



# Diseño e implementación de una base de datos relacional para la gestión de un videojuego

VERSIÓN  
**1.0**

DIRIGIDO A

---

**AULA DE PFC-BASES DE DATOS**

 **UOC** Universitat Oberta de Catalunya

**Datos de Control**

**Proyecto:** Proyecto Final de Carrera  
**Entidad de destino:** Aula de PFC-Bases de Datos  
**Título:** Diseño e implementación de una base de datos relacional para la gestión de un videojuego  
**Referencia:** INF-2013-00001-04  
**Versión:** 1.0  
**Fecha edición:** 11/01/2014  
**Fichero:** alopezjuan\_memoria.doc  
**Autor(es):** A. López  
 MEMORIA

**Control de firmas**

	<b>Autor</b>	<b>Revisado</b>
<b>Firma</b>	_____	_____
<b>Nombre</b>	Amalia López Juan	Juan Martínez Bolaños
<b>Cargo</b>	Alumno	Consultor
<b>Entidad</b>	UOC	UOC
<b>Fecha</b>	22/01/2014	22/01/2014

<b>Diseño e implementación de una base de datos relacional para la gestión de un videojuego</b>		
<b>Título:</b>		PFC
<b>Referencia:</b>	INF-2013-00001-04	
<b>Modificación:</b>	22/01/2014 17:07	Página 2 de 56

**Control de modificaciones por cambio de versión**

<i><b>Versión</b></i>	<i><b>Autor</b></i>	<i><b>Descripción</b></i>	<i><b>Fecha</b></i>
1	usuario	MEMORIA	26.12.13

<b>Diseño e implementación de una base de datos relacional para la gestión de un videojuego</b>			
<b>Título:</b>			PFC
<b>Referencia:</b>	INF-2013-00001-04		
<b>Modificación:</b>	22/01/2014 17:07		Página 3 de 56

## Resumen

Es objeto de este informe reflejar el desarrollo del Proyecto Final de Carrera en la modalidad de Bases de Datos. Este documento está dirigido al consultor y profesores de la asignatura para su valoración.

Se ha solicitado el diseño de una base de datos relacional para el control de un videojuego. En esta base de datos se almacenarán los datos de los usuarios, sus compras, toda la información generada en cada partida y permitirá controlar la gestión del juego. Aunque en el enunciado se pide extraer cálculos directamente desde las tablas, también se solicita el diseño de un Almacén de datos con el que se puedan obtener estadísticas más complejas.

Para la gestión del proyecto se ha seguido el estándar *PMBOOK* y se han tomado como base las directrices de las asignaturas “*metodología y gestión de proyectos informáticos*”, además de las de “*Sistemas de Gestión de Bases de Datos*” y “*Modelos multidimensionales y Almacenes de datos*”.

En ésta memoria se puede encontrar lo siguiente:

- El plan de trabajo: donde se deciden las tareas en las que se va a dividir el proyecto, su duración y las fechas en las que realizar cada una.
- Toda la documentación relacionada con el diseño de la Base de Datos: UML, casos de uso, especificaciones, diseño UML, lógico y Físico.
- Toda la documentación relacionada con la creación del Almacén de datos: especificaciones, decisiones sobre los hechos, dimensiones, medidas a controlar, diseño conceptual, diseño lógico y físico.
- Otros puntos de la gestión de un proyecto como el presupuesto.

<b>Diseño e implementación de una base de datos relacional para la gestión de un videojuego</b>		
<b>Título:</b>		PFC
<b>Referencia:</b>	INF-2013-00001-04	
<b>Modificación:</b>	22/01/2014 17:07	Página 4 de 56

## Índice de contenidos

1)	Introducción.....	7
1.1)	Justificación del PFC .....	7
1.2)	Objetivos del PFC.....	7
1.3)	Metodología de trabajo .....	8
1.4)	Plan de Trabajo .....	8
1.4.1)	Hitos .....	8
1.4.2)	Tareas .....	9
1.4.3)	Esfuerzo y carga de trabajo. ....	12
1.4.4)	Evaluación de Riesgos. ....	14
1.4.5)	Diagrama de Gantt.....	14
1.4.6)	Materiales. ....	17
1.5)	Productos Obtenidos. ....	17
1.6)	Descripción de los siguientes apartados. ....	18
2)	Base de Datos.....	19
2.1)	Análisis de Requisitos.....	19
2.1.1)	Requisitos del enunciado.....	19
2.1.2)	Otros requisitos tenidos en cuenta. ....	20
2.2)	Casos de uso.....	21
2.3)	Diseño Conceptual UML .....	25
2.4)	Diseño Lógico .....	26
2.5)	Diseño Físico de la BD. ....	27
2.6)	Descripción de los Procedimientos y Funciones. ....	28
2.7)	Procedimiento para pruebas. ....	33
3)	Almacén de Datos. ....	34
3.1)	Estudio de los requisitos .....	34
3.2)	Casos de uso.....	35
3.3)	Observaciones.....	37
3.4)	Decisión de los elementos de análisis.....	37
3.5)	Diseño Conceptual UML. ....	37
3.6)	Diseño Lógico .....	40
3.7)	Diseño Físico de la BD. ....	41
3.8)	Diseño de procesos ETL.....	42
3.9)	Pruebas .....	48
4)	Valoración Económica.....	51
4.1)	Clasificación de Tareas.....	51
4.2)	Presupuesto estimado. ....	52
4.3)	Presupuesto real.....	52
5)	Conclusiones.....	53
6)	Glosario.....	53
7)	Bibliografía .....	55
8)	Anexo. Procedimiento de instalación: .....	55

<b>Diseño e implementación de una base de datos relacional para la gestión de un videojuego</b>		
<b>Título:</b>		PFC
<b>Referencia:</b>	INF-2013-00001-04	
<b>Modificación:</b>	22/01/2014 17:07	Página 5 de 56

**Índice de figuras**

Figura 1. Diagrama de Gantt..... 17  
 Figura 2. Diagrama de casos de uso del Responsable de Dirección..... 21  
 Figura 3. Diagrama de casos de uso del Dpto. de compras..... 21  
 Figura 4. Diagrama de casos de uso del Administrador del videojuego ..... 22  
 Figura 5. Diagrama de casos de uso del usuario del juego..... 22  
 Figura 6. Diagrama UML de la Base de Datos ..... 25  
 Figura 7. Diagrama de casos de uso del Data Warehouse ..... 36  
 Figura 8. Diagrama UML Data Warehouse Hecho sesion..... 38  
 Figura 9. Diagrama UML Data Warehouse Hecho compras..... 38  
 Figura 10. Diagrama UML Data Warehouse Hecho amistad..... 39  
 Figura 11. Diagrama UML Data Warehouse Hecho Jugadas..... 39  
 Figura 12. Contenido del fichero JOB\_CARGA\_DW..... 42  
 Figura 13. Contenido del fichero TRA\_ETL\_DUSUARIO..... 43  
 Figura 14. Contenido del fichero TRA\_ETL\_DPLATAFORMA ..... 43  
 Figura 15. Contenido del fichero TRA\_ETL\_DNIVEL..... 43  
 Figura 16. Contenido del fichero TRA\_ETL\_HSESION..... 44  
 Figura 17. Contenido del fichero TRA\_ETL\_HJUGADAS ..... 45  
 Figura 18. Contenido del fichero TRA\_ETL\_HAMISTAD ..... 46  
 Figura 19. Contenido del fichero TRA\_ETL\_HCOMPRAS ..... 47  
 Figura 20. Análisis hecho jugadas numero\_jugadas por Usuario..... 48  
 Figura 21. Análisis hecho sesión métricas por usuario: ..... 49  
 Figura 22. Análisis hecho sesión. Métricas por Navegador..... 49  
 Figura 23. Análisis hecho compras importe por Usuario..... 50  
 Figura 24. Análisis hecho compras numero\_compras por Usuario. .... 50

<b>Diseño e implementación de una base de datos relacional para la gestión de un videojuego</b>		
<b>Título:</b>		PFC
<b>Referencia:</b>	INF-2013-00001-04	
<b>Modificación:</b>	22/01/2014 17:07	Página 6 de 56

# 1) *Introducción*

## 1.1) Justificación del PFC

El proyecto final de carrera permite agrupar y afianzar los conocimientos de distintas asignaturas, poner en práctica la teoría explicada y adaptarla a un entorno “real” aunque sea a pequeña escala.

He solicitado el área relacionada con las Bases de Datos porque es un punto muy importante en el puesto de trabajo que desempeño actualmente. En la mayoría de los casos trabajo con Bases de datos ya existentes implementadas por terceros, lo que me ha llevado a echar de menos en muchas ocasiones y valorar la redacción de una buena documentación. También tengo que diseñar e implementar algunas nuevas para que puedan funcionar con las herramientas de modernización administrativa (Servlets, Webservices, portlets y otros), y estoy estudiando la posible implantación de un Almacén de Datos que ayude en la toma de decisiones.

De acuerdo al enunciado del PFC la empresa está desarrollando un videojuego al que hay que incorporar un sistema para obtener datos de uso de los clientes tanto de navegador como de móvil. Nuestro equipo es el encargado de analizar los requisitos del nuevo sistema y la posterior implementación de la base de datos para satisfacerlos, así como también un almacén de datos (data warehouse) para extraer estadísticas.

## 1.2) Objetivos del PFC

El producto final es una Base de datos que recopila información de los usuarios para la gestión de un videojuego accesible desde varias plataformas móviles y varios navegadores Web con las siguientes funciones:

- Control sobre las vidas y las ayudas del juego.
- Controlar las compras de los usuarios.
- Obtener información de los usuarios.
- Gestión de niveles

Se solicita que toda la gestión y el acceso a la información se realice a través de procedimientos, no se permitirá ningún otro método de acceso a la BD. Se pide además facilitar la integración con el resto del sistema. (Logs, baterías de pruebas para testear la BD, etc.)

A partir de dicha Base de Datos se debe generar un Almacén de Datos para elaborar estadísticas.

<b>Diseño e implementación de una base de datos relacional para la gestión de un videojuego</b>		
<b>Título:</b>		PFC
<b>Referencia:</b>	INF-2013-00001-04	
<b>Modificación:</b>	22/01/2014 17:07	Página 7 de 56

## 1.3) Metodología de trabajo

Se sigue el desarrollo en cascada, "...también llamado modelo en cascada, es el enfoque metodológico que ordena rigurosamente las etapas del proceso para el desarrollo de software, de tal forma que el inicio de cada etapa debe esperar a la finalización de la etapa anterior..."<sup>(1)</sup>

Se ha tomado como fuente el estándar PMBOK, puesto que es el que más conozco a raíz de la asignatura MGPI. Puede que algunos apartados hayan sido simplificados porque al ser un proyecto pequeño no se den en este PFC, pero aún así creo que se adapta muy bien a la estructura del proyecto.

Dado el enunciado del PFC, y su división en varias PAC he decidido seguir la metodología de gestión de proyectos orientada a objetivos (goal directed project management), separando la planificación estratégica del proyecto de la planificación operativa (enfocada a actividades y tareas). En términos del PMBOK, el resultado principal de la planificación estratégica es la descomposición del trabajo en partes o paquetes de trabajo más pequeñas, o EDT (work breakdown structure, o estructura de distribución del trabajo), que en realidad son entregables (deliverables) parciales o generales. Seguidamente se lleva a cabo la planificación operativa, descomponiendo cada EDT en actividades, poniéndolas en secuencia, estimando los recursos necesarios y estableciendo un calendario preliminar. Finalmente se estiman los costes y se elabora el presupuesto.

## 1.4) Plan de Trabajo

### 1.4.1) Hitos

Los hitos principales y sus fechas límite han sido marcados por la propia UOC:

ENTREGA PAC1 : 7/10/13

ENTREGA PAC2: 11/11/13

ENTREGA PAC3: 11/12/13

ENTREGA FINAL: 13/01/14

Se han incluido otros hitos que se consideran como parte de la gestión de un proyecto.

PREPARACIÓN DEL PROYECTO: Los primeros días del inicio del curso.

FORMACIÓN: Necesaria durante todo el proyecto

SEGUIMIENTO Y CONTROL: Necesario durante todo el proyecto, especialmente en la publicación de la solución de cada entrega.

<b>Diseño e implementación de una base de datos relacional para la gestión de un videojuego</b>		
<b>Título:</b>		PFC
<b>Referencia:</b>	INF-2013-00001-04	
<b>Modificación:</b>	22/01/2014 17:07	Página 8 de 56



## 1.4.2) Tareas

### Preparación del proyecto

Descarga de documentación y materiales
Descarga desde el aula el enunciado del PFC, la documentación adicional aportada por el consultor, las recomendaciones y el material necesario para la asignatura.

Instalación de material
Instalación del Microsoft Project para comenzar con la planificación.

Lectura Documento Enunciado PFC.
Análisis del Enunciado del PFC para conocer los requisitos y los posibles hitos del proyecto.

### Formación

Recopilación información adicional
Buscar otros trabajos, manuales del software a instalar, foros, etc.

Lectura documentación adicional
Estudio del material descargado

### Elaboración PT (PAC 1)

Análisis de requisitos del proyecto. EDT
Definir los Hitos y sus fechas límite.

Definir actividades a llevar a cabo
Dividir los hitos de la actividad anterior en subtareas.

Establecer la secuencia de realización
Establecer la jerarquía de ejecución. Analizar qué subtareas no pueden empezar sin haber terminado las anteriores y cuales pueden realizarse en paralelo.

Estimar el esfuerzo y carga de trabajo requerida
Planificar el número de horas para cada una.

<b>Diseño e implementación de una base de datos relacional para la gestión de un videojuego</b>		
<b>Título:</b>		PFC
<b>Referencia:</b>	INF-2013-00001-04	
<b>Modificación:</b>	22/01/2014 17:07	Página 9 de 56

**Establecer el calendario de trabajo provisional**  
 Delimitar las fechas y los días de trabajo necesarios para cada tarea una vez estimado el esfuerzo.

**Evaluar riesgos y definir contingencias**  
 Estudiar el calendario y los posibles riesgos que pudieran surgir.

**Establecer el presupuesto provisional.**  
 Calcular el importe del proyecto, el coste de las horas a invertir, los gastos de material, etc.

**Redacción de documentos y entrega PAC1**  
 Generar la documentación y entregarla.

**Elaboración BD (PAC 2)**

**Decisión del software a utilizar e instalación**  
 Preparación del equipo e instalación del software para generar la Base de Datos.

**Diseño conceptual UML**  
 Estudio de los requisitos y diseño UML de la BD.

**Diseño Lógico**  
 Transformación del diseño en UML obtenido en el paso anterior en el diseño lógico.

**Consulta manual software instalado**  
 Consulta sobre el estándar SQL.

**Diseño Físico (Tablas, triggers, y procedimientos)**  
 Generación del código SQL necesario para obtener la BD (Las tablas, los triggers y los procedimientos)

**Pruebas y refinamiento del producto**  
 Montar la BD y probarla. Realimentar el diseño de la BD resolviendo los errores detectados en las pruebas.

**Redacción de documentación y entrega PAC2**  
 Redactar la documentación para la entrega parcial y subirla al aula mediante la página Web.

<b>Diseño e implementación de una base de datos relacional para la gestión de un videojuego</b>		
<b>Título:</b>		PFC
<b>Referencia:</b>	INF-2013-00001-04	
<b>Modificación:</b>	22/01/2014 17:07	Página 10 de 56

### Elaboración del Almacén de Datos (PAC 3)

Decisión del software a utilizar e instalación
Preparación del equipo e instalación del software para generar el data warehouse (Almacén de datos).
Consulta manual software instalado
Consulta de los lenguajes utilizados para el diseño del AD
Análisis funcional de los requisitos
Revisar el enunciado para obtener las funcionalidades deseadas.
Identificación de los elementos de análisis (indicadores, dimensiones, atributos)
Establecer las dimensiones, los atributos y los indicadores más apropiados a los requisitos obtenidos del paso anterior.
Diseño conceptual del modelo multidimensional
Diseño UML del AD
Diseño Lógico del modelo multidimensional
Transformación del diseño UML del paso anterior en el diseño lógico.
Diseño de procesos ETL
Definir las diferentes transformaciones que serán necesarias para volcar los datos de la BD obtenida en la PAC2 hacia el nuevo AD.
Diseño Físico
Obtener el diseño físico a partir del diseño lógico del AD
Construcción del AD y los procesos ETL
Aplicar los dos diseños (físico y ETL) en su aplicación correspondiente.
Pruebas y reparación del producto
Test del producto obtenido y realimentación de todas las fases que sean necesarias para resolver los errores detectados. Batería de pruebas.
Redacción de documento y entrega PAC3
Documentar el producto obtenido y entregarlo.

<b>Diseño e implementación de una base de datos relacional para la gestión de un videojuego</b>		
<b>Título:</b>		PFC
<b>Referencia:</b>	INF-2013-00001-04	
<b>Modificación:</b>	22/01/2014 17:07	Página 11 de 56

## Entrega Final

Redacción de la memoria
Redactar la memoria para la entrega final
Diseño de la Presentación
Diseñar la presentación para el tribunal virtual.
Revisión de la documentación y entrega PAC3
Revisar errores (de contexto, gramaticales, etc.) y realizar la entrega final.

## Seguimiento y Control

Consulta del foro
Consultar el foro del aula para escribir mensajes con mis dudas y conocer las respuestas a dudas de otros compañeros.
Seguimiento y Control PAC1
Revisar la planificación: Ver las notas de la PAC1 una vez revisado por el tutor y obtenida la solución. Modificar la planificación si es necesario.
Seguimiento y Control PAC2
Revisar la planificación Ver las notas de la PAC2 una vez revisado por el tutor y obtenida la solución. Modificar la planificación si es necesario.
Seguimiento y Control PAC3
Revisar la planificación Ver las notas de la PAC3 una vez revisado por el tutor y obtenida la solución. Modificar la planificación si es necesario.

