 - *Área*: Puede tomar los siguientes valores:
 - 'GCU' (Gestión Concesionario UOC),

- 'DWH' (Data WareHouse)
- *Descripción:* Nombre de la función
- 9. Nombre de procedures: 'PRC' + *Área* + *Descripción*
 - *Área:* Puede tomar los siguientes valores:
 - 'GCU' (Gestión Concesionario UOC),
 - 'DWH' (Data WareHouse)
 - *Descripción:* Nombre del procedimiento.
- 10. Nombre de los campos de las tablas: *Nombre corto de tabla* + *Nombre atributo*
 - *Nombre corto de tabla:* Abreviatura tabla (3 caracteres)
 - *Nombre atributo:* Nombre del atributo sobre el que se aplica la secuencia
- 11. Nombre de los índices: 'IND' + *Nombre corto de tabla* + *Número*
 - *Nombre corto de tabla:* Abreviatura tabla (3 caracteres)
 - *Número:* Número correlativo del índice.

Buenas prácticas

Además de las nomenclaturas de uso obligatorio, es importante cumplir con las siguientes buenas prácticas:

- Agregar un comentario a todas las tablas y a todos los atributos de las tablas.
- Añadir comentarios al código fuente de las funciones y procedimientos, indicando:
 - Comentario inicial. Se añade al comienzo de la función y se describe el funcionamiento general de la aplicación, y los parámetros de retorno de la función.
 - Comentarios detallados cuando sea necesario explicar un bloque de código.
- Evaluar el coste computacional de las sentencias SQL realizadas para optimizar el código y obtener mejor rendimiento.
- Usar el control de excepciones.
- Reutilizar código fuente.

Modelo físico de datos

Decisiones de diseño

Podría dividirse la creación de objetos en dos esquemas de base de datos, uno de ellos dedicado a la gestión del concesionario y el otro con los objetos propios de Data WareHouse. En este caso, simplemente hay que añadir los permisos necesarios entre esquemas para la correcta visibilidad de los datos. Finalmente, se decide crear todos los objetos necesarios, en un sólo esquema de base de datos para simplificar la exportación de la BD.

En cuanto al diseño sobre el Data WareHouse, después de realizar un estudio detallado sobre como modelar este tipo de sistemas, he seguido las siguientes pautas.

- Todos los objetos de DataWareHouse se identifican con la abreviación 'DWH'.
- Los objetos básicos de información son las vistas.
- Se crea una vista para cada entidad de *Dimensión*
- Se crea una vista para la entidad de *Hechos*

Modelo físico de datos

A continuación presentamos el modelo físico de datos, con la información detallada de las tablas y vistas.

Tablas

Table	Primary Key
UOC.GCU_AUDITORIA	AUD_ID
UOC.GCU_CARACTERISTICA	CAR_ID
UOC.GCU_CATALOGO	CAT_ID
UOC.GCU_CLIENTE	CLI_ID
UOC.GCU_CONCESIONARIO	CON_ID
UOC.GCU_CONFIGURACION	CNF_VEHID, CNF_CARID
UOC.GCU_EMPLEADO	EMP_ID
UOC.GCU_FACTURA	FAC_ID
UOC.GCU_LINEAFACTURA	LIN_ID, LIN_FACID
UOC.GCU_MARCA	MAR_ID
UOC.GCU_MODELO	MOD_ID
UOC.GCU_REPARACION	REP_ID
UOC.GCU_TAREA	TAR_ID, TAR_REPID
UOC.GCU_USUARIO	USR_ID
UOC.GCU_VEHICULO	VEH_ID
UOC.GCU_VENTA	VEN_ID

Table 'UOC.GCU_AUDITORIA'

Table	GCU_AUDITORIA
Primary Key	GCU_AUDITORIA_PK

Columns

Key	Column	Datatype	Not Null
PK	AUD_ID	NUMBER (9)	True
	AUD_TABLA	VARCHAR2 (20 Byte)	True
	AUD_ACCION	VARCHAR2 (200 Byte)	True
FK	AUD_USRID	NUMBER (9)	True
	AUD_FECHA	DATE	True

Foreign Key Constraints

Name	Type	Parent	Child
FK_AUD_USRID	Non Identifying	UOC.GCU_USUARIO	UOC.GCU_AUDITORIA

Table 'UOC.GCU_CARACTERISTICA'

Table	GCU_CARACTERISTICA
Primary Key	GCU_CARACTERISTICA_PK

Columns

Key	Column	Datatype	Not Null
PK	CAR_ID	NUMBER (9)	True
	CAR_DESCRIPCION	VARCHAR2 (100 Byte)	True
	CAR_AUDFALTA	DATE	False
FK	CAR_AUDUSRALTA	NUMBER (9)	False
	CAR_AUDFMOD	DATE	False
FK	CAR_AUDUSRMOD	NUMBER (9)	False

Foreign Key Constraints

Name	Type	Parent	Child
FK_CAT_CARID	Non Identifying	UOC.GCU_CARACTERISTICA	UOC.GCU_CATALOGO
FK_CAR_AUD2	Non Identifying	UOC.GCU_USUARIO	UOC.GCU_CARACTERISTICA
FK_CNF_CARID	Identifying	UOC.GCU_CARACTERISTICA	UOC.GCU_CONFIGURACION
FK_CAR_AUD1	Non Identifying	UOC.GCU_USUARIO	UOC.GCU_CARACTERISTICA

Table 'UOC.GCU_CATALOGO'

