

***UOC – LA UNIVERSIDAD VIRTUAL***

Ingeniería técnica en informática de sistemas.

***Diseño e implementación de  
un sistema de gestión e inteligente  
para una biblioteca***

Alumno: Julio Losa Andújar

Dirigida por: Gladys Utrera Iglesias

Curso 2005-2 (Septiembre – Enero)

**UNIVERSITAT OBERTA DE CATALUNYA**

Ingeniería.

**A mi mujer, por su paciencia durante estos  
años y por su incondicional apoyo.  
Y toda mi familia por animarme a llegar  
hasta el final.**

## RESUMEN

El objetivo del presente trabajo final de carrera consiste en el desarrollo de una base de datos que nos ha de permitir implementar un sistema de gestión e inteligente para una biblioteca. En mi caso he preferido enfocarlo hacia una biblioteca universitaria, pero puede ser aplicable a cualquier tipo de biblioteca, como podría haber sido, por ejemplo, una biblioteca que conozco muy bien: la Biblioteca Municipal de Mislata, de cuyo Ayuntamiento soy funcionario. Pero no va a ser posible ya que se encuentra conectada a la red pública de bibliotecas de la Comunidad Valenciana, dependiendo directamente de la Consellería de Cultura de la Generalitat.

Todas las herramientas utilizadas en el diseño del proyecto son las aprendidas en las asignaturas Bases de datos I y Bases de datos II de esta ingeniería, así como el uso del material bibliográfico que tengo a mi disposición, en este caso en concreto he necesitado consultar con frecuencia los títulos “El modelo entidad relación” de Richard Baker [BAR01], “Diseño conceptual de base de datos. Un enfoque de entidades relaciones” de Batini [BAT01] y “Database modelling and design” de Teoray [TEO01].

La parte de implementación tal vez haya sido la más costosa al ser esta mi primera toma de contacto con Oracle, pero puedo asegurar que me ha gustado y creo haber conseguido una buena base para profundizar más en este SGBD. He de decir que casi he “destrozado” la bibliografía que sobre Oracle poseo de tanto consultarla (referencias [ORA01] a [ORA07] de ka bibliografía).

El resultado final de todo este proceso ha sido la obtención de una base de datos que funciona conforme a los requerimientos pedidos en el enunciado del TFC, sus tablas, disparadores, funciones y procedimientos han sido desarrollados siguiendo los criterios dados y su funcionamiento se puede comprobar. Pienso que de haber tenido más conocimientos sobre Oracle el producto final obtenido hubiera estado mucho más refinado.

Como conclusión a este resumen inicial he de decir que desde mi punto de vista falta mucho por hacer y dar por terminada la aplicación. Creo que el proyecto puede verse enriquecido con la introducción de nuevos procesos que automaticen más la gestión de la biblioteca. También pienso que ya que Oracle nos da la posibilidad de acceder mediante web, cabría la posibilidad de dotar al sistema de esta posibilidad de forma que los usuarios pudieran realizar reservas de recursos disponibles desde su domicilio sin tener que desplazarse a ninguna de las sedes de la biblioteca.

**INDICE**

<b>1. Introducción.....</b>	<b>7</b>
<b>1.1. Objetivos.....</b>	<b>7</b>
<b>1.2. Contexto del TFC .....</b>	<b>7</b>
<b>1.3 Metodología seguida .....</b>	<b>8</b>
1.3.1. El modelo relacional .....	8
1.3.2. Fases en el diseño de una base de datos.....	8
1.3.3. El modelo ER.....	9
<b>2. Modelo propuesto y su desarrollo .....</b>	<b>10</b>
<b>2.1. El diseño .....</b>	<b>10</b>
<b>2.2. Planificación el proyecto.....</b>	<b>10</b>
2.2.1. Fases del proyecto .....	10
2.2.2. El proyecto en el tiempo (Temporalización).....	11
2.3.3. Diagrama de Gantt .....	12
<b>3. Elección de un SGBD.....</b>	<b>13</b>
<b>4. Diseño conceptual de la base de datos: El modelo ER .....</b>	<b>14</b>
<b>4.1. Definición del modelo ER.....</b>	<b>14</b>
4.1.1. Identificando las entidades. Primera aproximación.....	14
<b>4.2. Diagrama Entidad/Relación .....</b>	<b>15</b>
<b>4.3. Descripción Entidades .....</b>	<b>16</b>
<b>4.4. Comentarios sobre el modelo propuesto.....</b>	<b>18</b>
<b>4.5. Comentarios adicionales .....</b>	<b>19</b>
<b>5. Diseño lógico: Del modelo ER al modelo relacional .....</b>	<b>20</b>
<b>5.1. Normalización de tablas derivadas de un modelo ER .....</b>	<b>20</b>
<b>5.2. Introducción a la transformación de entidades e interrelaciones .....</b>	<b>20</b>
<b>5.3. Transformación al modelo relacional .....</b>	<b>21</b>
<b>5.4. Definición del modelo de datos.....</b>	<b>23</b>
<b>5.5 Esquema de las tablas de la B.D.....</b>	<b>31</b>
<b>6. Diseño de la base de datos Física.....</b>	<b>32</b>
<b>6.1. Definición del tamaño de las tablas .....</b>	<b>32</b>
6.1.1. Algunas consideraciones previas.....	32
6.1.2. Empresas.....	33
6.1.3. Personas.....	33
6.1.4. Empleados.....	34
6.1.5. Usuarios.....	34
6.1.6. Sectores.....	34
6.1.7 Temas .....	34
6.1.8. Subtemas.....	35
6.1.9. Recursos.....	35
6.1.10. Documentos.....	35
6.1.11. Mapas .....	35
6.1.12. Libros .....	36
6.1.13. Videos .....	36
6.1.14. Música .....	36
6.1.15. Copias.....	36
6.1.16. Retiradas .....	37
6.1.17. Prestables.....	37
6.1.18. Nuevos_Recursos .....	37
6.1.19. Nuevos_Documentos .....	37
6.1.20. Nuevos_Mapas.....	38