### 1.4.3) **Esfuerzo y carga de trabajo.**

Se ha estimado un número de horas para cada una de las tareas en las que se ha subdividido el proyecto. Las horas que se indican en la tabla se han calculado a partir del tiempo invertido en otras asignaturas para realizar prácticas similares.

Preparación del proyecto	Descarga de documentación y materiales	2 horas
	Instalación de material	3 horas
	Lectura Documento Enunciado PFC	1 hora
Formación	Recopilación información adicional	3 horas
	Lectura documentación adicional	4 horas

<b>Diseño e implementación de una base de datos relacional para la gestión de un videojuego</b>		
<b>Título:</b>		PFC
<b>Referencia:</b>	INF-2013-00001-04	
<b>Modificación:</b>	22/01/2014 17:07	Página 12 de 56

	Análisis de requisitos del proyecto. EDT	1 hora
	Definir actividades a llevar a cabo	4 horas
	Establecer la secuencia de realización	1 hora
Elaboración PT (PAC 1)	Estimar el esfuerzo y carga de trabajo requerida	2 horas
	Establecer el calendario de trabajo provisional	2 horas
	Evaluar riesgos y definir contingencias	2 horas
	Establecer el presupuesto provisional	1 hora
	Redacción de documentos y entrega PAC1	9 horas
	<hr/>	
Elaboración BD (PAC 2)	Decisión del software a utilizar e instalación	3 horas
	Diseño conceptual UML	9 horas
	Diseño Lógico	5 horas
	Consulta manual software instalado	3 horas
	Diseño Físico (Tablas, triggers, y procedimientos)	5 horas
	Pruebas y refinamiento del producto	4 horas
	Redacción de documentación y entrega PAC2	10 horas
<hr/>		
Elaboración del Almacén de Datos (PAC 3)	Decisión del software a utilizar e instalación	3 horas
	Consulta manual software instalado	5 horas
	Análisis funcional de los requisitos	5 horas
	Identificación de los elementos de análisis (indicadores, dimensiones, atributos)	5 horas
	Diseño conceptual del modelo multidimensional	10 horas
	Diseño Lógico del modelo multidimensional	5 horas
	Diseño de procesos ETL	4 horas
	Diseño Físico	4 horas
	Construcción del AD y los procesos ETL	5 horas
	Pruebas y reparación del producto	5 horas
Redacción de documento y entrega PAC3	10 horas	
<hr/>		
Entrega Final	Redacción de la memoria	10 horas
	Diseño de la Presentación	10 horas
	Revisión de la documentación y entrega PAC3	3 horas
<hr/>		
Seguimiento y Control	Consulta del foro	4 horas
	Seguimiento y Control PAC1. Revisar la planificación	4 horas

Diseño e implementación de una base de datos relacional para la gestión de un videojuego		
<b>Título:</b>		PFC
<b>Referencia:</b>	INF-2013-00001-04	
<b>Modificación:</b>	22/01/2014 17:07	Página 13 de 56

Seguimiento y Control PAC2. Revisar la planificación 4 horas

Seguimiento y Control PAC3. Revisar la planificación 4 horas

---

Total 174

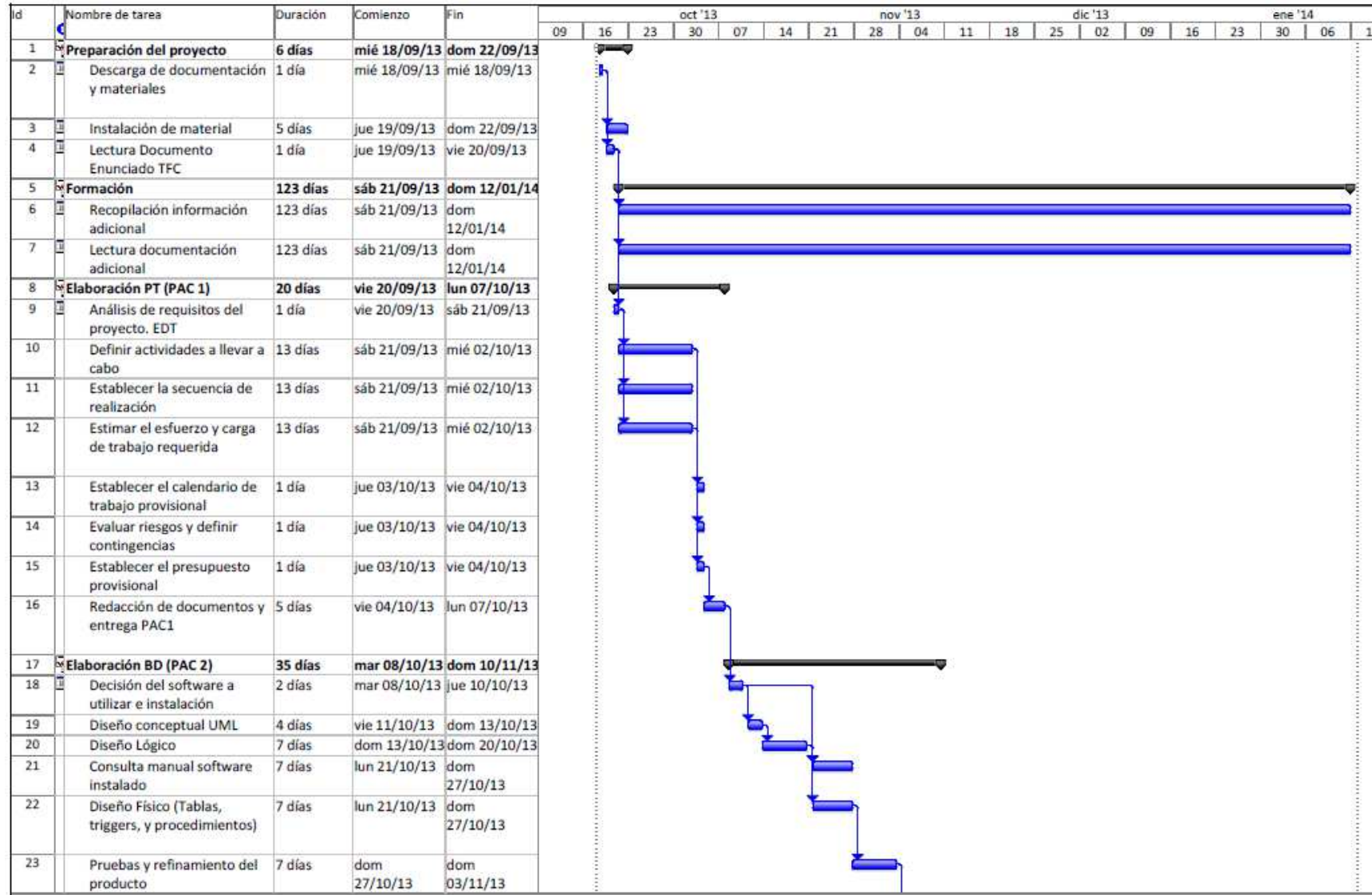
### 1.4.4) Evaluación de Riesgos.

ID	Incidencia	Consecuencia	Riesgo	Impacto	Medidas Correctoras
R1	Necesitar el fin de semana y/o el día festivo considerado como libre para una actividad con mayor prioridad	Retraso de las fechas de entrega o afectación a la calidad	Bajo	Alto	Existe un margen en el calendario entre semana para absorberlo. Repartir las horas libres que queden y si aún no es suficiente quitárselas a las reservadas para las actividades de ocio con horario fijo o coger un día de vacaciones del trabajo
R2	Realización de horas extras en el trabajo	Retraso de las fechas de entrega o afectación a la calidad	Muy bajo	Alto	Las tareas con fecha de entrega límite van desde el 1 de febrero al 15 de septiembre. Debería realizarse la misma que en el R1
R3	Planificación de horas incorrecta	Retraso de las fechas de entrega o afectación a la calidad	Alto	Alto	Se recomienda como medida preventiva adelantar el trabajo lo máximo posible utilizando el tiempo libre de margen aunque el proyecto se realice según la planificación.
R4	Viaje familiar	Retraso de las fechas de entrega o afectación a la calidad	Muy bajo	Alto	Misma que en R1
R5	Viaje de trabajo	Retraso de las fechas de entrega o afectación a la calidad	Muy bajo	Alto	Misma que en R1
R6	Enfermedad que impida trabajar	Retraso de las fechas de entrega o afectación a la calidad	Medio	Alto	Misma que en R1
R7	Fallo del único equipo de trabajo del que se dispone	Pérdida de los datos. Incremento de las horas necesarias para la realización del proyecto por la compra, instalación y puesta a punto de otro equipo	Medio	Muy Alto	Trabajar con los datos en la nube. Utilizar maquinas virtuales y mantener una copia de seguridad en un HDD externo

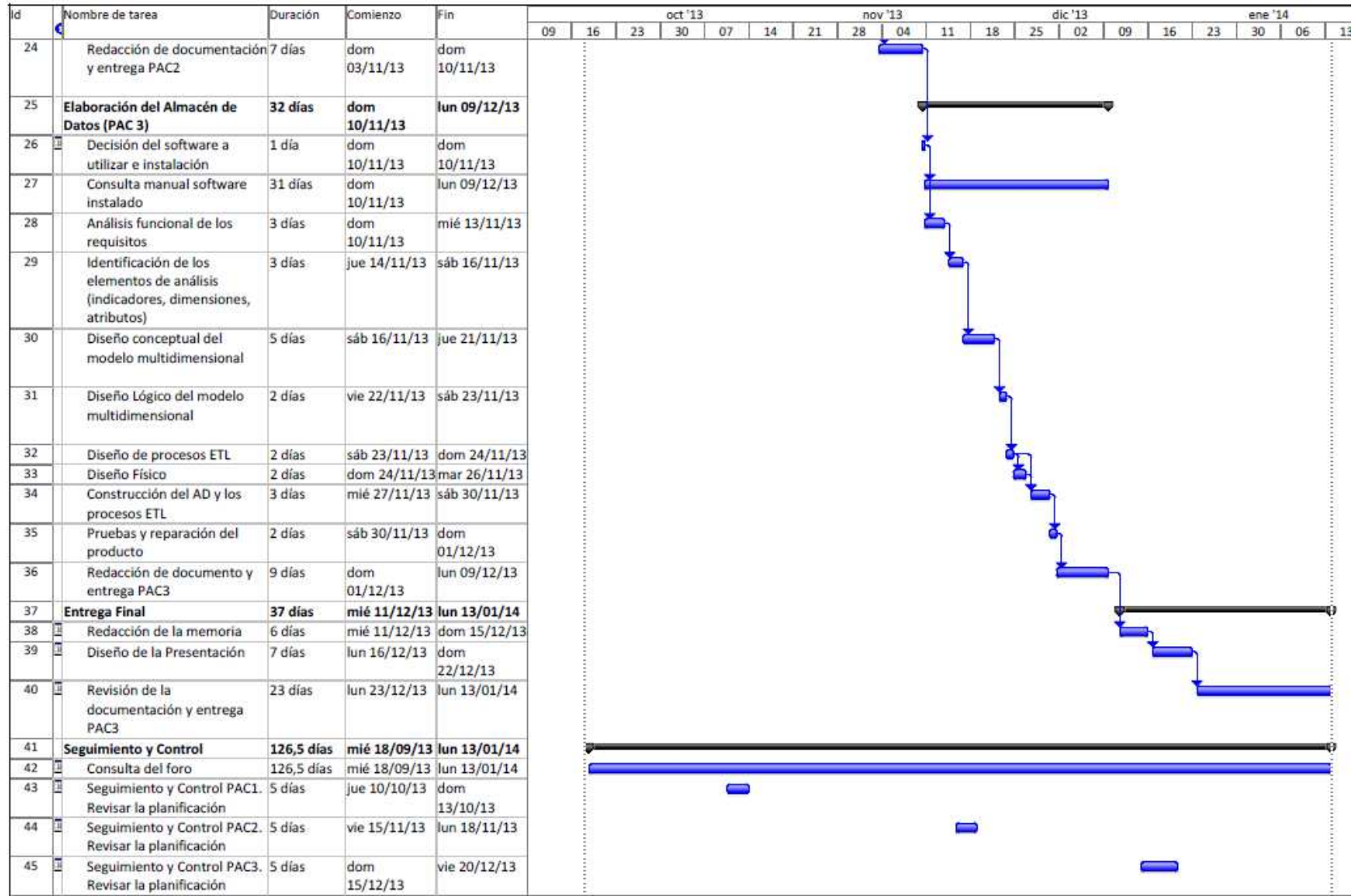
### 1.4.5) Diagrama de Gantt.

Se ha establecido un calendario reservando estrictamente el tiempo libre disponible para realizar el PFC y reservando un número de horas menor a las disponibles para no ir muy ajustada.

<b>Diseño e implementación de una base de datos relacional para la gestión de un videojuego</b>		
<b>Título:</b>		PFC
<b>Referencia:</b>	INF-2013-00001-04	
<b>Modificación:</b>	22/01/2014 17:07	Página 14 de 56



Diseño e implementación de una base de datos relacional para la gestión de un videojuego		
Título:		PFC
Referencia:	INF-2013-00001-04	
Modificación:	22/01/2014 17:07	Página 15 de 56



Diseño e implementación de una base de datos relacional para la gestión de un videojuego	
Título:	PFC
Referencia:	INF-2013-00001-04
Modificación:	22/01/2014 17:07



Figura 1. Diagrama de Gantt.

### 1.4.6) **Materiales.**

Se dispone de un solo ordenador portátil adquirido para realizar las asignaturas del segundo semestre del curso 2012-2013 con varias máquinas virtuales.

Durante todo el proyecto se hará uso de las herramientas OpenOffice, Microsoft Office y Project.

Para el diseño de la Base de datos el software a utilizar será el ya instalado para la asignatura de SGBD. Se empleará como SGBD Oracle Express 11g, para generar diagramas UML se empleará MagicDraw y se utilizará la herramienta *SQL Developer* para trabajar con la base de datos.

Para el diseño del DW se ha elegido el mismo software instalado para la asignatura MMD. Oracle Express 11g como SGBD para almacenar el AD y Pentaho para explotarlo. PDI para el diseño de las ETL y PSW para realizar consultas sobre los hechos y las dimensiones. También se usará la misma herramienta SQL que la utilizada para generar los scripts de creación de las tablas de la BD (*SQL Developer*).

## 1.5) **Productos Obtenidos.**

- Script SQL para la creación de la BD.

Fichero *alopezjuan\_PAC2.sql* con el código sql para la creación de las tablas, vistas, secuencias, desencadenantes, funciones y procedimientos utilizados por la BD.

- Script SQL para las pruebas de la BD.

Fichero *alopezjuan\_PAC2\_PruebasBD.sql* con el código sql para la creación del procedimiento de pruebas de la Base de datos y su ejecución.

- Script SQL para la creación del Almacén de datos

Fichero *alopezjuan\_PAC3.sql* con el código sql para la creación de las tablas, secuencias y desencadenantes donde almacenar la información extraída en los procesos ETL.

- Ficheros con el diseño de los procesos ETL.

Ficheros para rellenar los datos de las tablas de dimensiones: TRA\_ETL\_DNIVEL.ktr, TRA\_ETL\_DPLATAFORMA.ktr, TRA\_ETL\_DUSUARIO.ktr

Ficheros para rellenar los datos de las tablas de hechos: TRA\_ETL\_HAMISTAD.ktr, TRA\_ETL\_HCOMPRAS.ktr, TRA\_ETL\_HJUGADAS.ktr, TRA\_ETL\_HSESION.ktr

<b>Diseño e implementación de una base de datos relacional para la gestión de un videojuego</b>		
<b>Título:</b>		PFC
<b>Referencia:</b>	INF-2013-00001-04	
<b>Modificación:</b>	22/01/2014 17:07	Página 17 de 56

- Fichero para organizar el proceso de ejecución

Fichero JOB\_CARGA\_DW.kjb en el que se encuentra secuenciado la llamada a cada fichero ktr, y que permite programarlo para que se ejecute periódicamente a una hora determinada.

- Fichero XML con ejemplos de diseño de un cubo para explotar el DW.

PSW\_CUBOS.xml Permitirá publicar y analizar los datos contenidos en las tablas de hechos y dimensiones.

- Memoria

Documentación de todas las fases del proyecto

- Presentación

Diapositivas con un resumen gráfico de lo incluido en la memoria.

## 1.6) Descripción de los siguientes apartados.

Aquí termina la redacción del plan de trabajo. En los siguientes títulos podemos revisar la documentación obtenida en el proceso de creación de la Base de datos, y del Almacén de datos.

Dentro del apartado dedicado a la Base de datos se podrán consultar la especificación de los requisitos, los casos de uso del producto, su interpretación en el diagrama UML, el diseño lógico obtenido a partir de dicho diagrama, algunas aclaraciones sobre las decisiones tomadas en el diseño Físico, la descripción de los procedimientos creados para la explotación de la BD y ciertas especificaciones del procedimiento de pruebas.

La sección siguiente contiene un estudio de los requisitos, los casos de uso del Almacén de datos, especificaciones sobre los elementos de análisis detectados, el diseño conceptual y lógico del DW, unas notas sobre el diseño Físico, la descripción de los procesos ETL y un par de consultas estadísticas de ejemplo.