Table	GCU_CATALOGO
Primary Key	GCU_CATALOGO_PK

Columns

Key	Column	Datatype	Not Null
PK	CAT_ID	NUMBER (9)	True
FK	CAT_CARID	NUMBER (9)	True
FK	CAT_MODID	NUMBER (9)	True
	CAT_EXTRA	CHAR (1 Byte)	True
	CAT_PVP	NUMBER (15,3)	False
FK	CAT_AUDFALTA	DATE	False
	CAT_AUDUSRALTA	NUMBER (9)	False
	CAT_AUDFMOD	DATE	False
FK	CAT_AUDUSRMOD	NUMBER (9)	False

Foreign Key Constraints

Name	Type	Parent	Child
FK_CAT_CARID	Non Identifying	UOC.GCU_CARACTERISTICA	UOC.GCU_CATALOGO
FK_CAT_AUD2	Non Identifying	UOC.GCU_USUARIO	UOC.GCU_CATALOGO
FK_CAT_AUD1	Non Identifying	UOC.GCU_USUARIO	UOC.GCU_CATALOGO
FK_CAT_MODID	Non Identifying	UOC.GCU_MODELO	UOC.GCU_CATALOGO

Table 'UOC.GCU_CLIENTE'

Table	GCU_CLIENTE
Primary Key	GCU_CLIENTE_PK

Columns

Key	Column	Datatype	Not Null
PK	CLI_ID	NUMBER (9)	True
	CLI_NOMBRE	VARCHAR2 (20 Byte)	True
	CLI_APELLIDO	VARCHAR2 (50 Byte)	True
	CLI_TELEFONO	VARCHAR2 (15 Byte)	False
	CLI_EMAIL	VARCHAR2 (30 Byte)	False
	CLI_FNAC	DATE	False
	CLI_PROFESION	VARCHAR2 (20 Byte)	False
	CLI_AUDFALTA	DATE	False
FK	CLI_AUDUSRALTA	NUMBER (9)	False
	CLI_AUDFMOD	DATE	False
FK	CLI_AUDUSRMOD	NUMBER (9)	False

Foreign Key Constraints

Name	Type	Parent	Child
FK_CLI_AUD1	Non Identifying	UOC.GCU_USUARIO	UOC.GCU_CLIENTE
FK_VEN_CLIID	Non Identifying	UOC.GCU_CLIENTE	UOC.GCU_VENTA
FK_CLI_AUD2	Non Identifying	UOC.GCU_USUARIO	UOC.GCU_CLIENTE

Table 'UOC.GCU_CONCESIONARIO'

Table	GCU_CONCESIONARIO
Primary Key	GCU_CONCESIONARIO_PK

Columns

Key	Column	Datatype	Not Null
PK	CON_ID	NUMBER (9)	True
	CON_NOMBRE	VARCHAR2 (50 Byte)	True
	CON_DIRECCION	VARCHAR2 (100 Byte)	True
	CON_TELEFONO	VARCHAR2 (15 Byte)	True
	CON_AUDFALTA	DATE	False
FK	CON_AUDUSRALTA	NUMBER (9)	False
	CON_AUDFMOD	DATE	False
FK	CON_AUDUSRMOD	NUMBER (9)	False

Foreign Key Constraints

Name	Type	Parent	Child
FK_CON_AUD2	Non Identifying	UOC.GCU_USUARIO	UOC.GCU_CONCESIONARIO
FK_CON_AUD1	Non Identifying	UOC.GCU_USUARIO	UOC.GCU_CONCESIONARIO

Table 'UOC.GCU_CONFIGURACION'

Table	GCU_CONFIGURACION
Primary Key	GCU_CONFIGURACION_PK

Columns

Key	Column	Datatype	Not Null
PFK	CNF_VEHID	NUMBER (9)	True
PFK	CNF_CARID	NUMBER (9)	True
	CNF_AUDFALTA	DATE	False
FK	CNF_AUDUSRALTA	NUMBER (9)	False
	CNF_AUDFMOD	DATE	False
FK	CNF_AUDUSRMOD	NUMBER (9)	False

Foreign Key Constraints

Name	Type	Parent	Child
FK_CNF_CARID	Identifying	UOC.GCU_CARACTERISTICA	UOC.GCU_CONFIGURACION
FK_CNF_AUD1	Non Identifying	UOC.GCU_USUARIO	UOC.GCU_CONFIGURACION
FK_CNF_AUD2	Non Identifying	UOC.GCU_USUARIO	UOC.GCU_CONFIGURACION
FK_CNF_VEHID	Identifying	UOC.GCU_VEHICULO	UOC.GCU_CONFIGURACION

Table 'UOC.GCU_EMPLEADO'

Table	GCU_EMPLEADO
Primary Key	GCU_CONCESIONARIO_PK

Columns

Key	Column	Datatype	Not Null
PK	EMP_ID	NUMBER (9)	True
	EMP_CONID	NUMBER (9)	True
	EMP_NOMBRE	VARCHAR2 (25 Byte)	True
	EMP_APELLIDOS	VARCHAR2 (50 Byte)	True
	EMP_DIRECCION	VARCHAR2 (100 Byte)	True
	EMP_TELEFONO	VARCHAR2 (15 Byte)	True
	EMP_NSS	VARCHAR2 (10 Byte)	True
	EMP_DNI	VARCHAR2 (15 Byte)	True
	EMP_SUELDO	NUMBER (15,3)	True
	EMP_COSTE	NUMBER (15,3)	True
	EMP_JEFE	NUMBER (1)	True
	EMP_FECHAALTA	DATE	True
	EMP_FECHABAJA	DATE	False
	EMP_AUDFALTA	DATE	False
FK	EMP_AUDUSRALTA	NUMBER (9)	False
	EMP_AUDFMOD	DATE	False
FK	EMP_AUDUSRMOD	NUMBER (9)	False
	EMP_TIPOEMP	CHAR (1 Byte)	True