6.1.21. Nuevos_Libros.....	38
6.1.22. Nuevos_Videos.....	38
6.1.23. Nuevos_Música.....	38
6.1.24. Referencia.....	38
6.1.25. Lista_Espera.....	39
6.1.26. Préstamos.....	39
6.1.27. Solicitados.....	39
<b>6.2. Creación del Script de generación de la base de datos.....</b>	<b>39</b>
<b>6.3. Operaciones de gestión básicas.....</b>	<b>42</b>
<b>6.4. Packages (Procedimientos, funciones y consultas).....</b>	<b>43</b>
6.4.1 Package Sectores.....	44
6.4.1.1. PROCEDURE INS_SECTOR.....	44
6.4.2. Package Temas.....	44
6.4.2.1. FUNCTION COMPROBAR_TEMAS.....	44
6.4.2.2. PROCEDURE INS_TEMA.....	44
6.4.3. Package Subtemas.....	44
6.4.3.1. FUNCTION COMPROBAR_SUBTEMAS.....	44
6.4.3.2. FUNCTION COMPROBAR_NUMERO.....	44
6.4.3.3. PROCEDURE INS_SUBTEMA.....	44
6.4.4. Package Personas.....	44
6.4.4.1. PROCEDURE DEL_EMPLEADO.....	44
6.4.4.2. PROCEDURE I_EMPLEADO.....	45
6.4.4.3. PROCEDURE I_USUARIO.....	45
6.4.4.4. PROCEDURE MOD_UBICA_EM.....	45
6.4.4.5. PROCEDURE MOD_ESTADO_U.....	45
6.4.4.6. PROCEDURE MOD_D_PENA_U.....	45
6.4.5. Package Recursos.....	45
6.4.5.1. FUNCTION COMPROBAR_RECURSO.....	45
6.4.5.2. FUNCTION COMPROBAR_COPIAS.....	45
6.4.5.3. PROCEDURE INS_DOCUMENTO.....	45
6.4.5.4. PROCEDURE INS_MAPAS.....	45
6.4.5.5. PROCEDURE INS_LIBRO.....	45
6.4.5.6. PROCEDURE INS_VIDEO.....	45
6.4.5.7. PROCEDURE INS_MUSICA.....	45
6.4.5.8. PROCEDURE INS_COPIA.....	46
6.4.6. Package Nuevos Recursos.....	46
6.4.6.1. FUNCTION COMPROBAR_RECURSO.....	46
6.4.6.2. PROCEDURE INS_DOCUMENTO.....	46
6.4.6.3. PROCEDURE INS_MAPAS.....	46
6.4.6.4. PROCEDURE INS_LIBRO.....	46
6.4.6.5. PROCEDURE INS_VIDEO.....	46
6.4.6.6. PROCEDURE INS_MUSICA.....	46
6.4.7. Package Gestión.....	46
6.4.7.1. FUNCTION COMPROBAR_LISTA.....	46
6.4.7.2. FUNCTION COMPROBAR_TIEMPO_MEDIO.....	46
6.4.7.3. FUNCTION OBTENER_FP.....	46
6.4.7.4. FUNCTION OBTENER_FUD.....	46
6.4.7.5. FUNCTION COMPROBAR_PRESTAMOS.....	46
6.4.7.6. FUNCTION COMPROBAR_RETIRADA.....	46
6.4.7.7. FUNCTION OBTENER_PR.....	46
6.4.7.8. FUNCTION OBTENER_DP.....	47
6.4.7.9. FUNCTION OBTENER_ESTADO.....	47
6.4.7.10. FUNCTION OBTENER_PENALIZACION.....	47
6.4.7.11. PROCEDURE INS_LISTA.....	47
6.4.7.12. PROCEDURE TIEMPO_MEDIO.....	47
6.4.7.13. PROCEDURE INS_PRESTAMO.....	47
6.4.7.14. PROCEDURE DEV_PRESTAMO.....	47
6.4.7.15. PROCEDURE HIS_PRESTAMO.....	47
6.4.7.16. PROCEDURE UPD_COPIAS.....	47
6.4.8. Package Consultas.....	47