El resto de capítulos lo forman el presupuesto, donde se compara lo planificado con el importe real del proyecto, el glosario para aclarar términos poco conocidos y la bibliografía

<b>Diseño e implementación de una base de datos relacional para la gestión de un videojuego</b>		
<b>Título:</b>		PFC
<b>Referencia:</b>	INF-2013-00001-04	
<b>Modificación:</b>	22/01/2014 17:07	Página 18 de 56

## 2) Base de Datos.

### 2.1) Análisis de Requisitos

#### 2.1.1) Requisitos del enunciado

*“La nostra empresa està desenvolupant un videojoc al qual se l’ha d’incorporar un sistema per obtenir dades d’ús dels usuaris.*

*El nostre equip serà l’encarregat d’analitzar els requeriments del nou sistema, així com la posterior implementació de la base de dades per satisfer-los. Tota la gestió i accés a la informació es farà mitjançant procediments de base de dades, sent aquesta l’única manera d’accedir-hi.*

*El joc t diversos nivells, i l’usuari ha de superar un repte a cadascun d’ells per avançar. Un usuari només té 5 vides (possibilitats de repetir intent) al llarg d’un dia, tot i que pot aconseguir més demanant-ne a amics per alguna xarxa social o comprant-ne. A més a més de les vides, els usuaris també poden comprar ajudes que facilitin la superació de nivells.*

*Actualment el joc està disponible a diverses plataformes mòbils i navegadors web.*

*Per mesurar la penetració del joc, la BD ha de permetre obtenir informació sobre els usuaris (número, distribució geogràfica, sexe, etc.), sobre la seva interconnexió a les xarxes socials (vides que s’intercanvien) i la plataforma que utilitzen per jugar.*

*Igualment i per valorar la dificultat, la BD ha de permetre la gestió de nivells, obtenint informació sobre quants usuaris estan a cada nivell, el número d’intents que necessita cada usuari per superar el nivell, etc.*

*Per últim, ha de tenir constància de les compres que fan els usuaris, ja sigui de vides o ajudes.*

*La BD haurà de ser escalable per poder anar incorporant progressivament totes aquelles necessitats que sorgeixen durant la seva vigència..*

*...Finalment, per tal de facilitar el manteniment del sistema, es valorarà molt disposar de mecanismes que permetin resoldre potencials problemes d’integració amb la resta del sistema: un log de les accions fetes amb la BD, mecanismes per testejar la funcionalitat de la BD, etc.”*

<b>Diseño e implementación de una base de datos relacional para la gestión de un videojuego</b>		
<b>Título:</b>		PFC
<b>Referencia:</b>	INF-2013-00001-04	
<b>Modificación:</b>	22/01/2014 17:07	Página 19 de 56

## 2.1.2) Otros requisitos tenidos en cuenta.

Como lo definido en el enunciado es algo ambiguo, además se decide lo siguiente:

Cuando un jugador quiera acceder al juego por primera vez se le presentará un formulario de alta de usuario en el que se pedirá registrar todos los datos personales, sexo, fecha de nacimiento, etc. El servidor hará una llamada al procedimiento de alta del usuario al que se le enviarán todos los datos. Aunque podría existir un sistema de reconocimiento de localización geográfica a través de la IP se supone que el servidor los obtendrá a través del formulario correspondiente en el momento del registro.

Se asume que el usuario sólo puede mantener a la vez una sesión abierta. Si el servidor detecta que el jugador pierde la conexión deberá cerrarla llamando al procedimiento adecuado. Un usuario puede utilizar siempre el mismo medio o distintos dispositivos, plataformas o versiones.

Dada la posible internacionalización y las diferentes zonas horarias se considera que un usuario dispone de 5 vidas en las 24 horas siguientes a la primera solicitud de juego que realice sin vidas (o con todas las vidas caducadas), los plazos podrían cambiarse para que fueran o bien días estáticos de acuerdo a la hora del servidor o bien periodos fijos de 24 horas a partir de la primera conexión del jugador pero se considera más “justa” la opción escogida.

En cada jugada se registrará el nivel, la vida utilizada y la puntuación alcanzada. Un nivel se considerará superado si se ha obtenido al menos cierta puntuación mínima que será establecida en el momento de dar de alta el Nivel. En dicho proceso de alta también se aportará la descripción del reto a superar y ninguno de estos datos podrá modificarse una vez quede registrado.

Cuando no se ha superado un nivel, la vida con la que está jugando el usuario se pierde. Se puede volver a jugar un nivel ya superado pero si al volver a jugar un nivel no se alcanza la puntuación mínima la vida también se pierde.

Se pueden comprar varias vidas de una en una, o también adquirir más de una vida en una única compra.

El usuario dispone de ayudas, se considera que una ayuda pertenece a un solo nivel, y que un nivel puede tener varias ayudas. Se entiende que una vez que se ha comprado una ayuda sigue disponible y ya no se vuelve a comprar aunque necesite varios intentos para superar el nivel.

Los importes tanto de ayudas como de vidas se establecerán en una sola moneda. Será la misma para todas las compras sea de donde sea el usuario, al realizar la transacción ya se encargará la entidad bancaria en hacer la conversión si es necesario.

Sólo se podrán pedir vidas a otros usuarios cuando se hayan agotado todas las gratuitas y compradas disponibles, un jugador no podrá pedir vidas si aún le quedan.

<b>Diseño e implementación de una base de datos relacional para la gestión de un videojuego</b>		
<b>Título:</b>		PFC
<b>Referencia:</b>	INF-2013-00001-04	
<b>Modificación:</b>	22/01/2014 17:07	Página 20 de 56

Sólo podrán pedirse de una en una. Las vidas compradas no tendrán caducidad y también podrán cederse.

Cuando se cede una vida gratuita se revisará su caducidad. Estará disponible como máximo durante el mismo periodo de 24 horas de la última concesión de vidas del destinatario. Si por ejemplo las 5 vidas concedidas al iniciar la partida del jugador que pide la vida caducaban en 15 minutos, a los 15 minutos ésta también dejará de estar disponible, si la vida cedida caduca antes (por ejemplo a los 5 minutos) seguirá caducando a la misma hora. Las vidas cedidas que sean compradas tampoco caducarán para el nuevo usuario.

Para poder ceder vidas se deben tener 2 o más disponibles (da igual si son gratuitas o compradas). No se puede ceder si sólo se dispone de una vida (estaría en uso en ese momento).

Como los precios tanto de las vidas como de las ayudas se pueden modificar, se van a almacenar (en un caso real por motivos por ejemplo de facturación) los datos de los importes de las compras. Los importes de las ayudas pueden variar de unas a otras dependiendo por ejemplo del nivel en el que se encuentren o del número de ayudas que existan en un nivel.

Cada llamada a un procedimiento realizada desde el servidor será registrada creándose un histórico. Se mantendrá una tabla de Log para saber los movimientos realizados sobre la BBDD.

## 2.2) Casos de uso

Se distinguen los siguientes actores:

- Administrador de los datos básicos del videojuego
- Responsable dirección/Marketing. Analista de Estadísticas
- Dpto. Compras/Contabilidad
- Jugador de la partida

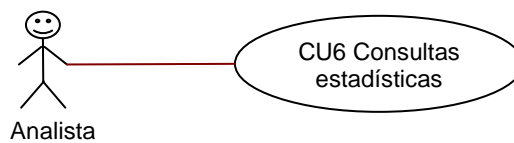


Figura 2. Diagrama de casos de uso del Responsable de Dirección.

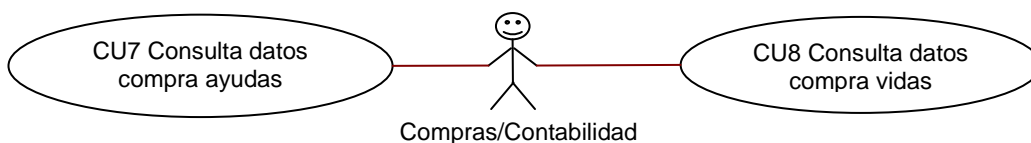


Figura 3. Diagrama de casos de uso del Dpto. de compras.

<b>Diseño e implementación de una base de datos relacional para la gestión de un videojuego</b>		
<b>Título:</b>		PFC
<b>Referencia:</b>	INF-2013-00001-04	
<b>Modificación:</b>	22/01/2014 17:07	Página 21 de 56

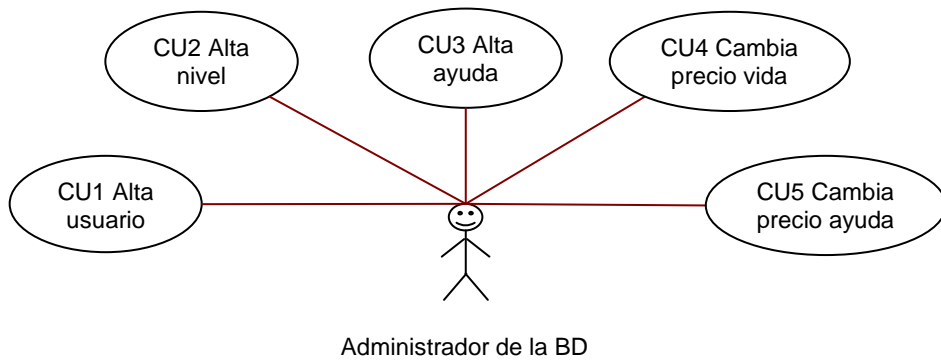


Figura 4. Diagrama de casos de uso del Administrador del videojuego

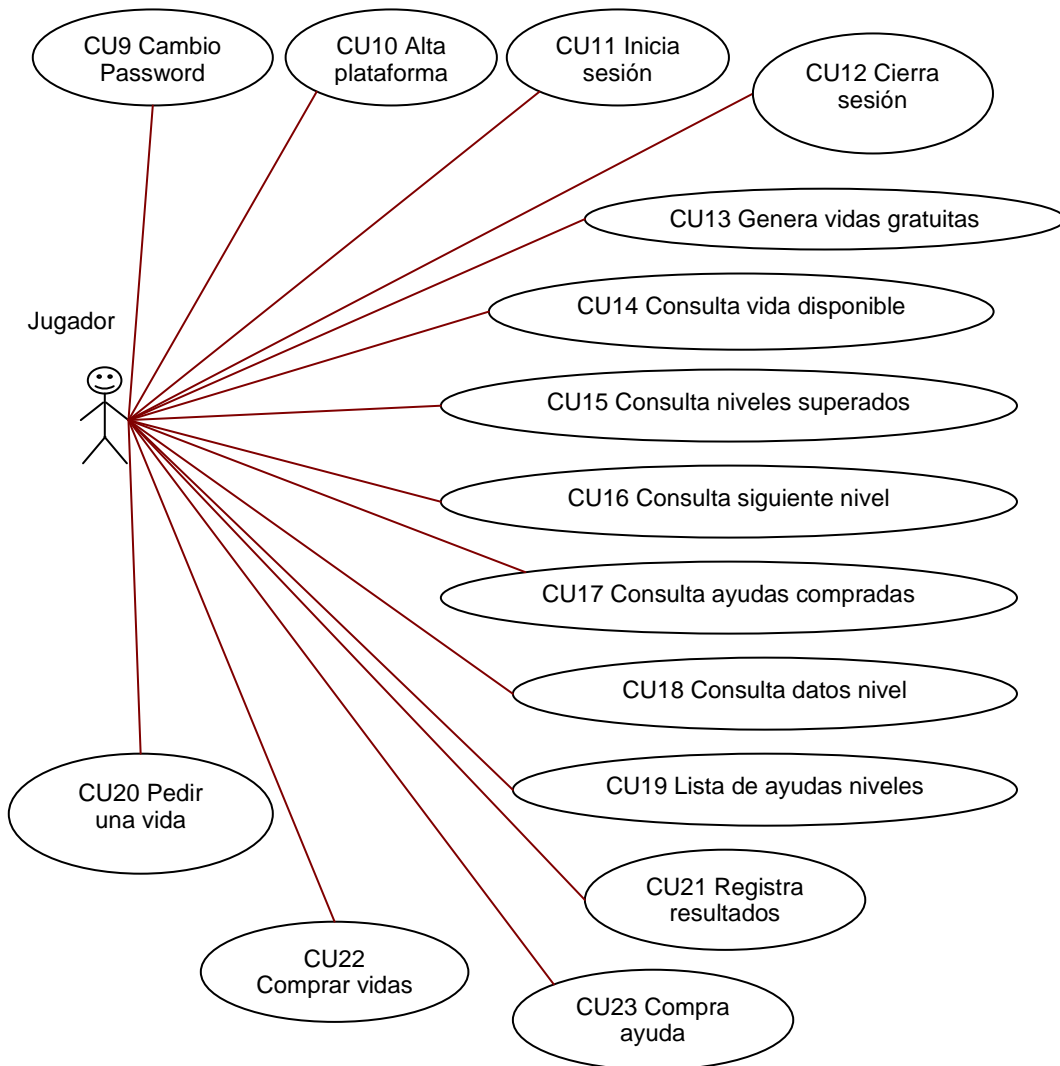


Figura 5. Diagrama de casos de uso del usuario del juego.

<b>Diseño e implementación de una base de datos relacional para la gestión de un videojuego</b>		
<b>Título:</b>		PFC
<b>Referencia:</b>	INF-2013-00001-04	
<b>Modificación:</b>	22/01/2014 17:07	Página 22 de 56

## Administrador de los datos del juego

CU1. Alta usuario
Registro de un nuevo usuario. Alta de todos los datos del nuevo jugador como identificador, nombre completo, fecha de nacimiento (para calcular la edad), domicilio fiscal, etc.

CU2. Alta nivel
Registro de un nuevo nivel. Registro del reto, la puntuación mínima necesaria para superar el nivel, etc.

CU3. Alta ayuda
Registro de una nueva ayuda. Registro del nivel para el que se crea la ayuda, su importe y descripción, etc.

CU4. Cambia precio vida
Cambia el precio de la vida.

CU5. Cambia precio ayuda
Modifica el precio de una ayuda.

**Analista de estadísticas del videojuego. Responsable de Dirección/Marketing.**

CU6. Consulta estadísticas
Llama al procedimiento para obtener estadísticas. Cuantos usuarios están en cada nivel, el número de intentos que necesita cada usuario para superar el nivel, etc.

**Dpto. de compras y contabilidad.**

CU7 Consulta datos compra ayudas
Consulta de datos para facturación y contabilidad de las compras de ayudas.

CU8 Consulta datos compra vidas
Consulta de datos para facturación y contabilidad de las compras de vidas.

**Dpto. de compras y contabilidad.**

CU9 Cambia password
Cambio de la contraseña del usuario.

<b>Diseño e implementación de una base de datos relacional para la gestión de un videojuego</b>		
<b>Título:</b>		PFC
<b>Referencia:</b>	INF-2013-00001-04	
<b>Modificación:</b>	22/01/2014 17:07	Página 23 de 56

**CU10 Alta plataforma**  
 Registro de la plataforma desde la que accede el usuario.

**CU11 Inicia sesión**  
 Inicia la sesión del jugador, se comprobará que no tiene ya ninguna otra conexión iniciada, que el usuario existe y su contraseña coincide con la almacenada en el sistema.

**CU12 Cierra sesión**  
 Cierra la sesión que tuviera iniciada el jugador.

**CU13 Genera vidas gratuitas**  
 Da de alta las 5 vidas gratuitas concedidas al usuario disponibles durante 24 horas a partir del momento que juega por primera vez o cuando ya le han caducado las anteriores generadas.

**CU14 Consulta vida disponible**  
 Consulta si hay alguna vida disponible para poder jugar

**CU15 Consulta niveles superados**  
 Lista de niveles que el usuario ha superado.

**CU16 Consulta siguiente nivel**  
 Procedimiento que devuelve el valor del siguiente reto a superar teniendo en cuenta los niveles que ha superado el usuario.

**CU17 Consulta ayudas compradas**  
 Lista de todas las ayudas que ha comprado el usuario.

**CU18 Consulta datos nivel**  
 Lista de todas las ayudas que ha comprado el usuario.

**CU19 Lista de ayudas en un nivel**  
 Lista de todas las ayudas disponibles en un nivel y su importe.

**CU20 Pedir una vida**  
 Procedimiento para pedir una vida a otro usuario. Comprobará si se cumplen las condiciones para que se pueda realizar el intercambio.

**CU21 Registra resultados**  
 Registra el resultado de una partida. El nivel jugado y la puntuación obtenida.

<b>Diseño e implementación de una base de datos relacional para la gestión de un videojuego</b>		
<b>Título:</b>		PFC
<b>Referencia:</b>	INF-2013-00001-04	
<b>Modificación:</b>	22/01/2014 17:07	Página 24 de 56



CU22 Comprar vidas  
 Procedimiento para realizar la compra de una o más vidas.

CU23 Compra ayuda  
 Procedimiento para comprar una ayuda.