Foreign Key Constraints

Name	Type	Parent	Child
FK_TAR_EMPID	Non Identifying	UOC.GCU_EMPLEADO	UOC.GCU_TAREA
FK_EMP_AUD1	Non Identifying	UOC.GCU_USUARIO	UOC.GCU_EMPLEADO
FK_USR_EMPID	Non Identifying	UOC.GCU_EMPLEADO	UOC.GCU_USUARIO
FK_EMP_AUD2	Non Identifying	UOC.GCU_USUARIO	UOC.GCU_EMPLEADO
FK_VEN_EMPID	Non Identifying	UOC.GCU_EMPLEADO	UOC.GCU_VENTA

Table 'UOC.GCU_FACTURA'

Table	GCU_FACTURA
Primary Key	SYS_C007143

Columns

Key	Column	Datatype	Not Null
PK	FAC_ID	NUMBER (9)	True
	FAC_CONTABILIZADA	CHAR (1 Byte)	False
	FAC_AUDFALTA	DATE	False
FK	FAC_AUDUSRALTA	NUMBER (9)	False
	FAC_AUDFMOD	DATE	False
FK	FAC_AUDUSRMOD	NUMBER (9)	False

Foreign Key Constraints

Name	Type	Parent	Child
FK_LIN_FACID	Identifying	UOC.GCU_FACTURA	UOC.GCU_LINEAFACTURA
FK_VEN_FACID	Non Identifying	UOC.GCU_FACTURA	UOC.GCU_VENTA
FK_REP_FACID	Non Identifying	UOC.GCU_FACTURA	UOC.GCU_REPARACION
FK_FAC_AUD1	Non Identifying	UOC.GCU_USUARIO	UOC.GCU_FACTURA
FK_FAC_AUD2	Non Identifying	UOC.GCU_USUARIO	UOC.GCU_FACTURA

Table 'UOC.GCU_LINEAFACTURA'

Table	GCU_LINEAFACTURA
Primary Key	GCU_LINEAFACTURA_PK

Columns

Key	Column	Datatype	Not Null
PK	LIN_ID	NUMBER (9)	True
PFK	LIN_FACID	NUMBER (9)	True
	LIN_CONCEPTO	VARCHAR2 (100 Byte)	True
	LIN_COSTE	NUMBER (15,3)	True
	LIN_PVP	NUMBER (15,3)	True
	LIN_AUDFALTA	DATE	False
FK	LIN_AUDUSRALTA	NUMBER (9)	False
	LIN_AUDFMOD	DATE	False
FK	LIN_AUDUSRMOD	NUMBER (9)	False

Foreign Key Constraints

Name	Type	Parent	Child
FK_LIN_FACID	Identifying	UOC.GCU_FACTURA	UOC.GCU_LINEAFACTURA
FK_LIN_AUD1	Non Identifying	UOC.GCU_USUARIO	UOC.GCU_LINEAFACTURA
FK_LIN_AUD2	Non Identifying	UOC.GCU_USUARIO	UOC.GCU_LINEAFACTURA

Table 'UOC.GCU_MARCA'

Table	GCU_MARCA
Primary Key	GCU_MARCA_PK

Columns

Key	Column	Datatype	Not Null
PK	MAR_ID	NUMBER (9)	True
	MAR_NOMBRE	VARCHAR2 (20 Byte)	True
	MAR_AUDFALTA	DATE	False
FK	MAR_AUDUSRALTA	NUMBER (9)	False
	MAR_AUDFMOD	DATE	False
FK	MAR_AUDUSRMOD	NUMBER (9)	False

Foreign Key Constraints

Name	Type	Parent	Child
FK_MAR_AUD1	Non Identifying	UOC.GCU_USUARIO	UOC.GCU_MARCA
FK_MOD_MARID	Non Identifying	UOC.GCU_MARCA	UOC.GCU_MODELO
FK_MAR_AUD2	Non Identifying	UOC.GCU_USUARIO	UOC.GCU_MARCA

Table 'UOC.GCU_MODELO'

Table	GCU_MODELO
Primary Key	GCU_MODELO_PK

Columns

Key	Column	Datatype	Not Null
PK	MOD_ID	NUMBER (9)	True
	MOD_NOMBRE	VARCHAR2 (50 Byte)	True
FK	MOD_MARID	NUMBER (9)	True
	MOD_PVP	NUMBER (15,3)	True
	MOD_AUDFALTA	DATE	False
FK	MOD_AUDUSRALTA	NUMBER (9)	False
	MOD_AUDFMOD	DATE	False
FK	MOD_AUDUSRMOD	NUMBER (9)	False

Foreign Key Constraints

Name	Type	Parent	Child
FK_MOD_MARID	Non Identifying	UOC.GCU_MARCA	UOC.GCU_MODELO
FK_MOD_AUD2	Non Identifying	UOC.GCU_USUARIO	UOC.GCU_MODELO
FK_MOD_AUD1	Non Identifying	UOC.GCU_USUARIO	UOC.GCU_MODELO
FK_CAT_MODID	Non Identifying	UOC.GCU_MODELO	UOC.GCU_CATALOGO
FK_VEH_MODID	Non Identifying	UOC.GCU_MODELO	UOC.GCU_VEHICULO

Table 'UOC.GCU_REPARACION'