6.4.8.1. PROCEDURE CONS_1.....	47
6.4.8.2. PROCEDURE CONS_2.....	47
6.4.8.3. PROCEDURE CONS_3.....	47
6.4.8.4. PROCEDURE CONS_4.....	47
6.4.8.5. PROCEDURE CONS_5.....	47
6.4.8.6. PROCEDURE CONS_6.....	48
6.4.8.7. PROCEDURE CONS_7.....	48
6.4.9. Package Estadística.....	48
6.4.9.1. PROCEDURE EST_1.....	48
6.4.9.2. PROCEDURE EST_2.....	48
6.4.9.3. PROCEDURE EST_3.....	48
6.4.9.4. PROCEDURE EST_4.....	48
6.4.9.5. PROCEDURE EST_5.....	48
6.4.9.6. PROCEDURE EST_6.....	48
<b>7. Valoración de costes del proyecto.....</b>	<b>49</b>
7.1. Valoración del Hardware.....	49
7.2. Valoración del software.....	53
<b>8. Conclusiones.....</b>	<b>54</b>
<b>9. Bibliografía.....</b>	<b>56</b>
<b>Anexo I. Enunciado del TFC.....</b>	<b>57</b>

## 1. Introducción

### 1.1. Objetivos

El objetivo de este TFC es el diseño e implementación de un sistema de gestión e inteligente para una biblioteca. A parte de lo que conlleva el diseño de la base de datos que contendrá todos los recursos con los que cuenta la biblioteca, así como de los datos de sus usuarios y administradores y, finalmente, de todos los movimientos de préstamos de los recursos disponibles, también deberemos realizar el desarrollo de una serie de herramientas inteligentes que sean capaces de llevar a cabo la gestión de los recursos de la misma, como por ejemplo:

- Saber en un momento dado en que estado se encuentra un recurso, esto es si se encuentra prestado, a quien, cuando lo debe devolver y si se produce retraso en su devolución. En principio sería el funcionamiento normal de la biblioteca.
- A tenor de lo anterior el sistema debe ser capaz de aplicar sanciones de forma automática a los infractores, esto es, a los usuarios que se retrasen en la devolución de los recursos solicitados, por lo que se deberá desarrollar la herramienta capaz de llevar esta gestión.
- El sistema también debe controlar la solicitud de recursos que no se encuentran entre el material disponible, en la definición del proyecto se habla solo de libros, pero creo interesante que se extienda al resto de los recursos disponibles. De esta forma mediante el desarrollo de la herramienta adecuada el sistema deberá poder realizar la petición del mismo de forma automática, deberá comprobar que el recurso no estaba ya en la lista de pedidos. Para poder realizar esta gestión deberán cumplirse una serie de requisitos que la herramienta desarrollada deberá ser capaz de comprobar, como, por ejemplo, que el recurso haya sido solicitado por un número determinado de usuarios, y en función de ese número deberá calcular el número de copias a solicitar.
- El sistema debe ser capaz de usar la información estadística que obtenga para poder realizar estas y otras acciones, entre las nuevas posibilidades, además de la adquisición, deberemos dotar a nuestro sistema de las herramientas necesarias para que pueda decidir cuando realizar la renovación de un recurso, cuando reubicarlo o cuando eliminarlo del catálogo.

Por otra parte debemos ver nuestra base de datos como una base de conocimientos. Si partimos de que una base de conocimientos es un depósito de información creado gracias a una extensa investigación organizada en un árbol de conocimientos completo. Nuestra base de conocimientos se crea con el propósito de cubrir todos los aspectos de la biblioteca y no únicamente las características y las funcionalidades. Al almacenar los datos de los recursos en un árbol de conocimientos podemos organizar con eficiencia todas las necesidades de la biblioteca y los usuarios pueden enfocarse en sus prioridades en el nivel de los detalles de su selección. Además de tener una visualización organizada de todos los aspectos de su evaluación, una base de conocimientos permite almacenar notas, comentarios y otros datos importantes para cada nivel del árbol de conocimientos.

### 1.2. Contexto del TFC

¿Cómo podría incluirse el presente trabajo en el mundo real? Hoy en día la mayor parte de las bibliotecas de centros oficiales se encuentran informatizadas, en mayor o menor medida, y tal vez este pequeño grano de arena podría seguir desarrollándose hasta llegar a ser un castillo que abarcará en su totalidad la gestión de una gran biblioteca con cientos de miles de volúmenes, donde la intervención externa se limitará a la pura gestión diaria (préstamos, adquisición de nuevos recursos,...) dejando que sea la propia base de datos la que de forma automática realice la mayor parte de los procesos.