### 2.3) Diseño Conceptual UML

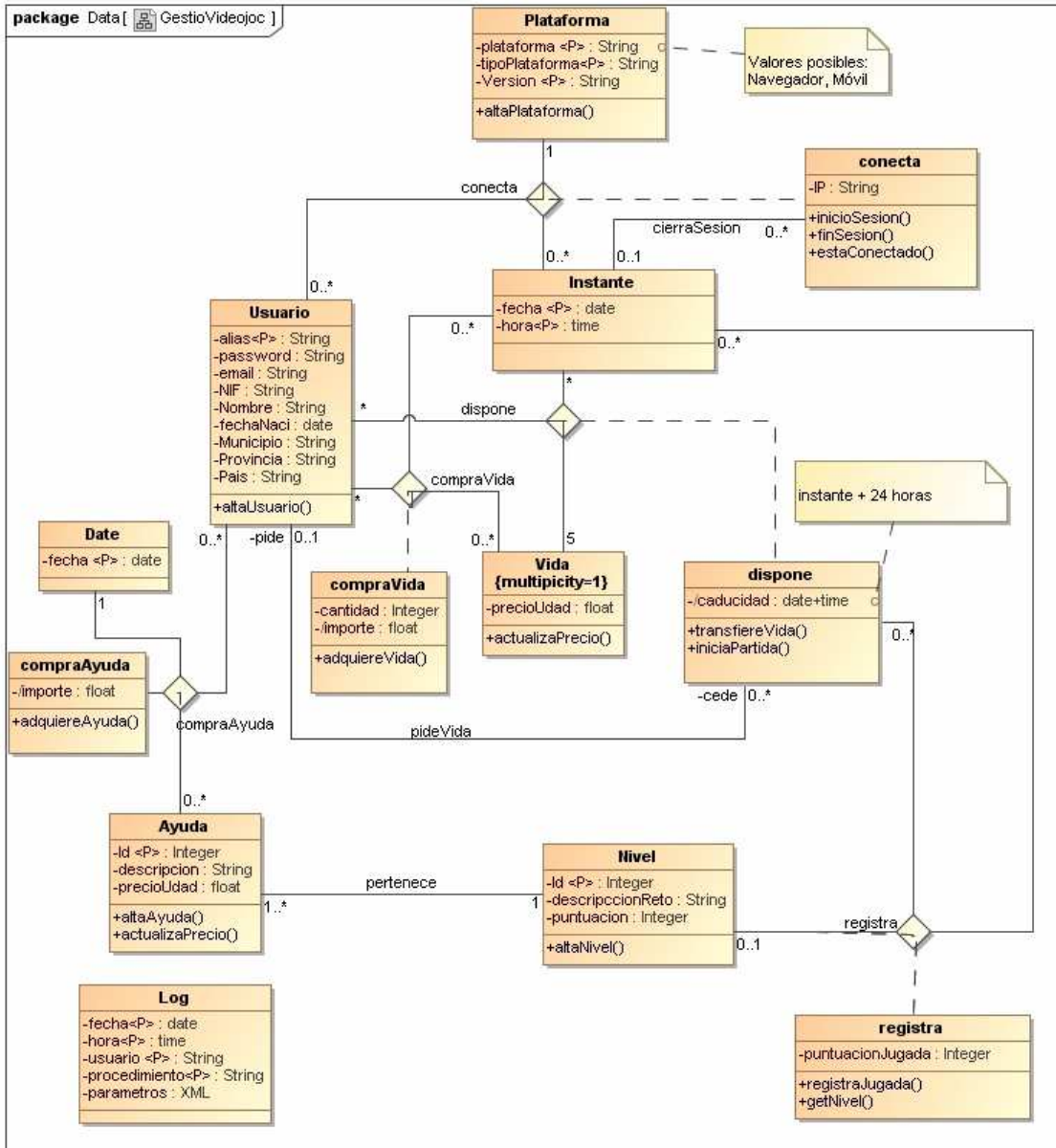


Figura 6. Diagrama UML de la Base de Datos

<b>Diseño e implementación de una base de datos relacional para la gestión de un videojuego</b>		
<b>Título:</b>		PFC
<b>Referencia:</b>	INF-2013-00001-04	
<b>Modificación:</b>	22/01/2014 17:07	Página 25 de 56

## 2.4) Diseño Lógico

Las relaciones se denotan a partir del nombre seguido de la lista de atributos entre paréntesis y separados por comas. Para identificar las claves primarias se subrayan los atributos en línea continua. Las claves foráneas se describen con anotaciones textuales. En negrita se marcan los campos que no son clave pero que no pueden ser nulos.

Para diferenciar la distinta caducidad de las vidas y dar respuesta lo más detallada posible a lo que requiere el enunciado sobre la gestión de niveles, (obtener el número de intentos que necesita cada usuario para superar el nivel), en lugar de utilizar un número distinto por usuario y fecha se decide identificar las vidas por un número único correlativo.

Para evitar errores al recoger los datos de la plataforma por si el valor de versión en algún caso pudiera ser nulo se decide generar un identificador numérico único.

Usuario (alias, **password**, **email**, **NIF**, **nombre**, **apellido1**, **apellido2**, **fechaNacimiento**, **domicilio**, **municipio**, **provincia**, **pais**)

Plataforma (ID, **plataforma**, **tipoPlataforma**, version)

Conecta (usuario, diaInicio, horaInicio, **idPlataforma**, diaFin, horaFin, **IP**)

{usuario} es clave foránea de Usuario

{idPlataforma} es clave foránea de Plataforma

Dispone(idVida , **usuario**, **caducidad**)

{usuario} es clave foránea de Usuario

Vida (**precioUdad**)

PideVida (idVida, usuarioCede, usuarioPide)

{usuarioPide, usuarioCede es clave foránea de Usuario

{idVida} son claves foráneas de Dispone

CompraVida(fechaCompra, horaCompra, usuario, **cantidad**, **importe**)

{usuario} es clave foránea de Usuario

Nivel (id, **reto**, **puntuación**)

Ayuda (id, **nivel**, **descripción**, **precio**)

{nivel} es clave foránea a Nivel

CompraAyuda (usuario, ayuda, **fechaCompra**, **importe**)

{usuario} es clave foránea de Usuario

{ayuda} es clave foránea de Ayuda

<b>Diseño e implementación de una base de datos relacional para la gestión de un videojuego</b>		
<b>Título:</b>		PFC
<b>Referencia:</b>	INF-2013-00001-04	
<b>Modificación:</b>	22/01/2014 17:07	Página 26 de 56

Registra ( fechaJugada, horaJugada, usuario, idVida, nivel, puntuaciónJugada)

{ idVida} es clave foránea de Dispone

{ usuario} es clave foránea de Usuario.

{ nivel} es clave foránea de Nivel

Log(fechaLlamada, horaLlamada, usuario, procedimiento, parametros, resultado)

## 2.5) Diseño Físico de la BD.

Se puede encontrar el código en el fichero adjunto alopezjuan\_PAC2.sql. Utilizo pdf como usuario y esquema en el que generar todos los componentes de la base de datos.

No existe control de seguridad en el acceso de los usuarios, no se bloqueará el usuario aunque haya varios intentos repetidos con la contraseña incorrecta, tampoco un almacenamiento de la password en modo seguro (como por ejemplo un hash MD5), se da por supuesto que para la comunicación se utilizará un sistema encriptado seguro como tunelización, SSH, etc. Si no fuera así, podría mejorarse la seguridad guardando el hash de las contraseñas y no la contraseña como texto plano añadiendo y modificando los procedimientos necesarios para gestionar las conexiones.

Se confía en el servidor para la gestión de los permisos, se entiende que habrá una comunicación protegida entre éste y los procedimientos de la BD y que se configurarán los permisos en el SGBD para que no permita el acceso por ningún otro medio. Como medida de control en todas las llamadas realizadas a las funciones deberá adjuntarse como parámetro el usuario que realiza la acción, para que quede registrado en el Log, y además cumplir con el requisito de escalabilidad de la BD. Se podría implementar un nuevo módulo si fuera necesario, un sistema de autorizaciones mejorado desde la propia base de datos para controlar los permisos desde los propios procedimientos antes de ejecutar ninguna acción.

Por seguridad no se permitirá que un mismo usuario intente iniciar varias sesiones a la vez, la función iniciaSesion devolverá un código de error para que el servidor pueda cerrar la que se encuentra abierta en ese momento, si le conviene, mediante el procedimiento adecuado indicando el valor del atributo horaFin. No se dará respuesta a ninguna petición de ningún usuario (nivel conseguido, solicitud de vida, etc.) que no tenga una sesión iniciada. Podría incrementarse la seguridad si fuera necesario identificando las sesiones y obligando a incluirla en cada petición, pero me ha parecido excesivo y se ha dejado como opción de estudio ante un caso real.

Se comprobará la validez de una vida cada vez que se registre la puntuación de un nivel. Si se registra un nivel con una vida caducada automáticamente se generarán 5 vidas nuevas y se comenzará a utilizar la primera sin necesidad de desconectar la sesión.

<b>Diseño e implementación de una base de datos relacional para la gestión de un videojuego</b>		
<b>Título:</b>		PFC
<b>Referencia:</b>	INF-2013-00001-04	
<b>Modificación:</b>	22/01/2014 17:07	Página 27 de 56

En un entorno real habría que negociar un protocolo para los códigos que devuelven las funciones. Para indicar distintas incidencias, a falta de más información, se ha decidido que las funciones devuelvan:

- 0 si su función es registrar datos y el resultado del proceso es correcto,
- Menor que cero si ha habido algún error (por ejemplo, que si al intentar iniciar la sesión se devuelve -1 indica que el usuario ya tiene una conexión abierta)
- Un valor mayor que cero para los casos de altas (identificador nuevo generado) o consulta de información concreta.

Una opción de mejora a tener en cuenta es que en el sistema de estadísticas se puede introducir un procedimiento que utilice consultas dinámicas para obtener los datos devolviendo un recordset con la información obtenida. El proceso a seguir sería consultar una tabla donde estarían almacenadas las sentencias OPEN SQL que se obtendrían buscando por un identificador y se ejecutaría con "execute immediate" devolviendo su resultado. Esto evitaría tener que modificar el código del procedimiento, sólo habría que agregar nuevas consultas con nuevos identificadores tal y como se van necesitando.

En la tabla Log se registran todas las ejecuciones tanto si su resultado es correcto como fallido. Se insertan los siguientes campos: el timestamp del momento de la ejecución, el usuario pasado como parámetro, el nombre del procedimiento al que se ha llamado, un XML generado con todos los datos de los parámetros de la llamada, y el resultado de la ejecución devuelto por la función.

Se ha optado por utilizar un campo XML ya que varía el tipo y el número de parámetros en cada función, de este modo se puede incorporar todo en la misma columna y seguir teniendo la posibilidad de realizar consultas sobre los datos.

En este caso no representa ningún problema pero en un entorno real podría ser demasiada información y hacer el sistema más lento. Una posible solución para evitar el exceso de tamaño sería añadir otro parámetro de entrada a todas las funciones para indicar el nivel de depuración que se requiere y registrar o no en el log dependiendo de su valor en el momento de la llamada.

## 2.6) Descripción de los Procedimientos y Funciones.

Como no se especifica claramente en el enunciado se han diseñado sólo las funciones consideradas como básicas para el control del juego. Se deja como posible mejora la implementación de los procedimientos para el acceso a los datos desde el departamento de compras o contabilidad.

<b>altaAyuda</b>
Se pasan los parámetros de entrada: nivel, descripción, precio y usuario. Inserta las columnas correspondientes en la tabla Ayuda Genera un nuevo registro en el log. Devuelve el identificador creado para la nueva ayuda.

<b>Diseño e implementación de una base de datos relacional para la gestión de un videojuego</b>		
<b>Título:</b>		PFC
<b>Referencia:</b>	INF-2013-00001-04	
<b>Modificación:</b>	22/01/2014 17:07	Página 28 de 56

**altaNivel**

Se pasan los parámetros de entrada: reto, puntuación y usuario.  
 Inserta las columnas correspondientes en la tabla Nivel  
 Genera un nuevo registro en el log.  
 Devuelve el identificador creado para el nuevo nivel.

**altaPlataforma**

Se pasan los parámetros de entrada: plataforma, tipoPlataforma, version y usuario.  
 Comprueba si no existen ya los datos, inserta las columnas correspondientes en la tabla Ayuda  
 Genera un nuevo registro en el log.  
 Devuelve el identificador creado para la nueva plataforma.

**altaUsuario**

Se pasan los parámetros de entrada: alias del usuario, NIF, password, email, sexo, nombre, apellido1, apellido2, fecha de nacimiento, municipio, provincia, país y domicilio.  
 Comprueba que no existe ningún registro con ese alias, inserta las columnas correspondientes en la tabla usuario.  
 Genera un nuevo registro en el log.  
 Devuelve el resultado del alta (0 si ha ido bien, o el número de usuarios que ya existen con dicho alias, es decir, 1).

**cambiaPrecioAyuda**

Se pasan los parámetros de entrada: identificador de la ayuda, precio nuevo y usuario.  
 Si el precio nuevo es mayor de cero actualiza la columna en la tabla Ayuda  
 Genera un nuevo registro en el log.  
 Devuelve el resultado de la operación: 0 si se ha actualizado correctamente, -1 si el importe no es correcto.

**cambiaPrecioVida**

Se pasan los parámetros de entrada: precio nuevo y usuario.  
 Si el precio nuevo es mayor de cero actualiza la tabla vida  
 Genera un nuevo registro en el log.  
 Devuelve el resultado de la operación: 0 si se ha actualizado correctamente, -1 si el importe no es correcto.

**cambioPassword**

Se pasan los parámetros de entrada: usuario, password anterior y nueva.  
 Comprueba si está conectado, si la contraseña anterior coincide con la que está en la base de datos y actualiza la columna en la tabla usuario.  
 Genera un nuevo registro en el log.  
 Devuelve el resultado de la operación: 0 si se ha actualizado correctamente, -1 si no está conectado, -2 si la contraseña anterior no coincide.

<b>Diseño e implementación de una base de datos relacional para la gestión de un videojuego</b>		
<b>Título:</b>		PFC
<b>Referencia:</b>	INF-2013-00001-04	
<b>Modificación:</b>	22/01/2014 17:07	Página 29 de 56

**cierraSesion**

Se pasa el parámetro de entrada: usuario.  
 Comprueba si el usuario existe, si hay una o más conexiones iniciadas con ese mismo usuario, actualiza la columna cierre en la tabla conecta.  
 Genera un nuevo registro en el log.  
 Devuelve el resultado de la operación: 0 si se ha actualizado correctamente, mayor que 0 si había más de una conexión abierta, -1 si no tenía ninguna conexión iniciada y -2 si el usuario no existe.

**estaConectado**

Se pasa el parámetro de entrada: usuario.  
 Comprueba si hay conexiones iniciadas con ese usuario.  
 Devuelve el resultado de la operación: el número de conexiones abiertas (1 o mayor que 0), -1 si no tenía ninguna conexión iniciada.

**estadísticas**

Se pasan los parámetros de entrada: opción elegida, nivel (0 si no se quiere filtrar por nivel o el identificador del nivel deseado) y usuario.  
 Como parámetro de salida se obtiene un recordset con los campos resultado de la estadística solicitada.  
 Ejecuta la consulta asociada a la opción pasada como parámetro.  
 Registra el log de la ejecución.  
 Devuelve el resultado de la operación: el número de conexiones abiertas (1 o mayor que 0), -1 si no tenía ninguna conexión iniciada.

**generaVidasGratis**

Se pasa el parámetro de entrada: usuario.  
 Comprueba si está conectado, inserta 5 filas en la tabla dispone con el usuario con un identificador de la vida obtenido a partir de la secuencia sq\_Dispone y la caducidad (el instante de la ejecución + 24 horas).  
 Registra el log de la ejecución.  
 Devuelve el resultado de la operación: el número de vidas creadas, -1 si no tenía ninguna conexión iniciada.

**getAyudasUsuario**

Se pasan los parámetros de entrada: usuario y nivel (0 si no se quiere filtrar por nivel o el identificador del nivel deseado).  
 Como parámetro de salida se obtiene un recordset con todas las ayudas compradas por un usuario.  
 Comprueba si está conectado, ejecuta la consulta recogiendo las filas resultado en el parámetro de salida.  
 Registra el log de la ejecución.  
 Devuelve el resultado de la operación: el número de vidas creadas, -1 si no tenía ninguna conexión iniciada.

<b>Diseño e implementación de una base de datos relacional para la gestión de un videojuego</b>		
<b>Título:</b>		PFC
<b>Referencia:</b>	INF-2013-00001-04	
<b>Modificación:</b>	22/01/2014 17:07	Página 30 de 56

**getNextNivelUsuario**

Se pasa el parámetro de entrada: usuario.  
 Comprueba si está conectado, consulta el identificador del siguiente nivel a superar.  
 Devuelve el resultado de la operación: el id del siguiente nivel no superado del usuario, -1 si no tenía ninguna conexión iniciada, -2 si el usuario los ha superado todos.

**getNivelesUsuario**

Se pasa el parámetro de entrada: usuario.  
 Como parámetro de salida se obtiene un recordset con todos los niveles superados de un usuario.  
 Comprueba si está conectado, ejecuta la consulta recogiendo las filas resultado en el parámetro de salida.  
 Registra el log de la ejecución.  
 Devuelve el resultado de la operación: 0 si ha tenido éxito, -1 si no tenía ninguna conexión iniciada.

**getVidaUsuario**

Se pasa el parámetro de entrada: usuario.  
 Se obtienen varios parámetros de salida: el identificador de la vida con la que está jugando, el total de vidas disponibles sin gastar incluyendo la que está en uso, y el siguiente nivel a superar.  
 Comprueba si está conectado, si hay que generarle las vidas gratuitas y ejecuta las consultas para asignar los valores obtenidos a los parámetros de salida.  
 Registra el log de la ejecución.  
 Devuelve el resultado de la operación: el número total de vidas gratuitas (utilizadas o no) que caducan en una fecha posterior al instante de la llamada, -1 si el usuario no tenía ninguna conexión iniciada.