Table	GCU_REPARACION
Primary Key	GCU_REPARACION_PK

Columns

Key	Column	Datatype	Not Null
PK	REP_ID	NUMBER (9)	True
	REP_FECHAINICIO	DATE	True
	REP_FECHAFIN	DATE	False
	REP_CLIENTE	VARCHAR2 (20 Byte)	True
	REP_TELEFONO	VARCHAR2 (15 Byte)	True
FK	REP_FACID	NUMBER (9)	False
	REP_REVISION	CHAR (1 Byte)	True
FK	REP_VEHID	NUMBER (9)	False
	REP_AUDFALTA	DATE	False
FK	REP_AUDUSRALTA	NUMBER (9)	False
	REP_AUDFMOD	DATE	False
FK	REP_AUDUSRMOD	NUMBER (9)	False

Foreign Key Constraints

Name	Type	Parent	Child
FK_REP_FACID	Non Identifying	UOC.GCU_FACTURA	UOC.GCU_REPARACION
FK_TAR_REPID	Identifying	UOC.GCU_REPARACION	UOC.GCU_TAREA
FK_REP_AUD2	Non Identifying	UOC.GCU_USUARIO	UOC.GCU_REPARACION
FK_REP_AUD1	Non Identifying	UOC.GCU_USUARIO	UOC.GCU_REPARACION
FK_REP_VEHID	Non Identifying	UOC.GCU_VEHICULO	UOC.GCU_REPARACION

Table 'UOC.GCU_USUARIO'

Table	GCU_USUARIO
Primary Key	GCU_USUARIO_PK

Columns

Key	Column	Datatype	Not Null
PK	USR_ID	NUMBER (9)	True
	USR_NOMUSR	VARCHAR2(10 Byte)	True
	USR_PASS	VARCHAR2 (20 Byte)	True
	USR_EMPID	NUMBER (9)	False

Foreign Key Constraints

Name	Type	Parent	Child
FK_CAR_AUD2	Non Identifying	UOC.GCU_USUARIO	UOC.GCU_CARACTERISTICA
FK_CAR_AUD1	Non Identifying	UOC.GCU_USUARIO	UOC.GCU_CARACTERISTICA
FK_FAC_AUD1	Non Identifying	UOC.GCU_USUARIO	UOC.GCU_FACTURA
FK_FAC_AUD2	Non Identifying	UOC.GCU_USUARIO	UOC.GCU_FACTURA
FK_REP_AUD2	Non Identifying	UOC.GCU_USUARIO	UOC.GCU_REPARACION
FK_REP_AUD1	Non Identifying	UOC.GCU_USUARIO	UOC.GCU_REPARACION
FK_CLI_AUD1	Non Identifying	UOC.GCU_USUARIO	UOC.GCU_CLIENTE
FK_CLI_AUD2	Non Identifying	UOC.GCU_USUARIO	UOC.GCU_CLIENTE
FK_MAR_AUD1	Non Identifying	UOC.GCU_USUARIO	UOC.GCU_MARCA
FK_MAR_AUD2	Non Identifying	UOC.GCU_USUARIO	UOC.GCU_MARCA
FK_TAR_AUD2	Non Identifying	UOC.GCU_USUARIO	UOC.GCU_TAREA
FK_TAR_AUD1	Non Identifying	UOC.GCU_USUARIO	UOC.GCU_TAREA
FK_EMP_AUD1	Non Identifying	UOC.GCU_USUARIO	UOC.GCU_EMPLEADO
FK_CNF_AUD1	Non Identifying	UOC.GCU_USUARIO	UOC.GCU_CONFIGURACION
FK_CON_AUD2	Non Identifying	UOC.GCU_USUARIO	UOC.GCU_CONCESIONARIO
FK_MOD_AUD2	Non Identifying	UOC.GCU_USUARIO	UOC.GCU_MODELO
FK_USR_EMPID	Non Identifying	UOC.GCU_EMPLEADO	UOC.GCU_USUARIO
FK_VEN_AUD2	Non Identifying	UOC.GCU_USUARIO	UOC.GCU_VENTA
FK_CAT_AUD2	Non Identifying	UOC.GCU_USUARIO	UOC.GCU_CATALOGO
FK_CNF_AUD2	Non Identifying	UOC.GCU_USUARIO	UOC.GCU_CONFIGURACION
FK_CAT_AUD1	Non Identifying	UOC.GCU_USUARIO	UOC.GCU_CATALOGO
FK_MOD_AUD1	Non Identifying	UOC.GCU_USUARIO	UOC.GCU_MODELO
FK_VEH_AUD2	Non Identifying	UOC.GCU_USUARIO	UOC.GCU_VEHICULO
FK_VEN_AUD1	Non Identifying	UOC.GCU_USUARIO	UOC.GCU_VENTA
FK_EMP_AUD2	Non Identifying	UOC.GCU_USUARIO	UOC.GCU_EMPLEADO
FK_VEH_AUD1	Non Identifying	UOC.GCU_USUARIO	UOC.GCU_VEHICULO
FK_LIN_AUD1	Non Identifying	UOC.GCU_USUARIO	UOC.GCU_LINEAFACTURA
FK_AUD_USRID	Non Identifying	UOC.GCU_USUARIO	UOC.GCU_AUDITORIA
FK_CON_AUD1	Non Identifying	UOC.GCU_USUARIO	UOC.GCU_CONCESIONARIO
FK_LIN_AUD2	Non Identifying	UOC.GCU_USUARIO	UOC.GCU_LINEAFACTURA

Table 'UOC.GCU_TAREA'