Quién sabe, igual ya hay algo así funcionando en alguna parte ¿o no?

## 1.3 Metodología seguida

### 1.3.1. El modelo relacional

El modelo relacional se basa en el modelo matemático de relación, en este modelo las estructuras de datos constan de un único elemento: la tabla, formada por registros y estos a su vez divididos en columnas.

El diseño de una base de datos relacional puede seguir dos caminos.

1. Puede crearse tomando como punto de partida la observación del universo en estudio, dando lugar a un conjunto de esquemas de relaciones, que contengan los atributos y sus restricciones.
2. Puede dividirse el diseño en dos fases, la primera de las cuales sería definir el modelo conceptual y su esquema, y la segunda transformar el esquema conceptual en un esquema relacional mediante una transformación realizada de acuerdo a unas reglas dadas.

### 1.3.2. Fases en el diseño de una base de datos

Sin perjuicio del rigor en el diseño relacional, el diseño de una base de datos no puede limitarse a la aplicación exclusiva de la teoría de la normalización. Del mismo modo que se ha visto la existencia de variadas metodologías en el ámbito de los sistemas de información, se encuentra el mismo panorama en el diseño de bases de datos, aunque aquí tampoco aparece una metodología consagrada. De esta forma, Elmasri y Navathe [ELM01] comparan el ciclo de diseño de los sistemas de información y de las bases de datos, y definen el problema de diseñar una base de datos como:

"El diseño de la estructura lógica y física de una o más bases de datos para acomodar las necesidades de información de los usuarios a un determinado conjunto de aplicaciones "

y señalan la existencia de seis fases en el proceso de diseño de una base de datos:

- *Fase 1: Recopilación y análisis de requerimientos.* En esta fase se trata de conocer las expectativas del usuario sobre la base de datos. Para ello, se identifican los grupos de usuarios reales y posibles y las áreas de aplicación, se revisa la documentación existente, se analiza el entorno operativo y los requerimientos de procesado, y se realizan entrevistas y cuestionarios con los usuarios.
- *Fase 2: Diseño conceptual de la base de datos.* Esta fase se subdivide en otras dos.
  - La Fase 2a corresponde al Diseño del esquema conceptual, esquema de especificación del modelo de datos a alto nivel, independiente de cualquier SGBD, que no puede utilizarse para implementar directamente la estructura de la base de datos. Para obtenerlo puede adoptarse un enfoque de esquema centralizado (en el cual se unen previamente los diferentes requerimientos a la realización del esquema), o un enfoque de integración de vistas (en el cual se unen los esquemas de cada requerimiento en uno global realizado a posteriori).
  - La Fase 2b corresponde al diseño de transacciones, es decir, a aquellas aplicaciones que van a manipular datos contenidos en la base de datos. Se suelen identificar mediante el estudio de las entradas y salidas de datos y su comportamiento

funcional. De esta forma se identifican transacciones de recuperación, de actualización y mixtas.

- *Fase 3: Elección de un SGBD.* Se consideran diferentes factores técnicos, económicos y de beneficio, de servicio técnico y formación de usuarios, organizativos de rendimiento, etc. Sin embargo, resulta difícil la medida y cuantificación ponderada de los diferentes factores.
- *Fase 4: Transformación del modelo de datos (o fase de diseño lógico).* En esta fase se crea un esquema conceptual y los esquemas externos necesarios en el modelo de datos del SGBD seleccionado, mediante la transformación de los esquemas de modelo de datos a alto nivel obtenidos en la Fase 2a, al modelo de datos ofrecido por el SGBD.
- *Fase 5: Diseño de la base de datos física.* Consiste en definir las estructuras de almacenamiento y de acceso para alcanzar una rendimiento óptimo de las aplicaciones de la base de datos. Los criterios adoptados suelen ser el tiempo de respuesta, la utilización de espacio y el volumen de transacciones por minuto.
- *Fase 6: Implementación del sistema de base de datos.* En esta fase final se hace realidad la base de datos, mediante la creación y la compilación del esquema de bases de datos y de los ficheros de bases de datos, así como de las transacciones, a través de las aplicaciones.

La metodología expuesta, que puede servir como marco de referencia general, puede modificarse según las características del contexto en el que se diseña e implanta el sistema de bases de datos.

### 1.3.3. El modelo ER

En el dinámico entorno de la información almacenada en las bases de datos, las recientes tendencias, derivadas en muchas ocasiones de las propias necesidades, han obligado a completar e incorporar nuevos conceptos y enfoques en el tratamiento de los datos. Por ejemplo, la existencia de relaciones complejas en el mundo real han obligado a la incorporación del modelado semántico, lo que ha dado como resultado la evolución del modelo entidad-relación extendido, con sus conceptos de superclases y subclases, y los procesos de generalización y especialización, así como la importante noción de herencia. También es necesaria la referencia ineludible al paradigma de la orientación a objetos (*IBERO1*), enfoque de tratamiento de la información que cobra cada vez mayor auge en aplicaciones comerciales, y que se configura como la opción de mayor futuro en el desarrollo de aplicaciones. La identificación de los datos y sus procesos como objetos individuales, el encapsulamiento y las propiedades de herencia son las características principales del enfoque a objetos. Por último, no puede olvidarse la creciente tendencia entre el enfoque relacional y el modelo de objetos, así como la integración de información referenciada espacialmente en modelos relacionales.