**IniciaSesion**

Se pasan los parámetros de entrada: usuario, password, plataforma, tipoPlataforma, versión y la dirección IP.  
 Comprueba si existe el usuario, si la contraseña coincide, si no está ya conectado, si la plataforma está dada de alta (si no es así llama al procedimiento de alta correspondiente), y registra la conexión insertando una fila en la tabla conecta.  
 Registra el log de la ejecución.  
 Devuelve el resultado de la operación: 0 si se ha ejecutado correctamente, -1 si el usuario no existe, -2 si la contraseña no existe, -3 si ya hay una conexión abierta, -4 si la plataforma no ha podido darse de alta correctamente.

**p\_compraAyuda**

Se pasan los parámetros de entrada: usuario, nivel.  
 Se obtienen los parámetros de salida: descripción de la ayuda comprada y el precio de la ayuda seleccionada.

<b>Diseño e implementación de una base de datos relacional para la gestión de un videojuego</b>		
<b>Título:</b>		PFC
<b>Referencia:</b>	INF-2013-00001-04	
<b>Modificación:</b>	22/01/2014 17:07	Página 31 de 56

Comprueba si el usuario está conectado, selecciona el código de la siguiente ayuda disponible en el nivel elegido que aún no ha sido comprada inserta los datos de la compra en la tabla compraAyuda y ejecuta las consultas para dar valores a los parámetros de salida.

Registra el log de la ejecución.

Devuelve el resultado de la operación: 0 si se ha ejecutado correctamente, -1 si no hay una conexión, -2 si no quedan ayudas disponibles en el nivel seleccionado.

**p\_compraVida**

Se pasan los parámetros de entrada: usuario y el número de vidas que se quiere comprar.

Comprueba si el usuario está conectado, inserta los datos de la compra en la tabla compraVida, y genera un registro por cada vida en la tabla Dispone con fecha de caducidad = null.

Registra el log de la ejecución.

Devuelve el resultado de la operación: 0 si se ha ejecutado correctamente, -1 si no hay una conexión.

**p\_pideVida**

Se pasan los parámetros de entrada: el alias del usuario que pide y el que cede la vida.

Comprueba si el usuario que pide la vida está conectado, si hay que generarle vidas gratuitas porque las que tenía han caducado, si le quedan vidas para poder jugar y si el usuario que cede tiene más de una vida para poder darla. Primero busca una vida gratuita para ceder, si no la encuentra busca una comprada. Actualiza el usuario de la vida seleccionada en la tabla dispone.

Registra el log de la ejecución.

Devuelve el resultado de la operación: el identificador de la vida cedida si se ha ejecutado correctamente, -1 si el usuario que pide no está conectado, -2 el usuario aún dispone de vidas para jugar, -3 y -4 el usuario no tiene vidas suficientes o no encuentra ninguna para poder ceder, -5 si se le han generado vidas gratuitas.

**registraJuego**

Se pasan los parámetros de entrada: usuario y puntuación

Se pasan los parámetros de entrada/salida: Como entrada representarán el nivel jugado, la vida jugada, el total de vidas disponibles. Como salida representarán al siguiente nivel a jugar, la vida disponible para jugar (0 si no le quedan), y el total de vidas disponibles (0 si no le quedan).

Comprueba si el usuario está conectado, inserta en la tabla registro la jugada realizada. Llama al procedimiento getVidaUsuario para dar valores a los parámetros de salida.

Registra el log de la ejecución.

Devuelve el resultado de la operación: el número de vidas que el usuario mantiene de las generadas en la última concesión de vidas gratuitas (tanto gastadas como disponibles y que no han sido cedidas), -1 si no hay una conexión

<b>Diseño e implementación de una base de datos relacional para la gestión de un videojuego</b>		
<b>Título:</b>		PFC
<b>Referencia:</b>	INF-2013-00001-04	
<b>Modificación:</b>	22/01/2014 17:07	Página 32 de 56



## 2.7) Procedimiento para pruebas.

Se ha implementado un script de pruebas que puede dar error al generar claves duplicadas si el equipo es demasiado rápido. Este procedimiento se da de alta con el mismo usuario y en el mismo esquema que la Base de datos (pfc).

A pesar de haber escogido timestamp en lugar de sólo date para que guarde incluso las diferencias en los milisegundos, al ejecutar las pruebas y realizar varias veces seguidas la misma acción con el mismo usuario puede darse el caso de que el procesador sea lo suficientemente rápido como para registrarlo en el mismo milisegundo. Por eso sería conveniente utilizar por ejemplo la instrucción `dbms_lock.sleep(3)` cuando se realizan acciones repetidas, en mi caso he insertado algunas instrucciones con usuarios que no existen y que no van a dar resultados para intentar ralentizarlo un poco. Aunque falle en alguna ocasión suele funcionar cuando se reintentan varias veces.

Por ejemplo: El caso de uso cuando intento registrar puntuaciones en varias partidas seguidas del mismo usuario al ir una instrucción seguida de otra (en la realidad no podrá ocurrir porque no podrá jugar y obtener una puntuación tan rápido como para que coincidan hasta los milisegundos).

<b>Diseño e implementación de una base de datos relacional para la gestión de un videojuego</b>		
<b>Título:</b>		PFC
<b>Referencia:</b>	INF-2013-00001-04	
<b>Modificación:</b>	22/01/2014 17:07	Página 33 de 56

### 3) Almacén de Datos.

#### 3.1) Estudio de los requisitos

Los requisitos no entran mucho en detalle. Según el enunciado:

*“També es voldria definir un magatzem de dades (data warehouse) per extreure estadístiques i saber, per exemple, a quin nivell es compren més vides, quin sexe compra i juga més, quin és el percentatge d'usuaris que gasten diners en el joc,…”*

Los criterios a estudiar se dejan bastante abiertos dando solamente algunas pautas como ejemplo:

- El nivel en el que se compran más vidas
- Qué sexo compra y juega más
- Qué porcentaje de usuarios son los que compran

Siguiendo las pautas marcadas se han extraído cuatro grupos de análisis que serán detallados en el apartado 3.2:

- Análisis de compras: Análisis detallado de las compras realizadas, ya sean vidas o ayudas.
- Análisis de sesiones: estudio del tiempo medio de juego, o de a qué horas se conectan más usuarios.
- Análisis de jugadas.
- Análisis de amistades: que porcentajes de usuarios ceden, cuantos piden, cuantas vidas son compradas cuantas son gratuitas.

Un requisito importante a tener en cuenta si fuera un entorno real es que la BD va a tener un uso constante, su disponibilidad será 24x7 (24 horas 7 días a la semana). Para no interferir en su funcionamiento y evitar algunos problemas provocados por el bloqueo de los registros, se podría estudiar el diseño de una BD operacional como paso previo a la creación del DW.

Se ejecutaría un paso intermedio previo a la carga de datos del DW. La información de la BD principal se duplicaría utilizando por ejemplo desencadenantes de tipo after update, lo que también permitiría traer la obtención de estadísticas básicas a esta BD en lugar de utilizar la principal.

Dado que estamos en un PFC y no se considera necesario no se ha implementado esta solución, los procesos ETL se diseñan sobre la misma BD principal.

<b>Diseño e implementación de una base de datos relacional para la gestión de un videojuego</b>		
<b>Título:</b>		PFC
<b>Referencia:</b>	INF-2013-00001-04	
<b>Modificación:</b>	22/01/2014 17:07	Página 34 de 56

## 3.2) Casos de uso

Caso de uso	Filtro	Descripción
CU24 Análisis de sesión	All	Media de la duración de la conexión
	Plataforma	Número de conexiones según la plataforma (móvil o Navegador) Media de la duración de la conexión según la plataforma
	Edad	Número de conexiones según la edad del usuario Media de la duración de la conexión por edades
	Sexo	Número de conexiones según el sexo del usuario Media de la duración de la conexión por sexos
	País	Número de conexiones según el país del usuario Media de la duración de la conexión por países
	Tiempo	Hora del día donde se realizan más conexiones
CU25 Análisis de amistad	All	Número de vidas cedidas
	Edad	Número de vidas cedidas por edades
	Usuario	% de usuarios que han cedido vidas % de usuario que han pedido vidas
	Sexo	Número de vidas cedidas por sexo % de usuarios que han cedido vidas por sexos % de usuarios que han pedido vidas por sexos
	Nivel	Nivel en el que se piden más vidas
CU26 Análisis de compras	All	Número de compras realizadas Importe total de las compras
	tipoProducto	Número de compras realizadas por tipo de producto Importe de compras realizadas por tipo de producto
	Nivel	Nivel en el que se han realizado más compras Nivel en el que se han gastado más importe
	Sexo	Sexo que ha realizado mas compras Sexo que se ha gastado mas dinero

<b>Diseño e implementación de una base de datos relacional para la gestión de un videojuego</b>		
<b>Título:</b>		PFC
<b>Referencia:</b>	INF-2013-00001-04	
<b>Modificación:</b>	22/01/2014 17:07	Página 35 de 56

	Edad	Número de compras realizadas por edades Importe de compras realizadas por edades
	Usuario	Porcentaje de usuarios que gastan dinero en el juego
CU27 Análisis de jugadas	All	Número de jugadas realizadas
	Edad	Número de jugadas realizadas por edades
	Sexo	Número de jugadas realizadas por sexo
	Nivel	Qué nivel se ha jugado con más vidas compradas
	Tiempo	A qué hora se juega más

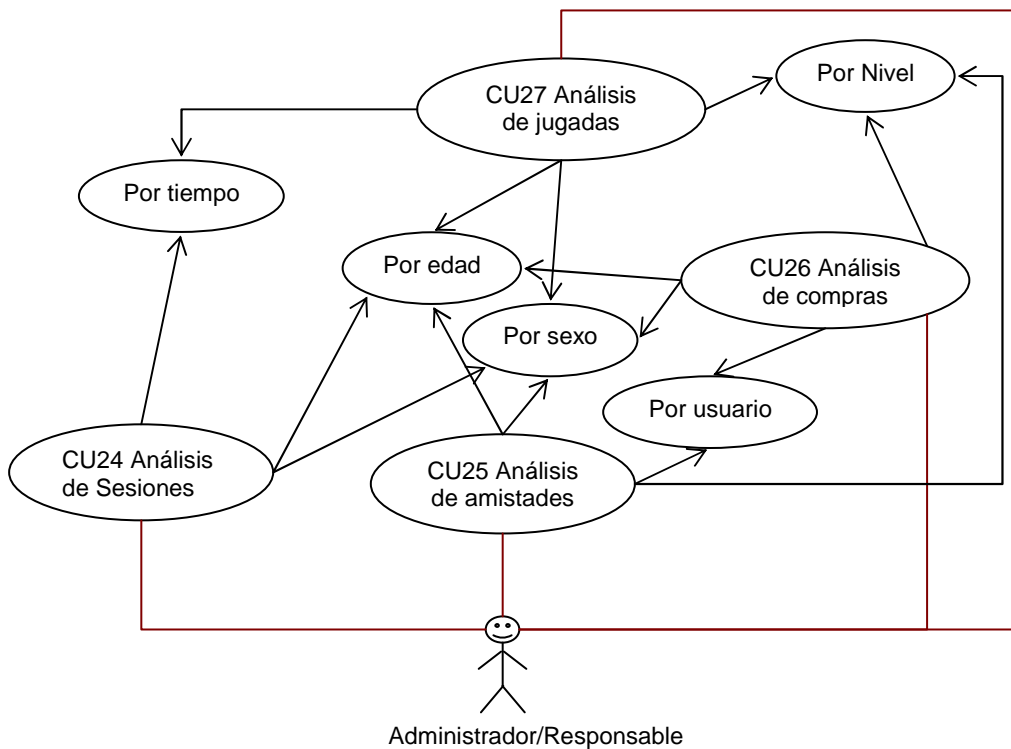


Figura 7. Diagrama de casos de uso del Data Warehouse

Se ha pensado en otro filtro (el de "pais" de los usuarios) para todos los casos de uso que no se indica para no sobrecargar demasiado el diagrama. En un principio no se considera tan interesante para el PFC como las otras opciones, pero en un

entorno real habría que proponerlo y que el administrador/responsable del tratamiento de la información tomara la decisión de añadirlo.

### 3.3) Observaciones

Existen muchas soluciones posibles distintas dependiendo de lo que se quiere analizar, que haría que el diseño aquí planteado tuviera que modificarse. En un entorno real deberíamos consultar al responsable del estudio de la información (Dirección o Marketing, etc.) aclarar cuáles son los requisitos deseados y adaptar el diseño según las necesidades.

Por ejemplo:

En lugar de simplemente tomar el nivel de la compra a partir de la última jugada realizada, (ya sea sobre un nivel superado o no), se podría indicar el primer nivel jugado con esta vida una vez comprada, con lo que habría que añadir un campo fecha a la tabla dispone que almacenara el instante en el que se ha generado la gratuita o en el que se ha comprado para así poder establecer una relación directa entre la jugada y los datos de la compra.

En el hecho compra se podrían indicar tres campos de nivel, el máximo nivel superado, el último nivel fallado previo a la compra, el primer nivel en el que se ha utilizado la vida comprada, etc.....

### 3.4) Decisión de los elementos de análisis

Se han definido los siguientes elementos:

**Hechos:** compras, jugadas, amistades y sesiones

**Dimensiones:** Tiempo, Usuario, Nivel, Plataforma, Conecta, Puntuación, Producto.

**Medidas:** numero\_conexiones y duración, numero\_cesiones, numero\_compras e importe\_compras, numero\_jugadas.

**Atributos:** se ha decidido quitar varios que no se consideran como información relevante, por ejemplo se ha obviado el campo reto en Nivel y la descripción de la ayuda. Además de los que ya vienen definidos por las tablas origen de la BD, se han añadido otros para facilitar el tiempo de respuesta y liberar de cálculos a las consultas.

### 3.5) Diseño Conceptual UML.

Para evitar duplicar dibujos y no extender demasiado el documento, en el diagrama UML, en lugar de indicar solamente la relación entre los nombres de tablas y dimensiones y completarla más adelante, se han incluido ya los atributos y las métricas del diseño lógico que se explican con detalle en el siguiente apartado.

En lugar de un solo diagrama UML, se ha separado en cuatro, uno para cada hecho para que quede más claro repitiendo las mismas dimensiones en aquellos en

<b>Diseño e implementación de una base de datos relacional para la gestión de un videojuego</b>		
<b>Título:</b>		PFC
<b>Referencia:</b>	INF-2013-00001-04	
<b>Modificación:</b>	22/01/2014 17:07	Página 37 de 56

los que coincidan. En todos los hechos se registra un valor de la dimensión tiempo y un valor de la dimensión usuario. La dimensión “Nivel” aparecerá en todos los hechos menos en el de sesión.

El hecho “sesion”, se asocia también a las dimensiones plataforma y conecta.

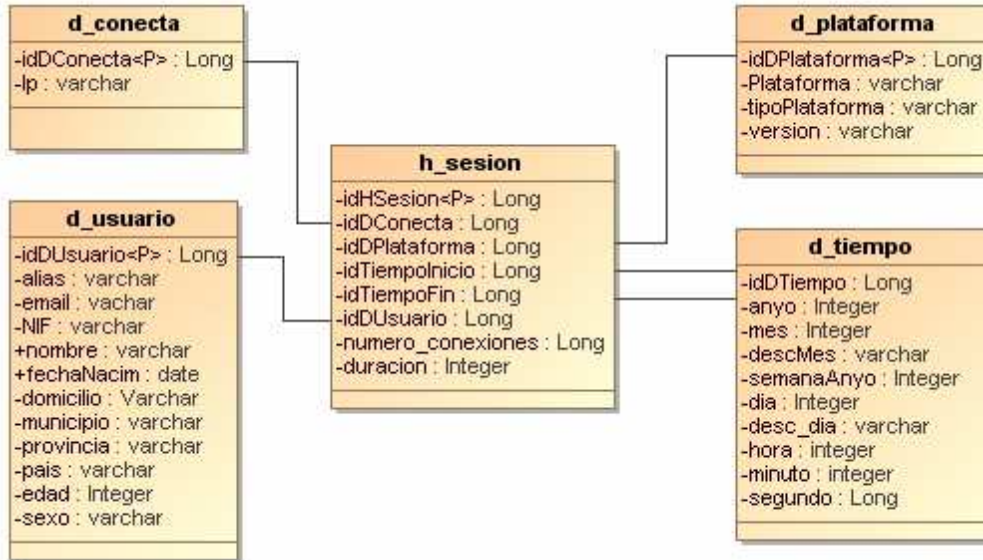


Figura 8. Diagrama UML Data Warehouse Hecho sesion

En el control de las compras se ha decidido unificar las filas de vida y ayudas en una sola tabla denominada “Producto”.

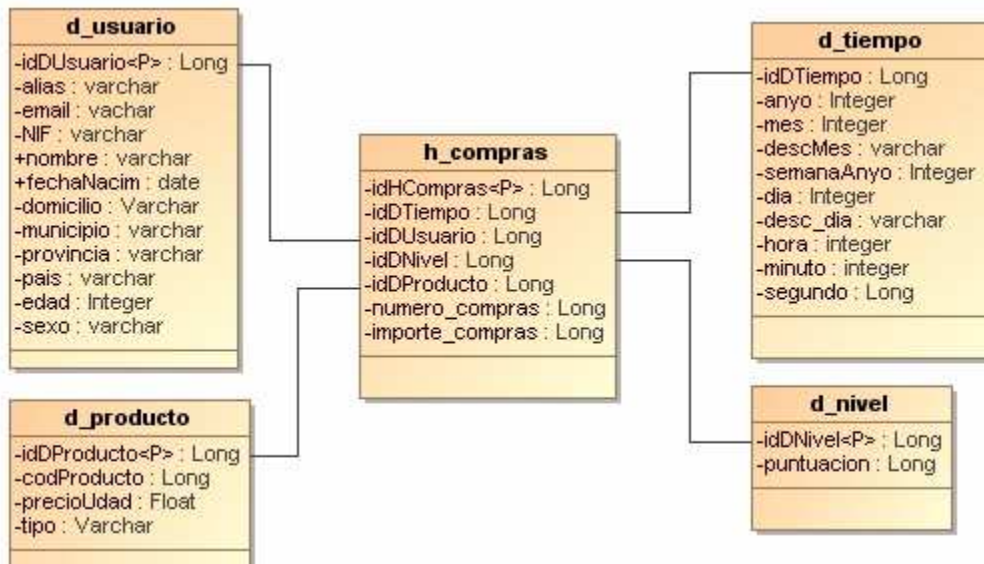


Figura 9. Diagrama UML Data Warehouse Hecho compras

Diseño e implementación de una base de datos relacional para la gestión de un videojuego		
<b>Título:</b>		PFC
<b>Referencia:</b>	INF-2013-00001-04	
<b>Modificación:</b>	22/01/2014 17:07	Página 38 de 56

En el hecho “*amistad*” hay otro valor más de la dimensión usuario, ya que en el proceso intervienen dos jugadores, uno que pide y otro que cede.