Table	GCU_TAREA
Primary Key	GCU_TAREA_PK

Columns

Key	Column	Datatype	Not Null
PK	TAR_ID	NUMBER (9)	True
PFK	TAR_REPID	NUMBER (9)	True
	TAR_CONCEPTO	VARCHAR2 (100 Byte)	True
	TAR_HORAS	NUMBER (9,2)	True
FK	TAR_EMPID	NUMBER (9)	True
	TAR_FACTURABLE	CHAR (1 Byte)	True
	TAR_AUDFALTA	DATE	False
FK	TAR_AUDUSRALTA	NUMBER (9)	False
	TAR_AUDFMOD	DATE	False
FK	TAR_AUDUSRMOD	NUMBER (9)	False

Foreign Key Constraints

Name	Type	Parent	Child
FK_TAR_REPID	Identifying	UOC.GCU_REPARACION	UOC.GCU_TAREA
FK_TAR_EMPID	Non Identifying	UOC.GCU_EMPLEADO	UOC.GCU_TAREA
FK_TAR_AUD2	Non Identifying	UOC.GCU_USUARIO	UOC.GCU_TAREA
FK_TAR_AUD1	Non Identifying	UOC.GCU_USUARIO	UOC.GCU_TAREA

Table 'UOC.GCU_VEHICULO'

Table	GCU_VEHICULO
Primary Key	GCU_VEHICULO_PK

Columns

Key	Column	Datatype	Not Null
PK	VEH_ID	NUMBER (9)	True
	VEH_BASTIDOR	VARCHAR2 (15 Byte)	True
	VEH_MATRICULA	VARCHAR2 (10 Byte)	False
	VEH_COLOR	VARCHAR2 (15 Byte)	True
FK	VEH_MODID	NUMBER (9)	True
	VEH_VENDIDO	CHAR (1 Byte)	True
	VEH_AUDFALTA	DATE	False
FK	VEH_AUDUSRALTA	NUMBER (9)	False
	VEH_AUDFMOD	DATE	False
FK	VEH_AUDUSRMOD	NUMBER (9)	False

Foreign Key Constraints

Name	Type	Parent	Child
FK_REP_VEHID	Non Identifying	UOC.GCU_VEHICULO	UOC.GCU_REPARACION
FK_VEH_AUD2	Non Identifying	UOC.GCU_USUARIO	UOC.GCU_VEHICULO
FK_VEH_AUD1	Non Identifying	UOC.GCU_USUARIO	UOC.GCU_VEHICULO
FK_CNF_VEHID	Identifying	UOC.GCU_VEHICULO	UOC.GCU_CONFIGURACION
FK_VEH_MODID	Non Identifying	UOC.GCU_MODELO	UOC.GCU_VEHICULO
FK_VEN_VEHID	Non Identifying	UOC.GCU_VEHICULO	UOC.GCU_VENTA

Table 'UOC.GCU_VENTA'

Table	GCU_VENTA
Primary Key	GCU_VENTA_PK

Columns

Key	Column	Datatype	Not Null
PK	VEN_ID	NUMBER (9)	True
FK	VEN_VEHID	NUMBER (9)	True
FK	VEN_CLIID	NUMBER	True
	VEN_FECHA	DATE	True
FK	VEN_EMPID	NUMBER	True
FK	VEN_FACID	NUMBER	True
	VEN_AUDFALTA	DATE	False
FK	VEN_AUDUSRALTA	NUMBER (9)	False
	VEN_AUDFMOD	DATE	False
FK	VEN_AUDUSRMOD	NUMBER (9)	False

Foreign Key Constraints

Name	Type	Parent	Child
FK_VEN_FACID	Non Identifying	UOC.GCU_FACTURA	UOC.GCU_VENTA
FK_VEN_CLIID	Non Identifying	UOC.GCU_CLIENTE	UOC.GCU_VENTA
FK_VEN_AUD2	Non Identifying	UOC.GCU_USUARIO	UOC.GCU_VENTA
FK_VEN_AUD1	Non Identifying	UOC.GCU_USUARIO	UOC.GCU_VENTA
FK_VEN_EMPID	Non Identifying	UOC.GCU_EMPLEADO	UOC.GCU_VENTA
FK_VEN_VEHID	Non Identifying	UOC.GCU_VEHICULO	UOC.GCU_VENTA

Capítulo 4: Implementación

Introducción

En la fase de implementación se crearán todos los objetos de base de datos necesarios para poder llevar a cabo las funcionalidades que dicta el análisis.

La primera decisión tomada es crear un esquema de base de datos llamado UOC, donde almacenaremos todos los objetos necesarios. En nuestro esquema de base de datos crearemos:

- Tablas
- Vistas
- Triggers: para gestionar correctamente la auditoría de información.
- Funciones: para reutilizar código fuente.
- Packages: para agrupar el código según el área.
- Jobs: para lanzar los procesos automáticos de DataWareHouse.

Buenas prácticas

En la implementación de código fuente se han llevado a cabo las siguientes buenas prácticas:

- Todos los procedimientos y funciones deben estar correctamente comentados para facilitar la comprensión del código PL/SQL.
- Todas las variables utilizadas para añadir o modificar un registro de una tabla, deben definirse del siguiente modo:

*Var tabla.attribute%**Type***

De este modo estamos creando una variable llamada *Var* cuyo tipo de datos es el mismo que el *attribute* de la *tabla*

- Las variables que almacenen el resultado de un cursor debe declararse del siguiente modo:

*Var nombre_cursor%**RowType***

Así podemos realizar cualquier modificación sobre el cursor *nombre_cursor* sin tener que modificar la variable.