Es innegable que la gestión y la explotación subsiguiente de los registros que contienen datos, y, como consecuencia, información, depende de las herramientas existentes en el campo de la gestión de la información, por una parte, y del cuerpo teórico de la ciencia de la información, por otra. La explotación satisfactoria de esta información, de la misma forma, demanda experiencia en dos áreas de conocimiento: en las técnicas de recuperación de información y en el estudio de las necesidades de los usuarios.

## 2. Modelo propuesto y su desarrollo

### 2.1. El diseño

A la hora de realizar el diseño de la base de datos de los dos caminos propuestos en el punto 1.2.1 seguiré el segundo, esto es, realizar la definición del modelo conceptual y realizar su esquema siguiendo el modelo de entidad-relación, tras lo cual proceder a transformar el esquema ER obtenido en un esquema relacional aplicando las reglas dadas en la descripción del trabajo a realizar.

### 2.2. Planificación el proyecto

#### 2.2.1. Fases del proyecto

El siguiente paso será realizar la temporalización del trabajo a realizar, podemos seguir el modelo de 6 fases propuesto por Elmasri y Navathe [ELM01] y aplicarlo a nuestro caso.

Aplicando las diferentes fases a nuestro caso obtendremos el siguiente resultado:

- *Fase 1: Recopilación y análisis de requerimientos.* Esta parte corresponde al enunciado entregado (puede verse en el Anexo I) que es el que nos va servir de base para la realización de las siguientes fases.
- *Fase 2: Diseño conceptual de la base de datos.* Esta fase corresponde al Diseño del modelo ER, independiente de cualquier SGBD, el cual no se puede utilizar para implementar directamente la estructura de la base de datos. Para obtenerlo adoptaremos un enfoque de esquema centralizado. La duración estimada de esta fase es de 20 días.
- *Fase 3: Elección de un SGBD.* En este caso esta fase ya está realizada ya que vamos a utilizar Oracle 9i como SGBD.
- *Fase 4: Transformación del modelo de datos (o fase de diseño lógico).* En esta fase crearemos el esquema conceptual a partir del modelo ER mediante la transformación del esquema obtenido en la fase anterior al modelo de datos ofrecido por el SGBD, en este caso Oracle 9i. La duración aproximada de esta fase será de 15 días.
- *Fase 5: Diseño de la base de datos física.* Una vez obtenido el modelo lógico, estudiaremos las necesidades de tamaño de las diferentes tablas a crear en función de la previsión de los movimientos que puedan llevarse a cabo, para finalizar con el desarrollo de los Script necesarios para la generación de las tablas que compondrán nuestra base de datos. La duración aproximada de esta fase será de 5 días.
- *Fase 6: Implementación del sistema de base de datos.* En esta última fase se ejecutan los Script de creación de la base de datos y se desarrollaran las funciones y procedimientos necesarios para el correcto funcionamiento de la base de datos. La duración aproximada de esta fase será de 30 días.
- *Fase pruebas:* Esta fase que no corresponde con el modelo servirá para la carga de datos de prueba y las posteriores pruebas de las funciones y procedimientos desarrollados. En esta fase podremos depurar posibles errores de programación cometidos durante las fases anteriores, el tiempo estimado sería 15 días.