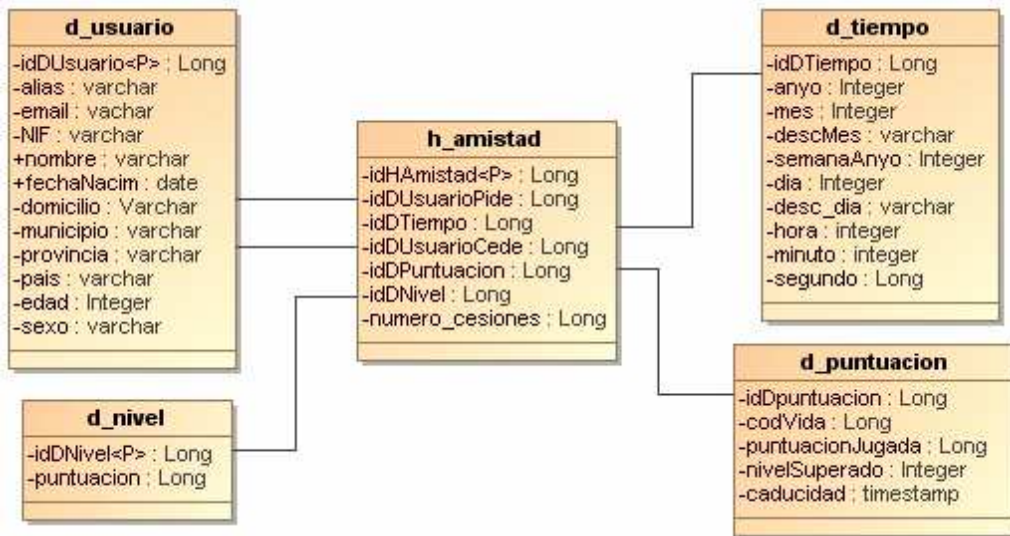


Figura 10. Diagrama UML Data Warehouse Hecho amistad

Tanto en este hecho como en el de “*Jugadas*” tendrán asociados un registro de la tabla de dimensión “*d\_puntuación*”, con las distintas puntuaciones de las vidas registradas.

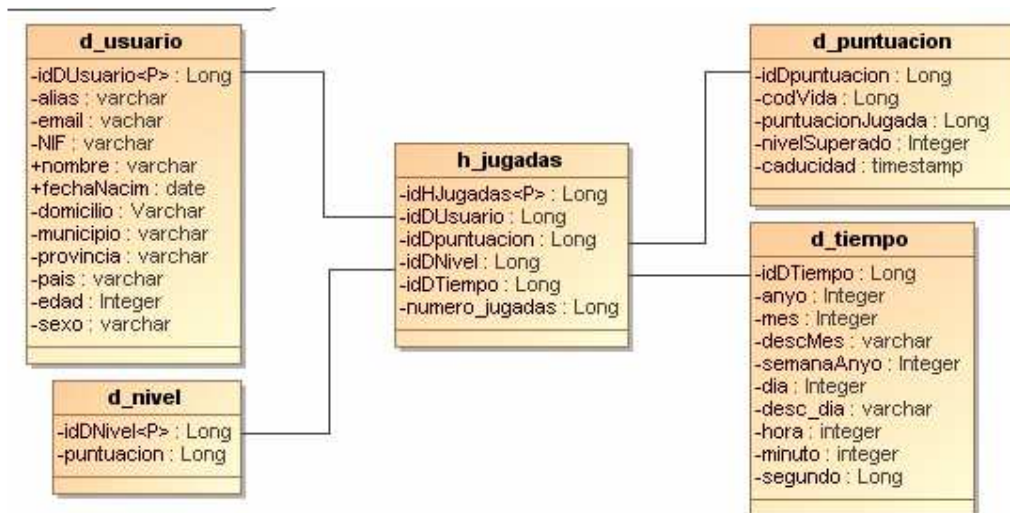


Figura 11. Diagrama UML Data Warehouse Hecho Jugadas

Diseño e implementación de una base de datos relacional para la gestión de un videojuego		
<b>Título:</b>		PFC
<b>Referencia:</b>	INF-2013-00001-04	
<b>Modificación:</b>	22/01/2014 17:07	Página 39 de 56

### 3.6) Diseño Lógico

Tabla de Hecho	Claves foráneas	Métricas
H_sesion	idDConecta idDPlataforma idDTiempoInicio idDTiempoFin idDUsuario	Numero_conexiones
H_amistad	idDFecha idDUsuarioPide idDUsuarioCede idDNivel idDPuntuacion	Numero_cesiones
H_compras	idDFecha idDUsuario idDNivel idDProducto	Numero_compras Importe_compras
H_jugadas	idDUsuario idDPuntuacion idDNivel idDFecha	Numero_jugadas

Tabla de Dimensión	Clave primaria	Atributos	Jerarquía
D_plataforma	idDPlataforma	Plataforma tipoPlataforma version	Plataforma → tipoPlataforma → version
D_conecta	idDConecta	IP	IP
D_usuario	idDUsuario	Alias email NIF nombre edad sexo domicilio municipio provincia pais	All → Edad All → Sexo All → Pais
D_tiempo	idDTiempo	Anyo mes Desc_mes semanaAnyo dia Desc_dia hora minuto segundo	Anyo → mes → semana → día → hora → minuto → segundo
d_nivel	idDNivel	Puntuación	Puntuación
D_producto	idDProducto	codProducto precioUdad tipo	Tipo → codproducto → precioUdad
D_puntuacion	idDPuntuacion	Codvida caducidad puntuacionjugada nivelSuperado	Caducidad → Codvida → puntuacionjugada → nivelSuperado

En la tabla Tiempo se han separado los distintos valores de la fecha y se han incluido otros campos, (semana, desc\_mes, etc.) para poder obtener las distintas jerarquías.

En el diseño de la tabla *d\_puntuacion* se ha decidido incluir el campo nivelSuperado: en un principio iba a especificarse como un booleano, pero como no existe dicho tipo en Oracle, se ha considerado que se da más información si se almacena el resultado de restar la puntuación obtenida en la jugada menos la puntuación mínima para superar el nivel. Si el contenido del campo es negativo no se



ha superado, si es positivo o 0 sí y además proporciona una idea del margen en el que se ha movido el usuario en cada jugada.

En la tabla de Producto se han unificado las filas de vida y ayudas y se ha agregado a la tabla un campo llamado “*tipo*” para diferenciarlas, aunque sólo hay un producto de tipo vida (se le asigna el código 1), para evitar duplicidad al unificar las dos tablas y al registrar las filas con los mismos códigos por cambios en el precio se ha generado un identificador único para la tabla de la dimensión.

En la tabla Usuario se ha generado un campo nuevo llamado edad que se calcula en el momento de realizar la carga, habría que plantear al usuario del DW si desea ir actualizando el campo sobrescribiendo la misma fila o por el contrario desea generar nuevas filas para cada edad mientras siga jugando.

Se ha decidido tomar como nivel asociado el de la última jugada previa más cercana al momento de la compra. Lo más razonable es que se necesite comprar vidas porque en la última jugada el usuario ha fallado y se ha quedado sin vidas disponibles, pero no hay restricciones a la hora de comprar y se pueden adquirir una o más vidas en cualquier momento. Una compra puede no tener ningún nivel previo registrado por lo tanto a la hora de generar los datos de la dimensión “*nivel*” habrá que incluir, el valor 0, por ejemplo, para no incumplir con la restricción de clave foránea.

Ya que la clave de nivel en la BD origen es un “autonumérico” y que no habrá modificaciones se ha decidido utilizar el mismo número como identificador de la tabla de dimensión destino “*Nivel*”. Podría agregarse un id específico al igual que en el resto de dimensiones si al final se permitiera cambiar algún dato de los niveles.

### 3.7) Diseño Físico de la BD.

Se adjunta fichero SQL con los procesos de creación de las tablas de hechos, dimensiones y las restricciones.

Se ha utilizado Oracle 11g igual que para la BD de la PAC 2, y se ha implementado sobre el esquema de un nuevo usuario denominado MD, con contraseña MD.

Hay columnas que están indexadas por ser claves foráneas o claves principales de las tablas. Sería interesante estudiar qué tipo de consultas se van a realizar con más asiduidad y añadir algún índice también para mejorar la respuesta de las consultas, por ejemplo, a los campo “*edad*” y “*sexo*” de la tabla usuario, a los campos “*hora*” y “*Desc\_mes*” de la tabla tiempo, al campo nivelSuperado de la dimensión “*Puntuacion*”, etc..

<b>Diseño e implementación de una base de datos relacional para la gestión de un videojuego</b>		
<b>Título:</b>		PFC
<b>Referencia:</b>	INF-2013-00001-04	
<b>Modificación:</b>	22/01/2014 17:07	Página 41 de 56

### 3.8) Diseño de procesos ETL

Se han utilizado los datos de prueba de la BD denominada PFC creada con el usuario pfc (password pfc) disponibles el fichero sql de creación de la BD y en el del creación del procedimiento de prueba.

La base de datos generada para el DW en Oracle 11g (esquema, usuario y contraseña MD) se ha cargado con los datos obtenidos mediante procesos ETL diseñados en PDI. Dada la finalidad de la BD no hay un momento del día en el que se sepa que va a haber menor carga de trabajo, así que el proceso se podría programar a una hora del día cualquiera, o ejecutarse justo en el momento previo a obtener los análisis deseados. Dependiendo de lo que interese tener actualizados los cambios en los datos de los usuarios, domicilio, edad, etc.

A continuación se describen todos los ficheros creados para llevar a cabo los procesos ETL que se pueden encontrar en el fichero zip de la entrega.

**JOB\_CARGA\_DW**

Fichero de tipo job que puede ser programado a una hora concreta del día para lanzar todos los procesos necesarios.

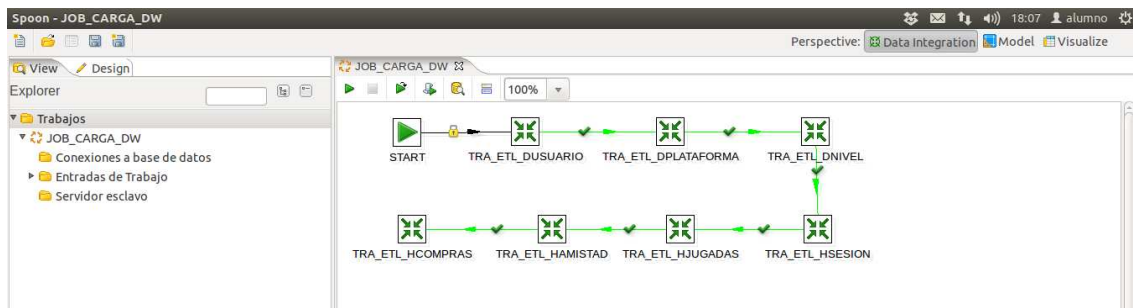


Figura 12. Contenido del fichero JOB\_CARGA\_DW

En este fichero se establece el orden de ejecución de los distintos procesos ETL (ubicados en los ficheros ktr). Los tres primeros ficheros que cargan las dimensiones usuario, plataforma y nivel pueden intercambiarse, pero deben ejecutarse siempre en primer lugar seguido de los procesos de carga de los cuatro hechos. Estos cuatro ficheros ETL también pueden intercambiar su orden porque no interfieren unos con otros.

**TRA\_ETL\_DUSUARIO**

Se dan de alta los datos de los usuarios

Se diseña en un fichero ktr aparte para que no haya que modificar el resto de los procesos si se decide cambiar la forma de tratar las actualizaciones de los datos de edad, domicilio, etc.

<b>Diseño e implementación de una base de datos relacional para la gestión de un videojuego</b>		
<b>Título:</b>		PFC
<b>Referencia:</b>	INF-2013-00001-04	
<b>Modificación:</b>	22/01/2014 17:07	Página 42 de 56

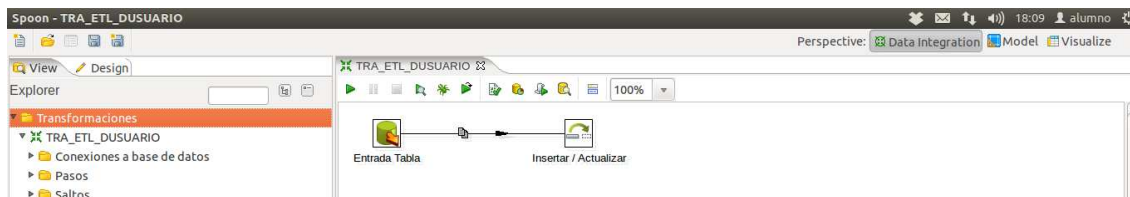


Figura 13. Contenido del fichero TRA\_ETL\_DUSUARIO

**Entrada Tabla**      Selecciona los datos a cargar en la dimensión usuario.  
 Extrae los campos de la tabla usuario y calcula el valor para el campo edad.

**Insertar/Actualizar**      Agrega a la tabla del hecho aquellos registros que no han sido ya insertados en ejecuciones previas del proceso

**TRA\_ETL\_DPLATAFORMA**  
 Se añade a la dimensión plataforma los nuevos datos dados de alta desde el último proceso de carga

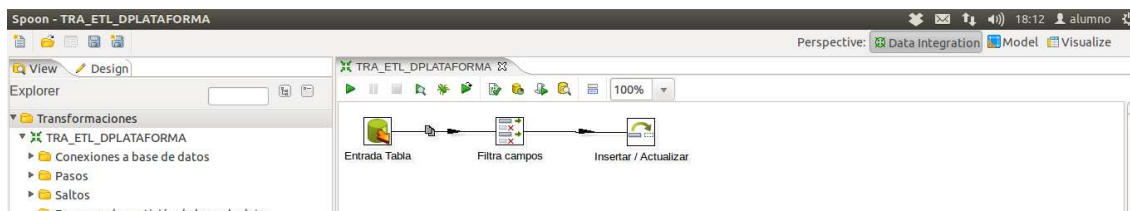


Figura 14. Contenido del fichero TRA\_ETL\_DPLATAFORMA

**Entrada Tabla**      Selecciona los datos a cargar en la dimensión plataforma.  
 Extrae todos los campos de la tabla plataforma.

**Filtrar campos**      Renombra el campo id de la tabla origen a idDPlataforma de la tabla destino.

**Insertar/Actualizar**      Agrega a la tabla d\_plataforma aquellos registros que no han sido ya insertados en ejecuciones previas del proceso

**TRA\_ETL\_DNIVEL:**  
 Se dan de alta los niveles.

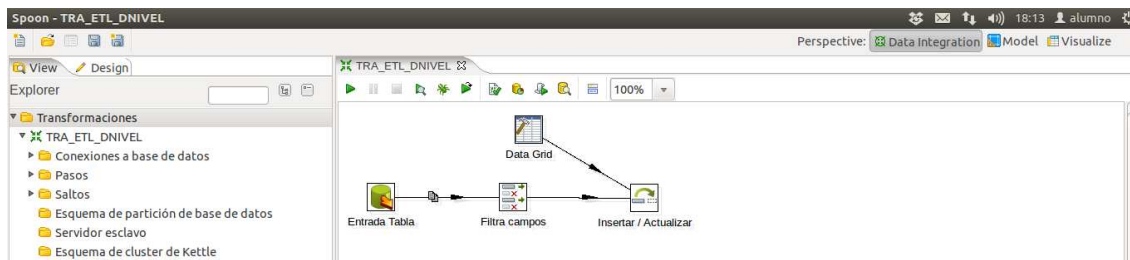


Figura 15. Contenido del fichero TRA\_ETL\_DNIVEL

<b>Diseño e implementación de una base de datos relacional para la gestión de un videojuego</b>		
<b>Título:</b>		PFC
<b>Referencia:</b>	INF-2013-00001-04	
<b>Modificación:</b>	22/01/2014 17:07	Página 43 de 56

- Data Grid Se asignan valores a cero para cada columna de d\_nivel. Así se podrán dar de alta hechos que no estén relacionados con ningún nivel concreto
- Entrada Tabla Selecciona los datos a cargar en la dimensión nivel. Extrae los campos id y puntuación de la tabla nivel.
- Filtra campos Renombra el campo id de la BD al campo idDNivel del DW.
- Insertar/Actualizar Agrega a la tabla d\_nivel aquellos registros que no han sido ya insertados en ejecuciones previas del proceso

Estos dos últimos procesos mencionados se encuentran en ficheros aparte porque los datos no van a sufrir cambios una vez cargados.