- Siempre que realicemos una inserción en una tabla, debemos indicar los atributos que se están insertando. En Oracle, existe la opción de realizar un

una inserción de un registro especificando únicamente los valores de los atributos, que se insertarán en la tabla por orden de ID atributo.

- Todos los errores se gestionarán con el control de excepciones instalado en las funciones y packages. Se crea la tabla SIS_ERRORES para guardar la información de los errores del sistema.

Código	Error
20001	Error al crear concesionario
20002	Error al modificar concesionario
20003	Error al eliminar concesionario
20004	Error al crear marca
20005	Error al modificar marca
20006	Error al eliminar marca
20007	Error al crear modelo
20008	Error al modificar modelo
20009	Error al eliminar modelo
20010	Error al crear característica
20011	Error al modificar el característica
20012	Error al eliminar la característica
20013	Error al crear catálogo
20014	Error al modificar el catálogo
20015	Error al eliminar el catálogo
20016	Error al crear el empleado
20017	Error al modificar el empleado
20018	Error al eliminar el empleado
20019	Error al crear la venta
20020	Error al generar líneas de factura
20021	Error al facturar la venta
20022	Error al facturar la reparación
20023	Error al crear vehículo
20024	Error al modificar el vehículo
20025	Error al eliminar el vehículo
20026	Error al marcar el vehículo como vendido
20027	Característica repetida
20028	Característica no válida
20029	Error al eliminar la característica
20030	Error al crear reparación
20031	Error al crear la tarea
20032	Error al modificar la tarea
20033	Error al eliminar la tarea
20034	Error al cerrar la facturación

- El inventario de vehículos se gestiona a partir de una vista que mostrará todos los vehículos no vendidos y que por tanto están disponibles.

Decisiones de Diseño

Con el fin de comprender el funcionamiento de la base de datos implementada, se definen las decisiones de diseño más importantes.

1. La tabla que guarda la información de los usuarios, *GCU_USUARIO* tiene un atributo llamado *USR_PASS* que guarda la contraseña del usuario. Por seguridad hemos decidido encriptar esta contraseña. Para ello creamos la función *FUN_ENCRIPTA* que aplica el algoritmo de encriptación MD5 al parámetro de entrada y devuelve la clave encriptada resultante. Cabe destacar que para la realización de esta función se ha utilizado el package del sistema *DBMS_OBFUSCATION_TOOLKIT* y por tanto el script de creación de usuario encriptado tiene incluidas las sentencias *GRANT* necesarias para dar permisos sobre este objeto.
2. Todas las acciones sobre la auditoría se llevan a cabo con la creación de triggers sobre todas las tablas formen parte del sistema de información:
 - *TRG_ Nombre-Corto-Tabla _AUD*: registra las modificaciones e inserciones en la auditoría.
 - *TRG_ Nombre-Corto-Tabla_MOD*: registra las eliminaciones en la auditoría.
3. Las tablas que tienen como clave primaria un campo de id de línea usan una secuencia para obtener los valores. En este caso se han creado 2 triggers diferentes para la misma función:
 - *TRG_ Nombre-Corto-Tabla_PK*: obtiene el siguiente valor de la secuencia y lo inserta.
 - *TRG_ Nombre-Corto-Tabla_USR*: obtiene el siguiente valor de la secuencia y lo inserta. Además el usuario de inserción por defecto es el administrador.

Dado que ambos triggers utilizan la misma secuencia, sólo uno de ellos podrá estar operativo. Se ha tomado la decisión de realizar esta separación para trabajar con o sin control de usuario. Se han creado los siguientes scripts para facilitar el cambio de un modo a otro:

- *Control de Tablas con Usuario.sql*: Inserta el registro y el usuario.
- *Control de Tablas sin Usuario.sql*: Únicamente inserta el registro.

Para que el control sea completo, también hay que modificar los packages ya que las funciones que manejan los datos reciben como parámetro el id

del usuario. Esto lo solventaremos gracias a la propiedad *default* para dar un valor predeterminado al parámetro.

4. La organización de los packages se realiza por área y funcionalidad:

- **PKG_GCU_MAESTROS:** Se crean las funciones de inserción, modificación y borrado de las tablas maestro.

Gestión de Concesionarios

- Crea_Concesionario
- Modifica_Concesionario
- Elimina_Concesionario

Gestión de Marcas

- Crea_Marca
- Modifica_Marca
- Elimina_Marca

Gestión de Modelos

- Crea_Modelo
- Modifica_Modelo
- Elimina_Modelo

Gestión de Características

- Crea_Característica
- Modifica_Característica
- Elimina_Característica

Gestión de Catalogo

- Crea_Catalogo
- Modifica_Catalogo
- Elimina_Catalogo

Gestión de Empleados

- Crea_Empleado
- Modifica_Empleado
- Elimina_Empleado

- **PKG_GCU_VEHICULOS:** Mantenimiento de los vehículos

Gestión de Vehículos

- Crea_Vehiculo
- Modifica_Vehiculo
- Elimina_Vehiculo
- Vehículo_Vendido

- Configuración de Vehículo
 - Añade_Caracteristica
 - Elimina_Caracteristica
- PKG_GCU_COMERCIAL: Realización de ventas
 - Gestión de Ventas
 - Nueva Venta
- PKG_GCU_TALLER: Gestión de reparaciones
 - Gestión de Reparaciones
 - Nueva_Reparacion
 - Nueva_Tarea
 - Modifica_Tarea
 - Elimina_Tarea
 - Cierra_Reparacion
- PKG_GCU_FACTURACIÓN: Generación y gestión de facturas.
 - Gestión de Facturas
 - Facturar_Venta
 - Facturar_Reparacion

El package de facturación se ha separado ya que siempre debe ser una parte independiente en el sistema para facilitar cualquier integración posterior con algún sistema contable. En nuestro proyecto, hemos añadido en la factura un marcador para identificar si se ha contabilizado.