### 2.2.2. El proyecto en el tiempo (Temporalización)

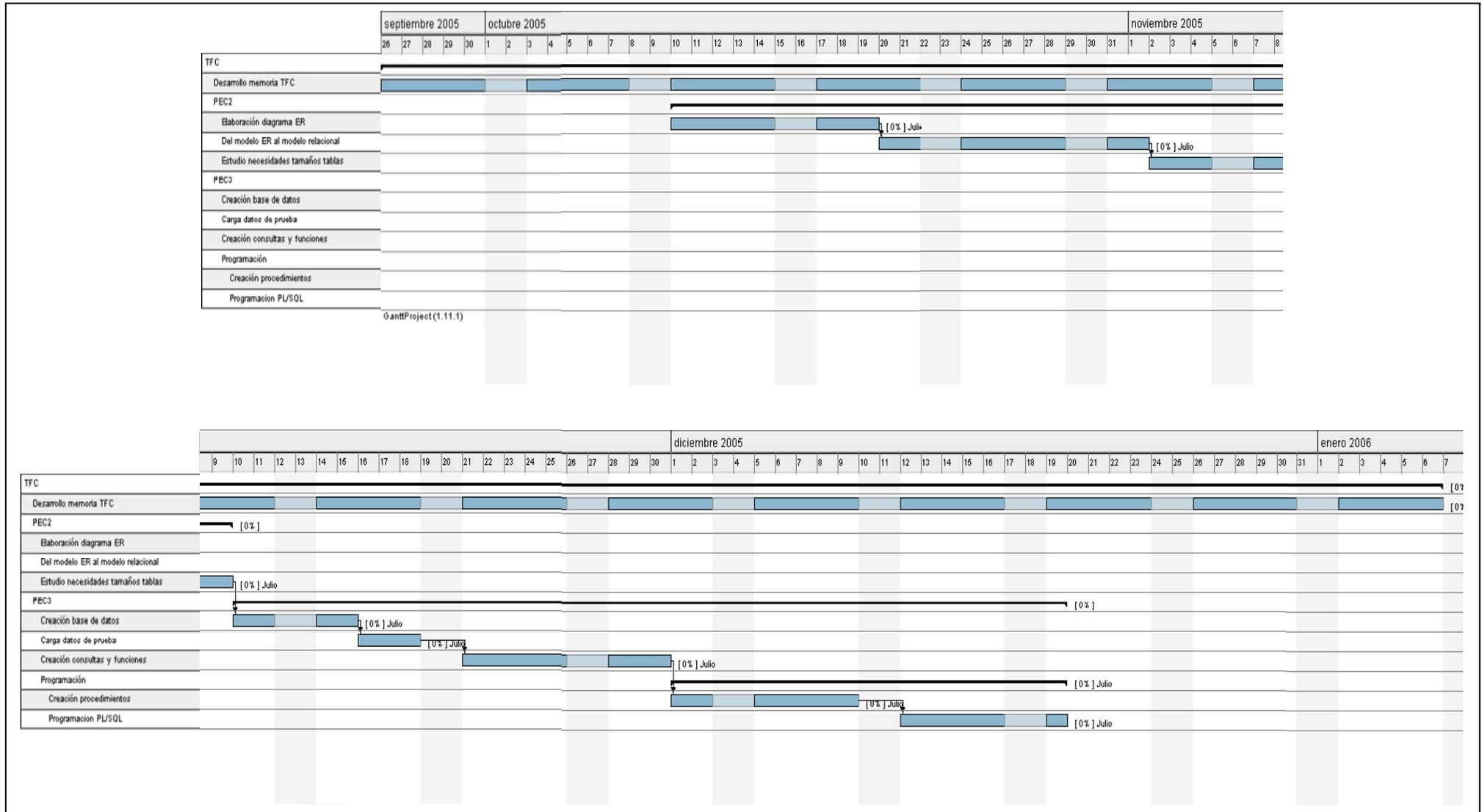
La fase 2 de diseño conceptual de la base de datos comienza el día 10 de octubre con la realización del diagrama ER que deberá estar finalizada el 19 del mismo, la fase 4, la transformación del modelo ER en un modelo relacional, se iniciaría el día 20 de octubre finalizando el 9 de noviembre, hasta aquí llegará el contenido de la PEC2. A continuación comienza la fase 5, inicio de la PEC3, que debe llevarse a cabo entre el 10 y el 18 de noviembre (implica esta fase tanto la creación como la carga de datos de prueba). La fase 6 comenzará el 19 de noviembre y finaliza el 20 de diciembre, durante la cual se montaran las consultas, funciones y procedimientos necesarios para la obtención de los objetivos propuestos. Finalmente, a partir del día 21 de diciembre y hasta el 3 de enero será un tiempo de prueba y depuración del trabajo realizado. Quedan del 4 al 9 de enero para retocar la memoria y la presentación.

Con el propósito de que no recaiga la carga de la realización de la memoria en los últimos 5 días, a la vez que se van realizando las diferentes fases se irá creando la misma, así como la preparación de la presentación virtual a desarrollar en PowerPoint.

Toda la planificación se puede ver en el diagrama de Gantt de la siguiente página, como los encabezamientos se pueden ver mejor en la siguiente figura:

TFC
Desarrollo memoria TFC
PEC2
Elaboración diagrama ER
Del modelo ER al modelo relacional
Estudio necesidades tamaños tablas
PEC3
Creación base de datos
Carga datos de prueba
Creación consultas y funciones
Programación
Creación procedimientos
Programacion PL/SQL

### 2.3.3. Diagrama de Gantt



### 3. Elección de un SGBD

Aunque en este caso no sea necesaria la elección de un SGBD, ya que el elegido para la realización del proyecto sea Oracle 9i, me parece interesante el desarrollo de este punto como referencia o ayuda a la hora de la elección de un SGBD para la implementación de un sistema de bases de datos.

En algunas ocasiones puede representar un problema la elección del un determinado SGBD para realizar la implementación de nuestro sistema de base de datos.

La elección que realicemos es muy importante ya que condicionará el desarrollo y la explotación de aplicaciones futuras y, por otra parte, es una elección difícil, ya que en ella intervienen muchas circunstancias tanto de carácter técnico como no técnico que son de difícil evaluación.