Una vez finalizados los procesos de inserción de registros de las tres dimensiones, el resto de las dimensiones pendientes de actualizar se irán cargando conforme se vayan registrando los hechos.

**TRA\_ETL\_HSESSION:**  
Completa el Hecho Sesión.

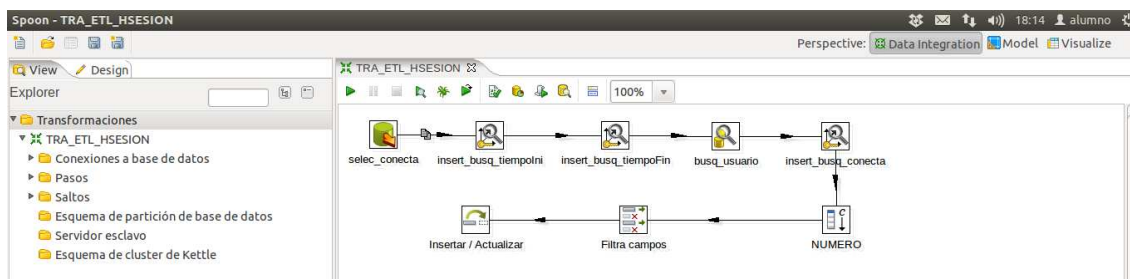


Figura 16. Contenido del fichero TRA\_ETL\_HSESSION

- selec\_conecta Selecciona los campos de la tabla conecta, y calcula el valor en minutos para el campo de la métrica duración = fecha cierre sesión – fecha inicio. Filtra los registros eligiendo sólo las sesiones que hayan terminado, no se escogen las que siguen abiertas (*cierre is not null*). Divide los campos timestamp de inicio y fin de sesión en sus componentes individuales para poder trabajar con la tabla de dimensión tiempo.
- Insert\_busq\_tiempoini Busca en la tabla d\_tiempo el idDTiempo correspondiente al instante de inicio de la sesión. Inserta el timestamp en la dimensión si no se encuentra.
- Insert\_busq\_tiempoFin Busca en la tabla d\_tiempo el idDTiempo correspondiente al instante de fin de la sesión. Inserta el timestamp en la dimensión si no se encuentra.

<b>Diseño e implementación de una base de datos relacional para la gestión de un videojuego</b>		
<b>Título:</b>		PFC
<b>Referencia:</b>	INF-2013-00001-04	
<b>Modificación:</b>	22/01/2014 17:07	Página 44 de 56

Busq_usuario	Obtiene el idDUusuario de la tabla d_usuario.
Insert_busq_conecta	Busca la IP en la tabla conecta, obtiene el idDConecta. Da de alta el registro si no lo encuentra.
Numero	Se da valor (1) al campo de la métrica numero_conexiones
Filtra campos	Filtra los campos a insertar en la tabla del Hecho.
Insertar/Actualizar	Agrega a la tabla del hecho aquellos registros que no han sido ya insertados en ejecuciones previas del proceso

**TRA\_ETL\_JUGADAS:**  
Completa el Hecho Jugadas.

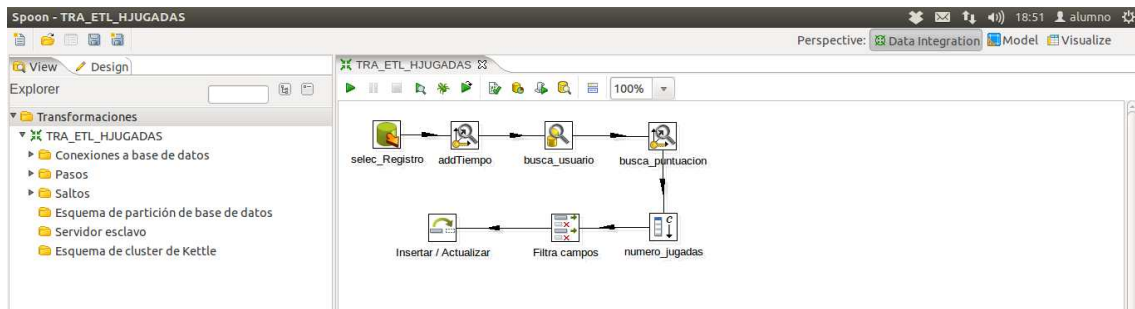


Figura 17. Contenido del fichero TRA\_ETL\_HJUGADAS

selec_Registro	Selecciona todos los campos de la tabla registro. Busca el campo caducidad de la vida utilizada en la tabla dispone. Calcula el valor para el campo nivelSuperado restando la puntuación mínima para superar el nivel jugado a la puntuación obtenida en la partida. Divide el campo timestamp en sus componentes individuales para poder trabajar con la tabla de dimensión tiempo.
addTiempo	Busca en la tabla d_tiempo el idDTiempo correspondiente al instante de inicio de la sesión, inserta el timestamp en la dimensión si no se encuentra.
Busca_usuario	Obtiene el idDUusuario de la tabla d_usuario.
busca_puntuacion	Busca los datos coincidentes en la tabla de la dimensión puntuación y obtiene el idDPuntuacion. Inserta los datos si no se encuentran en la tabla.
numero_jugadas	Se da valor (1) al campo de la métrica numero_jugadas
Filtra campos	Filtra los campos a insertar en la tabla del Hecho.
Insertar/Actualizar	Agrega a la tabla del hecho aquellos registros que no han sido ya insertados en ejecuciones previas del proceso

**TRA\_ETL\_AMISTAD:**  
**Completa el Hecho Amistad.**

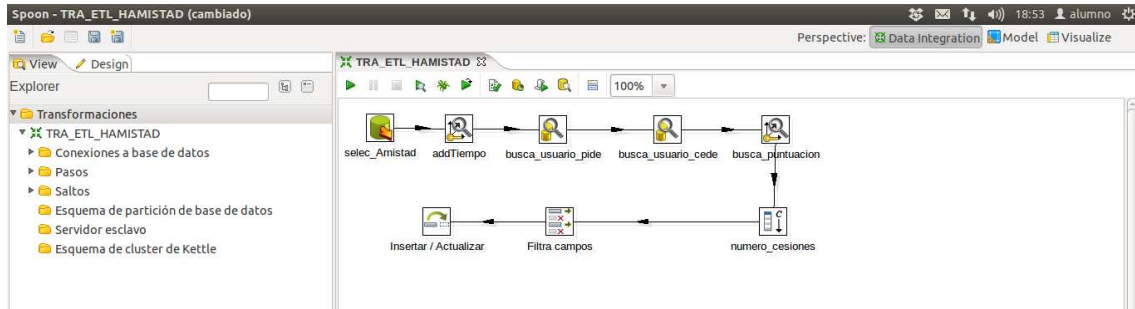


Figura 18. Contenido del fichero TRA\_ETL\_HAMISTAD

**selec\_Amistad**

Selecciona los campos de las tablas pideVida y registro. Filtra la primera tupla de la tabla registro donde se ha utilizado la vida cedida para saber el nivel en el que se ha solicitado. Divide el campo fecha registro en sus componentes individuales para poder trabajar con la tabla de dimensión tiempo. Busca el campo caducidad de la vida utilizada en el registro. Calcula el valor para el campo destino nivelSuperado restando la puntuación mínima para superar el nivel jugado a la puntuación obtenida en la jugada.

**addTiempo**

Busca en la tabla d\_tiempo el idDTiempo correspondiente al instante de inicio de la sesión e inserta el timestamp en la dimensión si no se encuentra.

**Busca\_usuario\_pide**

Obtiene el idDUsuario de la tabla d\_usuario del que pide la vida.

**Busca\_usuario\_cede**

Obtiene el idDUsuario de la tabla d\_usuario del que cede la vida.

**busca\_puntuacion**

Busca los datos coincidentes en la tabla de la dimensión puntuación. Obtiene el idDPuntuacion. Inserta los datos si no se encuentran en la tabla.

**numero\_cesiones**

Se da valor (1) al campo de la métrica numero\_cesiones

**Filtra campos**

Filtra los campos a insertar en la tabla del Hecho.

**Insertar/Actualizar**

Agrega a la tabla del hecho aquellos registros que no han sido ya insertados en ejecuciones previas del proceso

<b>Diseño e implementación de una base de datos relacional para la gestión de un videojuego</b>		
<b>Título:</b>		PFC
<b>Referencia:</b>	INF-2013-00001-04	
<b>Modificación:</b>	22/01/2014 17:07	Página 46 de 56

**TRA\_ETL\_COMPRAS:**  
**Completa el Hecho Compras**

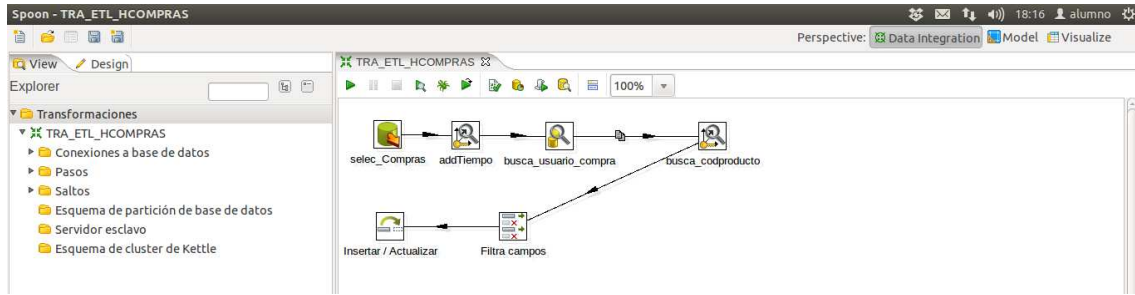


Figura 19. Contenido del fichero TRA\_ETL\_HCOMPRAS

**selec\_Compras**

Se ejecuta una consulta de unión con las tablas compraAyuda y compra\_Vida.

Se da valor al campo tipo para indicar si es ayuda o vida.

Selecciona los siguientes campos:

- Fechacompra (en ambas tablas), divide el timestamp en sus componentes individuales para poder trabajar con la tabla de dimensión tiempo.
- El nivel (campo nivel en la tabla Ayuda o el nivel de la jugada anterior a la compra de la vida será 0 si el usuario compra antes de haber jugando ningún nivel)
- El identificador (código de la ayuda o el valor 1 si es una vida)
- El importe de la compra, el número de unidades compradas (1 si es ayuda)
- El usuario de la compra (en ambas tablas)
- El precio unitario (importe en la ayuda y para la vida importe/udes compradas)

**addTiempo**

Busca en la tabla d\_tiempo el idDTiempo correspondiente al instante de inicio de la sesión.

Inserta el timestamp en la dimensión si no se encuentra.

**busca\_usuario\_compra**

Obtiene el idDUsuario de la tabla d\_usuario.

**busca\_codproducto**

Busca los datos coincidentes tanto de vidas como de ayudas en la tabla de la dimensión producto.

Obtiene el idDProducto. Inserta los nuevos registros si es necesario.

**Filtra campos**

Filtra los campos a insertar en la tabla del Hecho.

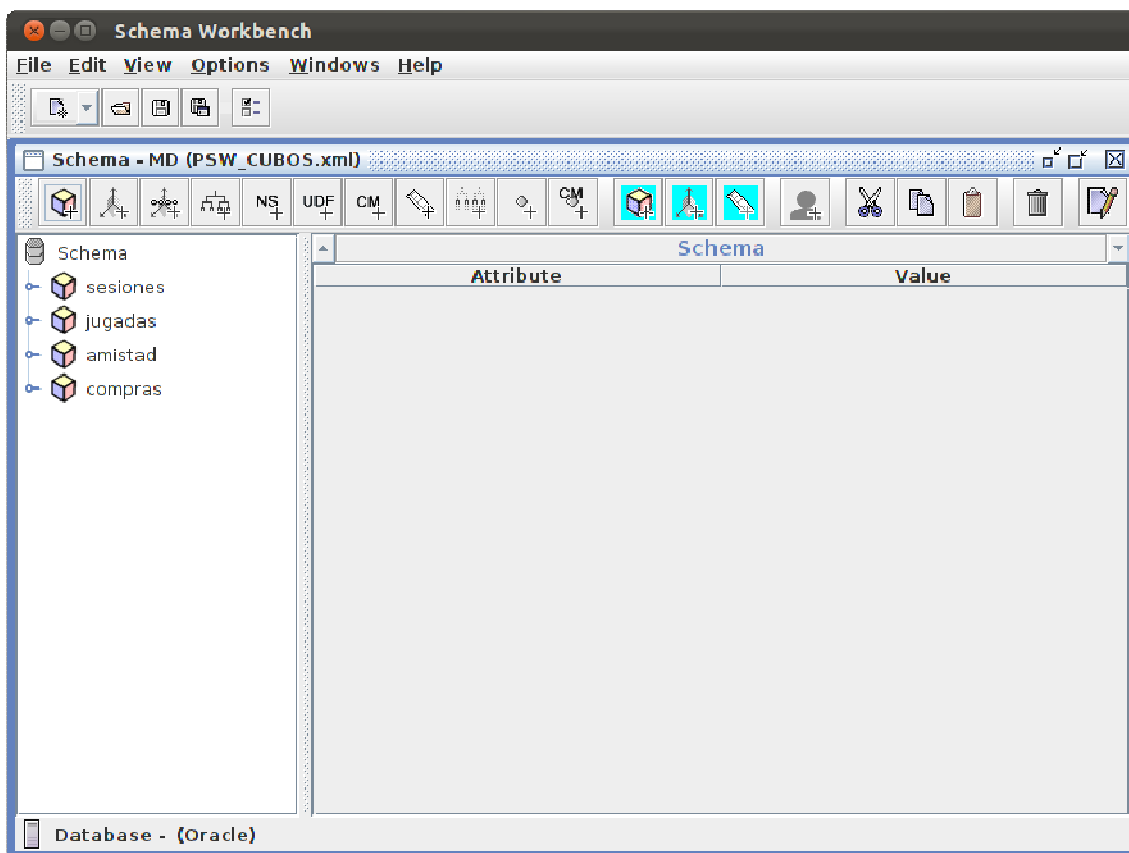
**Insertar/Actualizar**

Agrega a la tabla del hecho aquellos registros que no han sido ya insertados en ejecuciones previas del proceso

<b>Diseño e implementación de una base de datos relacional para la gestión de un videojuego</b>		
<b>Título:</b>		PFC
<b>Referencia:</b>	INF-2013-00001-04	
<b>Modificación:</b>	22/01/2014 17:07	Página 47 de 56

### 3.9) Pruebas

Se ha generado un cubo con la utilidad PSW para publicarlo en pentaho y poder realizar una pequeña prueba de consulta. Se adjunta el fichero xml.



Se han realizado unos análisis de ejemplo de los que se adjuntan las siguientes capturas de pantalla.

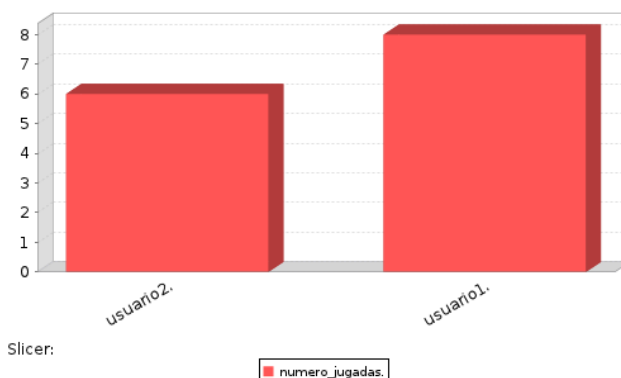


Figura 20. Análisis hecho jugadas numero\_jugadas por Usuario.