Algunas funciones no se muestran, ya que en el desglose realizado sólo se muestran las funciones públicas. En las descripciones de los cuerpos de los paquetes se podrán encontrar más funciones.

5. Se crean algunas funciones para encapsular código fuente.
 - FUN_GET_TOTAL_FACTURA obtiene el total de factura sumando el importe de sus líneas.
 - FUN_ENCRIPTA codifica la contraseña del usuario
 - FUN_GET_CONID_BY_USER obtiene el código del concesionario a partir del código del usuario
 - FUN_GET_TIPEMP_BY_USER obtiene el tipo de empleado a partir del código de usuario.
6. Para cumplir la relación <<include>> entre los procesos *Realiza venta* y *Genera Factura* hay generar una venta, automáticamente se emite una factura.

7. La gestión de datos de DWH se lleva a cabo con tablas independientes. Un Job que se ejecuta cada noche actualiza los datos. Este Job podría ejecutarse más a menudo si se considera necesario. Simplemente habría que re planificar el Job.
8. El package PKG_DWH_CARGA se ocupa de realizar la carga de datos cuando el Job lo precisa.

Scripts de implementación

Para poder importar correctamente el trabajo práctico realizado, en la entrega del proyecto se hace entrega de una carpeta llamada "Scripts_Entrega" que contiene los scripts necesarios para importar la base de datos realizada.

Todos los scripts de la carpeta están numerados y deben ejecutarse por orden para evitar errores. A continuación explicamos para que se debe utilizar cada uno de ellos:

- *0_Creación_usuario.sql* crea un usuario de base de datos con el nombre UOC. Se asignan los roles de *RESOURCE* Y *CONNECT*. Se incluyen todos los grants que se asignan por defecto a la hora de crear un usuario y dos adicionales situados en último lugar, que hemos añadido para poder encriptar las contraseñas de los usuarios.
- *1_Secuencias.sql* contiene la creación de todas las secuencias utilizadas.
- *2_Tablas.sql* incluye la creación de todas las tablas.
- *3_PK_Indices.sql* incluye la creación de todas las constraint de primary key, los índices, las claves únicas y las constraints tipo check.
- *4_FK_Constraints1.sql*
- *5_Triggers.sql* contiene la creación de los desencadenadores
- *6_Jobs.sql* contiene la creación del job encargado de realizar la carga de datos DWH.
- *7_Packages.sql* contiene el código PL/SQL de todos los packages que gestionan la información.
- *8_Funciones.sql* contiene el código PL/SQL de las funciones.

Capítulo 5: Valoración económica

En base a la planificación realizada, podemos realizar el una valoración económica del coste del proyecto. Para ello, consideramos ciertas las siguientes afirmaciones:

- No se computa ninguna partida presupuestaria dedicada al hardware necesario para realizar el proyecto.
- El grupo de trabajo, en este caso una única persona, dispone de todos los conocimientos técnicos necesarios para la consecución del proyecto y por tanto no necesita realizar ninguna jornada de formación.

Para realizar el presupuesto, debemos conocer el coste de una hora de trabajo. En este presupuesto partiremos de la base de que el coste de cada hora de trabajo es de 30€³.

Finalmente presentamos el coste final del proyecto en el siguiente presupuesto:

Presupuesto				
Concepto		Horas	Coste Uni.	Coste
Gestión de Concesionario				
Plan de proyecto		39	30,00 €	1.170,00 €
Análisis y Diseño		120	30,00 €	3.600,00 €
Implementación		132	30,00 €	3.960,00 €
Seguimiento y Control		18	30,00 €	540,00 €
Total				9.270,00 €

Hubiera sido posible realizar el presupuesto diferenciando el coste de las horas, según el rol que desempeña la tarea. Es frecuente encontrar distintas valoraciones para los jefes de proyecto, los analistas o diseñadores y los programadores. En este el presupuesto se calcula a partir de las horas dedicadas por cada rol. En este proyecto, suponiendo que los precios son 60€, 45€ y 30€ para los jefes de proyecto, analistas o diseñadores y programadores respectivamente, el presupuesto final del proyecto serían: **12.240,00€**

³ Todos los costes mostrados en la valoración económica se han calculado realizando la media de los precios que ofrecen 3 consultorías de las Islas Baleares dedicadas al desarrollo de aplicaciones.

Capítulo 6: Conclusiones

Una vez finalizado el proyecto, valoramos el trabajo realizado en función del resultado obtenido.

En primer lugar cabe destacar que la planificación de las tareas relacionadas con el Data WareHouse se ha retrasado debido a la falta de experiencia en este campo.

Una vez realizado el proyecto, se presentan una lista de las posibles mejoras que se podrían realizar en el sistema, con el fin de añadir más funcionalidades al sistema y hacerlo más completo.

- Añadir al modelo de datos un sistema de permisos para obtener información según los permisos del usuario que los solicita
- Realizar una interfaz hacia el sistema contable, para poder contabilizar las facturas sin tener que mecanizarlas manualmente en otro sistema
- Aprovechar los emails de los clientes y realizar campañas de envío de información (por ejemplo nuevos modelos) vía correo electrónico.
- Poder modificar importes de venta / reparación.
- Añadir ofertas o aplicar descuentos automáticamente al realizar una venta o reparación.
- En las modificaciones de datos, hacer uso de la sentencia NVL y en el caso de que el parámetro pasado sea nulo no actualizar dicho campo. Así no será necesario obtener todos los datos de una entidad para modificar alguno de sus campos.
- Implementar el cambio de contraseña.