No existe in SGBD que sea el mejor, cada uno de ellos tiene una serie de características que le hacen ser el mejor dentro de su entorno, esto es, para una determinada máquina, para un determinado sistema operativo, para un determinado volumen de datos o para unas aplicaciones de unas características determinadas.

Jaume Sistac ([SIS01] páginas 177-180), describe una metodología en seis pasos para ayudarnos en la elección del SGBD que mejor se adapta a nuestras necesidades. Los pasos de los que consta esta metodología son los siguientes:

- 1) Determinar las características de las aplicaciones con relación a los accesos a las bases de datos, esto es, cuáles son los requerimientos de aplicación, de usuario, de seguridad, de integridad, de concurrencia, etc., respecto a la base de datos.
- 2) Determinar los criterios que utilizaremos para evaluar los SGBD, y las condiciones mínimas que han de cumplir. Los criterios de evaluación son consecuencia de las características de las aplicaciones que hemos determinado, estos criterios los podemos agrupar en tres grandes grupos.
  - a. Relativos al desarrollo y mantenimiento de aplicaciones.
  - b. Relativos al funcionamiento de la base de datos.
  - c. Relativos a la comercialización del producto.
- 3) Localizar los SGBD disponibles en el mercado y seleccionar los que cumplan los criterios anteriores.
- 4) Analizar y evaluar cada SGBD seleccionado en relación con cada uno de los criterios anteriores. La evaluación que realicemos podrá ser cuantitativa o cualitativa.
  - a. Evaluación cuantitativa. Consiste en asignar un peso numérico a cada uno de los criterios fijados para evaluar los SGBD. El peso numérico se asigna en función de la importancia relativa que damos a cada criterio. Puntuaremos cada SGBD según el grado de cumplimiento de cada criterio.
  - b. Evaluación cualitativa. Calificaremos el grado de cumplimiento de cada criterio de la forma más objetiva posible.
- 5) Resumen comparativo del análisis anterior. Si hemos optado por una evaluación cuantitativa bastará con sumar las puntuaciones obtenidas por cada SGBD examinado. Si, en cambio, hemos optado por la evaluación cualitativa deberemos comparar el conjunto de los criterios elegidos.
- 6) Recomendar la opción mejor evaluada, que haya obtenido la mejor puntuación o que haya sido la mejor evaluada cualitativamente. Finalmente, podemos proponer un plan para implementar el sistema.

## 4. Diseño conceptual de la base de datos: El modelo ER

Dada su simplicidad y legibilidad el modelo ER es uno de los enfoques de modelización de datos que más se utiliza en la actualidad, esto se debe a una notación diagramática muy comprensiva. Es una herramienta útil ya que tiene una doble función, por un lado ayuda al diseñador a plasmar en un modelo conceptual los requisitos del mundo real y por otro lado comprobar sobre el modelo conceptual si satisface las necesidades del usuario final.

Cuando utilizamos el modelo ER para comunicarnos con el usuario final es conveniente emplear en el modelado sus elementos más simples, como son las entidades, atributos e interrelaciones, y en algunos casos podemos utilizar entidades débiles y dependencias de existencia. En el momento de desarrollar la tarea de modelizar entonces usaremos un modelo ER más completo que incluyan construcciones más avanzadas.

Por último, y antes de pasar a desarrollar el diseño conceptual de nuestra base de datos, debemos tener en cuenta que el modelo ER se utiliza para reflejar aspectos de las estructuras de los datos y de su integridad pero no de su manipulación.

La base bibliográfica para la realización de este punto han sido *Disseny de bases de dades [SIS01]*, *Bases de datos I [SIS02]* y *Bases de datos II [SIS03]*.

### 4.1. Definición del modelo ER

El siguiente paso en la temporalización propuesta es la elaboración del diagrama entidad-relación que nos servirá como base para la posterior construcción de nuestra base de datos.