<b>Diseño e implementación de una base de datos relacional para la gestión de un videojuego</b>		
<b>Título:</b>		PFC
<b>Referencia:</b>	INF-2013-00001-04	
<b>Modificación:</b>	22/01/2014 17:07	Página 48 de 56



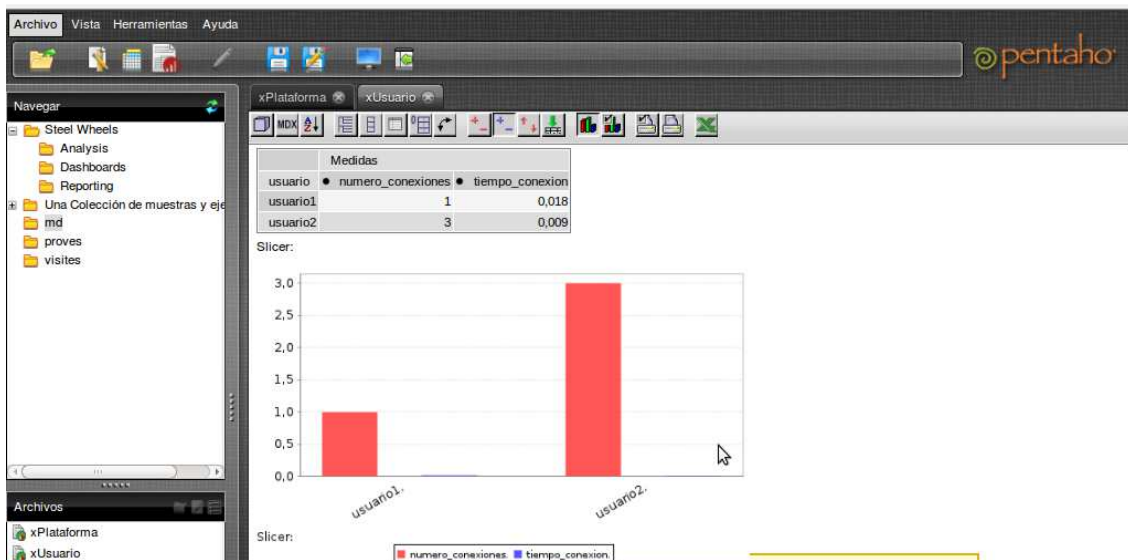


Figura 21. Análisis hecho sesión métricas por usuario:

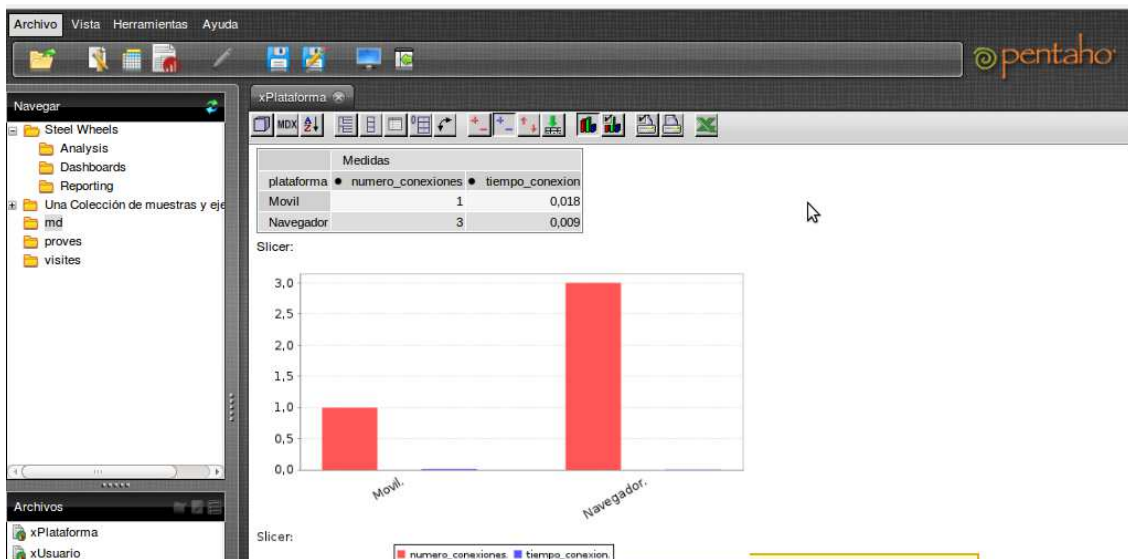
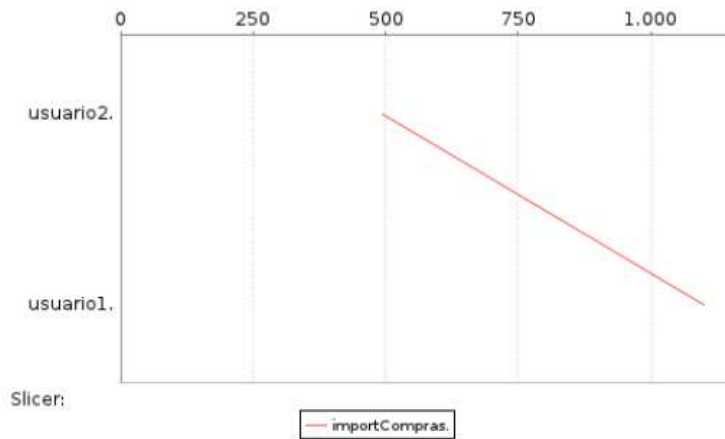


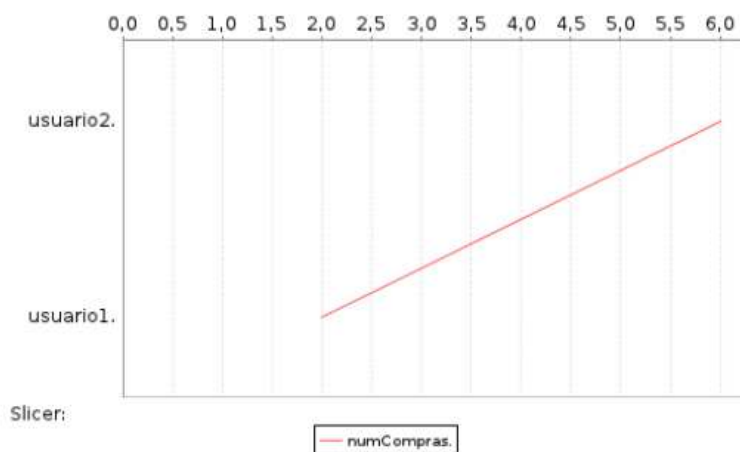
Figura 22. Análisis hecho sesión. Métricas por Navegador.

Diseño e implementación de una base de datos relacional para la gestión de un videojuego		
<b>Título:</b>		PFC
<b>Referencia:</b>	INF-2013-00001-04	
<b>Modificación:</b>	22/01/2014 17:07	Página 49 de 56



	Medidas
usuario	importCompras
usuario2	495,01
usuario1	1.100

Figura 23. Análisis hecho compras importe por Usuario.



	Medidas
usuario	numCompras
usuario2	6
usuario1	2

Figura 24. Análisis hecho compras numero\_compras por Usuario.

## 4) Valoración Económica

### 4.1) Clasificación de Tareas

Para definir los posibles actores y precios se han tomado como referencia los vistos en otros proyectos.

Actor	Función	Plan	Real
Director de proyectos	Descarga de documentación y materiales	2	4
	Lectura Documento TFC	1	1
	Recopilación información adicional (otros proyectos, materiales de otras asignaturas)	3	2,5
	Análisis de requisitos del proyecto. EDT	1	1
	Definir actividades a llevar a cabo	4	3
	Establecer la secuencia de realización	1	1
	Estimar el esfuerzo y carga de trabajo requerida	2	2,5
	Establecer el calendario de trabajo provisional	2	2
	Evaluar riesgos y establecer contingencias	2	1
	Establecer el presupuesto provisional	1	1
	Redacción de documentación y entrega PAC1	9	11
	Redacción de documentación y entrega PAC2	10	12,5
	Redacción de documento y entrega PAC3	10	21
	Redacción de la memoria	10	36,25
	Diseño de la Presentación	10	16,25
	Revisión de la documentación y entrega Final	3	5
	Consulta del foro	4	1,5
	Seguimiento y Control resultados PAC1 y Replanificación	4	0,25
	Seguimiento y Control resultados PAC2 y Replanificación	4	1,75
	Seguimiento y Control resultados PAC3 y Replanificación	4	2,5
Analista	Lectura Documentación adicional	4	2,75
	Decisión del software a utilizar BD	0,5	0,5
	Diseño conceptual UML BD	9	14,25
	Diseño Lógico BD	5	5,5
	Diseño Físico (Tablas, triggers, y procedimientos)	2	6,75
	Decisión del software a utilizar DW	0,5	0,25
	Análisis funcional de los requisitos del DW	5	2
	Identificación de indicadores, dimensiones, atributos DW	5	3,5
	Diseño conceptual del modelo multidimensional	10	9,5
	Diseño Lógico del modelo multidimensional	5	2
	Diseño de procesos ETL	4	3
Diseño Físico DW	4	3	
Programador	Instalación de material	3	1
	Instalación del software a utilizar BD	2,5	0,5
	Consulta manual software instalado BD	3	3,75
	Diseño Físico (Tablas, triggers, y procedimientos)	3	8,5
	Pruebas y refinamiento de la BD	4	10
	Instalación del software a utilizar DW	2,5	1
	Consulta manual software instalado DW	5	2,5

<b>Diseño e implementación de una base de datos relacional para la gestión de un videojuego</b>		
<b>Título:</b>		PFC
<b>Referencia:</b>	INF-2013-00001-04	
<b>Modificación:</b>	22/01/2014 17:07	Página 51 de 56

construcción del Almacén de datos y los procesos ETL	5	7,5
Pruebas y reparación del DW	5	16,25
<b>Total</b>	<b>174</b>	<b>231</b>

Tabla con los importes/hora de cada perfil:

Perfil	Precio/Hora
Director de proyectos	55 €
Analista	46 €
Programador	32 €

## 4.2) Presupuesto estimado.

Para el desarrollo del proyecto todo el software lo proporciona la UOC, habría que estudiar si incluir parte del coste de la matrícula pero se descarta la idea porque no se pagan los materiales. No se pide que se instale en ningún servidor, el ordenador fue comprado a principios de año para las asignaturas del segundo semestre del curso 2012-2013 y el material adicional son documentos e información descargada desde Internet. No es necesario viajar para llevarlo a cabo.

En materiales considero que no hay gasto económico. En un entorno real habría que incluir el coste de los programas, el precio del hardware y otras facturas como las del proveedor de Internet, medios de transporte, hoteles, etc.

El presupuesto de este proyecto se calcularía a partir de la planificación realizada por los distintos actores

Tabla de precios estimada:

Perfil	Horas Plan	Precio/Hora	Importe
Director de proyectos	87	55 €	4.785 €
Analista	54	46 €	2.484 €
Programador	33	32 €	1.056 €
<b>Total</b>	<b>174</b>		<b>8.325 €</b>

## 4.3) Presupuesto real.

El total de horas calculado para la PAC2 fueron 39, en realidad se han empleado en completar todas las tareas un total de 65 horas, incluyendo media hora de formación y se ha entregado un día después de lo previsto inicialmente

El total de horas calculado para la PAC3 fueron 61, este cálculo se ha ajustado bastante a la realidad ya que se han empleado en completar todas las tareas un total de 67 horas, incluyendo 1 hora de formación y 1 hora y media de seguimiento y control. Se ha entregado dos días después de lo previsto inicialmente, esta fecha límite se modificó tras ver la cantidad de horas invertidas en la PAC 2 (casi el doble de lo

<b>Diseño e implementación de una base de datos relacional para la gestión de un videojuego</b>		
<b>Título:</b>		PFC
<b>Referencia:</b>	INF-2013-00001-04	
<b>Modificación:</b>	22/01/2014 17:07	Página 52 de 56

inicialmente planificado) que daban una idea de lo irreales que podían llegar a ser en este proceso de creación del DW.

Se reservaron 23 horas para la Entrega Final, cuando en realidad se han invertido 59. Es un desfase considerable que ha podido ser absorbido quitando horas de otras actividades programadas menos importantes. Si este caso hubiera sido un caso real se habría perdido mucho dinero.

Perfil	Horas Reales	Precio/Hora	Importe
Director de proyectos	127	55 €	6.985 €
Analista	53	46 €	2.438 €
Programador	51	32 €	1.632 €
<b>Total</b>	<b>231</b>		<b>11.055 €</b>

Se ha terminado el proyecto con una desviación de importe del 33%, 2.730 €

## 5) Conclusiones.

Esta asignatura me ha aportado más experiencia porque permite abordar un proyecto desde todas las áreas, Gestión, Análisis, Diseño y Programación. No es muy habitual hoy en día, al menos en mi caso, ya que normalmente las funciones se encuentran muy repartidas entre varias personas y cada una suele tener el mismo rol en todos los proyectos.

Las tareas del Plan de trabajo se han diseñado como si se ejecutaran individualmente una tras otra aunque en la práctica algunas de ellas se han llevado a cabo simultáneamente. En cada entrega se ha generado un prototipo inicial y se ha revisado y modificado progresivamente hasta alcanzar un producto lo más preciso posible.

Se ha intentado crear un plan lo más ajustado a la realidad posible, y aunque los tiempos calculados (especialmente en la PAC2) han sido erróneos, al final se ha logrado entregar en cada PAC los productos marcados.

Agradezco a todos la atención prestada, muchas gracias Juan por responder a todas las preguntas formuladas. Me ha gustado mucho esta experiencia y debo reconocer lo efectivo de esta metodología.

## 6) Glosario

- Actor            Personaje o entidad que participa en un caso de uso
- BD                Base de datos
- Caso de uso    Un caso de uso es una descripción de los pasos o las actividades que deberán realizarse para llevar a cabo algún proceso. Los personajes o entidades que participarán en un caso de uso se denominan actores.

<b>Diseño e implementación de una base de datos relacional para la gestión de un videojuego</b>		
<b>Título:</b>		PFC
<b>Referencia:</b>	INF-2013-00001-04	
<b>Modificación:</b>	22/01/2014 17:07	Página 53 de 56

Data Warehouse	Almacén de Datos. es una colección de datos orientada a un determinado ámbito (empresa, organización, etc.), integrado, no volátil y variable en el tiempo, que ayuda a la toma de decisiones
Diagrama de Gantt	Herramienta gráfica cuyo objetivo es mostrar el tiempo de dedicación previsto para diferentes tareas o actividades a lo largo de un tiempo total determinado.
Dimensión	Representa un punto de vista utilizado en los análisis de datos. Son las perspectivas de análisis de las variables. La información general complementaria a cada uno de los registros de la tabla de hechos.
DW	Data Warehouse.
EDT	Estructura de Descomposición del Trabajo o EDT, también conocida por su nombre en inglés Work Breakdown Structure o WBS, es en gestión de proyectos una descomposición jerárquica orientada al entregable, del trabajo a ser ejecutado por el equipo de proyecto, para cumplir con los objetivos de éste y crear los entregables requeridos
ETL	Extract, Transform and Load (Extraer, transformar y cargar en castellano, frecuentemente abreviado a ETL) es el proceso que permite a las organizaciones mover datos desde múltiples fuentes, reformatearlos y limpiarlos, y cargarlos en otra base de datos, data mart, o data warehouse para analizar, o en otro sistema operacional para apoyar un proceso de negocio.
Hecho	Un hecho representa un objeto de análisis
Métrica	Medida o conjunto de medidas destinadas a conocer o estimar el tamaño u otra característica
PDI	Pentaho Data Integration. Software para diseñar procesos ETL
PFC	Proyecto final de carrera.
PMBOOK	Desarrollada por el Project Management Institute (PMI), la Guía del PMBOK es el conjunto de conocimientos en Dirección/Gestión/Administración de Proyectos generalmente reconocidos como «buenas prácticas», y que se constituye como estándar de Administración de proyectos.
Recordset	Es una estructura de datos usada en programación cuya utilidad es la de almacenar información desde una tabla de una base de datos. Se usa con frecuencia para obtener conexiones con bases de datos y almacenar el resultado de las posibles consultas que se realicen.
SGBD	Siglas correspondientes a la asignatura de Sistemas de gestión de Bases de datos

<b>Diseño e implementación de una base de datos relacional para la gestión de un videojuego</b>		
<b>Título:</b>		PFC
<b>Referencia:</b>	INF-2013-00001-04	
<b>Modificación:</b>	22/01/2014 17:07	Página 54 de 56

- SGBD** Un sistema de gestión de bases de datos (SGBD) es un conjunto de programas que permiten el almacenamiento, modificación y extracción de la información en una base de datos, además de proporcionar herramientas para añadir, borrar, modificar y analizar los datos. Los usuarios pueden acceder a la información usando herramientas específicas de interrogación y de generación de informes, o bien mediante aplicaciones al efecto
- Trigger** Un trigger (también denominado disparador o desencadenante) en una Base de datos, es un procedimiento que se ejecuta cuando se cumple una condición establecida al realizar una operación. Dependiendo de la base de datos, los triggers pueden ser de inserción (INSERT), actualización (UPDATE) o borrado (DELETE).
- UML** Lenguaje Unificado de Modelado (LUM o UML, por sus siglas en inglés, Unified Modeling Language) es el lenguaje de modelado de sistemas de software más conocido y utilizado en la actualidad; Es un lenguaje gráfico para visualizar, especificar, construir y documentar un sistema.
- WBS** Work Breakdown Structure, EDT.

## 7) Bibliografía

- (1) Wikipedia
- (2) Materiales de la asignatura.
- (3) Módulos de la asignatura “Metodologia i gestió de projectes informàtics “
- (4) Módulos de la asignatura “Sistemes de gestió de base de dades”
- (5) Módulos de la asignatura “Models multidimensionals i magatzems de dades”
- (6) Otros PFC obtenidos de la fuente de información del aula.

## 8) Anexo. Procedimiento de instalación:

Es necesario que el equipo destino disponga del siguiente software:

- Oracle 11 g.
  - SQL Developer o similar para conexión con la base de datos y ejecución de los scripts.
  - PDI para el proceso de los ficheros JOB y ETL
  - PSW para crear el cubo.
  - Pentaho para realizar análisis.
- Dar de alta los usuarios pfc y MD en Oracle.
  - Abrir el script del fichero alopezjuan\_PAC2.sql en sql Developer y ejecutarlo para crear los componentes (tablas, desencadenantes, funciones y procedimientos) en el esquema pfc.

<b>Diseño e implementación de una base de datos relacional para la gestión de un videojuego</b>		
<b>Título:</b>		PFC
<b>Referencia:</b>	INF-2013-00001-04	
<b>Modificación:</b>	22/01/2014 17:07	Página 55 de 56

- Abrir el script del fichero alopezjuan\_PAC2\_Pruebas.sql en sql Developer y ejecutarlo para generar los datos de las pruebas.
- Abrir el script del fichero alopezjuan\_PAC3.sql en sql Developer y ejecutarlo para crear las tablas y desencadenantes del almacén de datos en el esquema MD.
- Abrir el fichero JOB\_CARGA\_DW.kjb en PDI y ejecutarlo para cargar con datos los hechos y dimensiones del MD.
- Abrir con PSW el fichero PSW\_CUBOS.xml y publicarlo para poder realizar análisis con pentaho.

<b>Diseño e implementación de una base de datos relacional para la gestión de un videojuego</b>		
<b>Título:</b>		PFC
<b>Referencia:</b>	INF-2013-00001-04	
<b>Modificación:</b>	22/01/2014 17:07	Página 56 de 56