Finalmente, para el desarrollo de este proyecto, he puesto en práctica muchos conocimientos adquiridos en otras asignaturas y en mi formación laboral con el fin de realizar un proyecto completo.

Glosario

Check Constraint: Restricción de chequeo.

Constraint: Restricción de base de datos.

DWH, Data WareHousw: Almacén de datos.

Encriptar: Utilizar algoritmo de encriptación para cifrar los datos.

Foreign key: Clave foránea de la tabla.

Grant: Permiso.

Job: Trabajo automatizado de base de datos.

Package: Paquete de base de datos.

Primary key: Clave primaria de la tabla.

Procedure: Procedimiento de base de datos

Trigger: Desencadenador de base de datos.

Unique key: Clave única.

Bibliografía

CODINA, LLUIÍS (1996). *Análisis de sistemas y metodología de diseño de bases de datos documentales*. Universidad Pompeu Fabra de Barcelona. [Fecha de consulta: 02/10/2012]

Oracle® Database (2005) *Data Warehousing Guide 10g Release 2 (10.2)* [En línea] http://docs.oracle.com/cd/B19306_01/server.102/b14223.pdf

Ministerio del Interior. *Curso Experto en Business Intelligence y Gestión del Conocimiento. Módulo 9: Migración e integración de datos. Módulo 10: Data Warehouse, Data Mart y tecnología OLAP con Microsoft*

Anexo A – Documentación

A.1. Modelo relacional completo

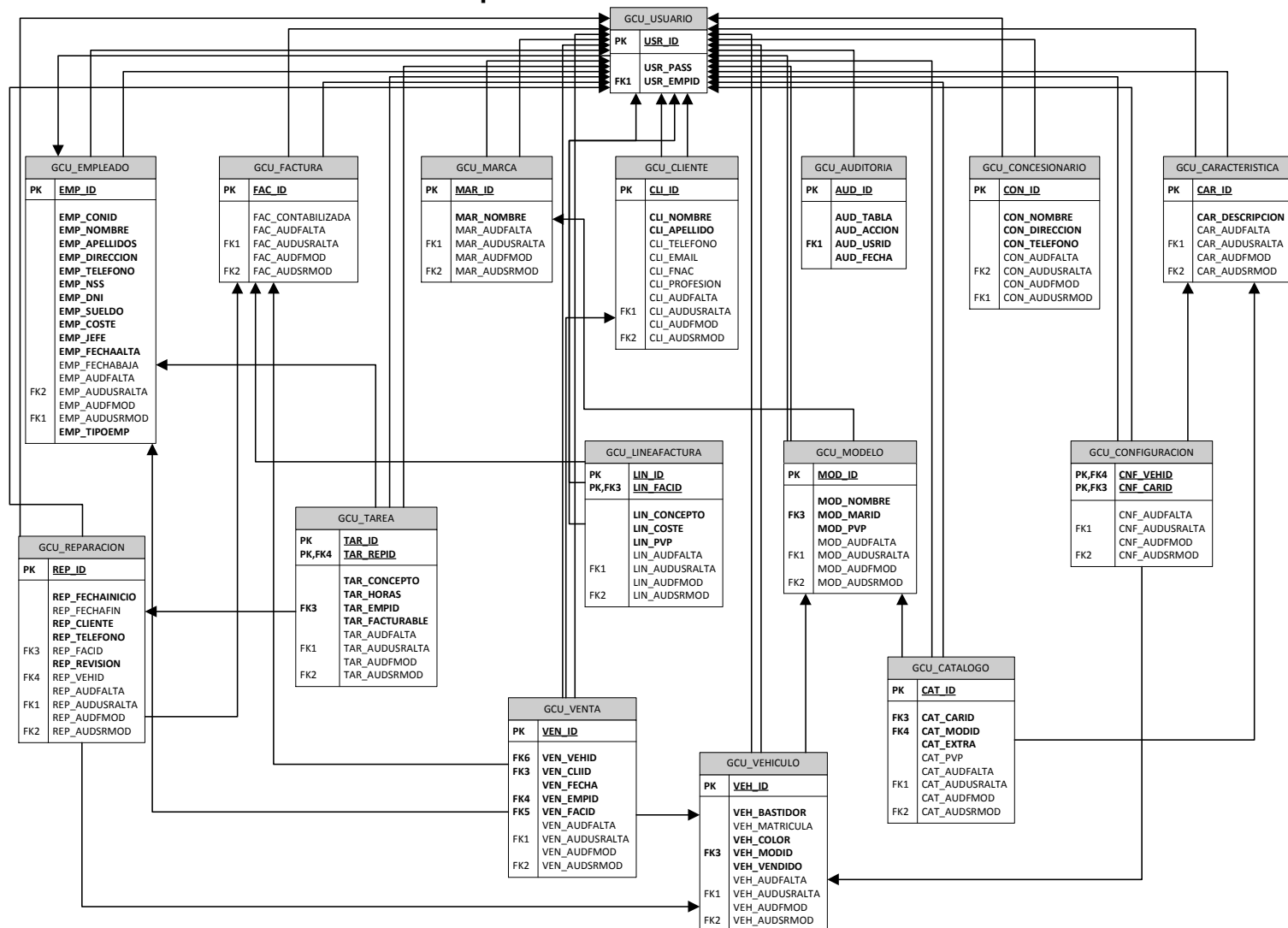


ILUSTRACIÓN A.1. MODELO RELACIONAL