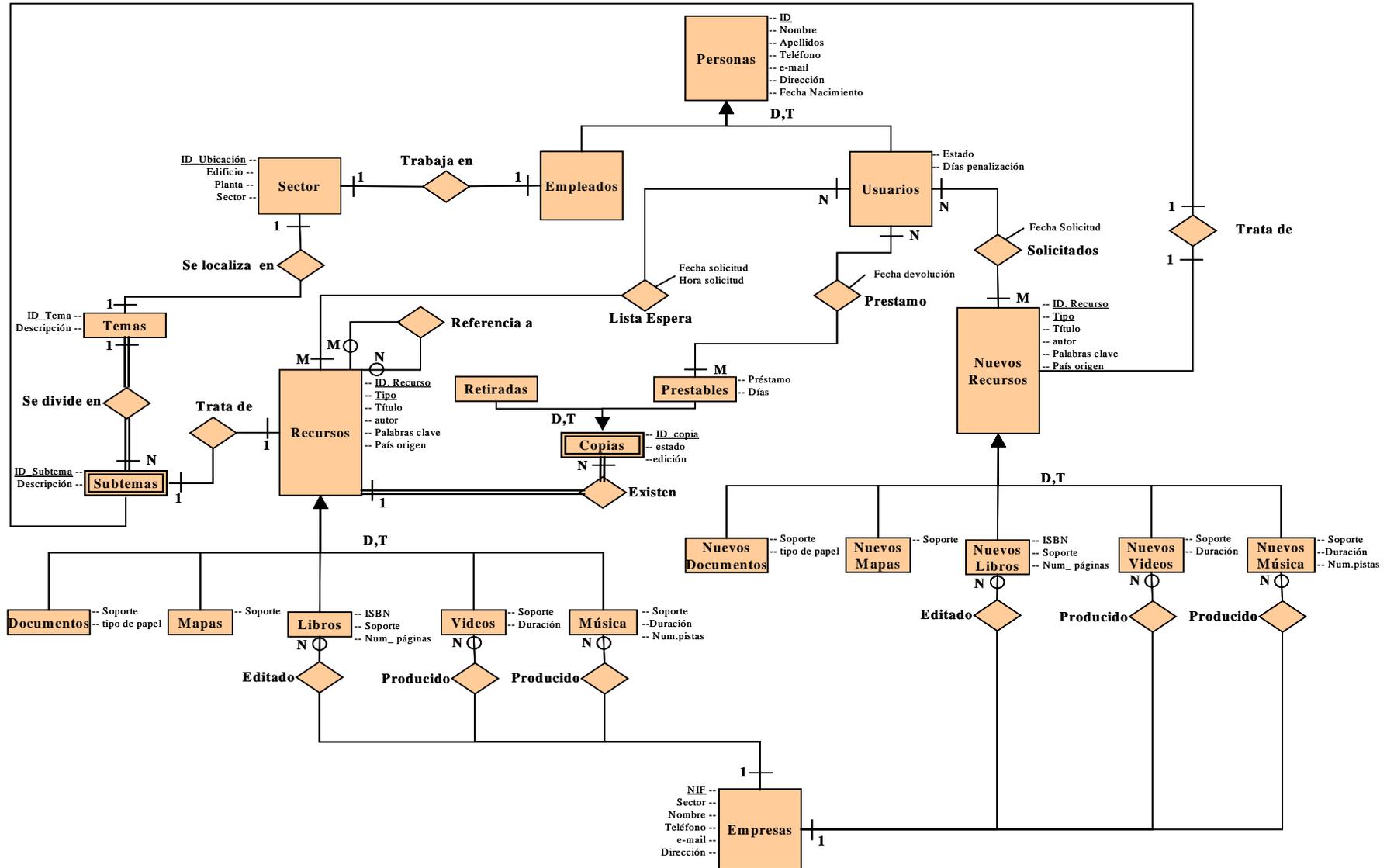
Para ello partimos del enunciado dado y que se puede ver en el anexo I de este documento, intentando identificar las diferentes entidades, atributos, interrelaciones, etc. que van a formar parte del diagrama propuesto.

#### 4.1.1. Identificando las entidades. Primera aproximación

Llega la hora de realizar el diagrama y tras una primera lectura ya podemos hacernos una idea de como puede ser. Así pues podemos que existirá una entidad a la que llamaremos Recursos, que apunta a ser una superclase, de la que dependerán Documentos, Mapas, Libros, Vídeos y Música, que serán subclases. Otras dos entidades que aparecen claramente definidas son los usuarios y los empleados.

El siguiente paso es hacer una lectura más profunda y obtener el diagrama Entidad-Relación tal y como lo podemos observar en el siguiente punto.

## 4.2. Diagrama Entidad/Relación



### 4.3. Descripción Entidades

En este punto podemos ver una breve descripción de las entidades utilizadas en el diagrama anterior así como de los atributos que las componen. El atributo que aparece subrayado es la clave primaria de esa entidad.

#### Personas

Superclase que contiene la información común a Empleados y a Usuarios.

Atributos:

ID, nombre, apellidos, teléfono, e-mail, dirección, fecha de nacimiento

#### Empleados

Subclase de Personas, hereda todos los atributos de Personas y añade uno nuevo que identifica al empleado.

Atributos:

ID

#### Usuarios

Subclase de Personas, hereda todos los atributos de Personas y añade la identificación del usuario y un atributo para recoger los días de penalización que pueda tener el usuario en caso de retrasos en las devoluciones.

Atributos:

ID, estado, días\_penalización

#### Sectores

Contiene todos los sectores que pueden formar parte de la biblioteca, los atributos identifican el edificio, la planta y el sector dentro de esta, la clave primaria identifica a estos tres atributos..

Atributos:

ID\_Sector, edificio, planta, sector

#### Tema

Contiene todos los temas sobre los que pueden versar los recursos.

Atributos:

ID\_Tema, Descripción

#### Subtema

Entidad débil de temas, contiene la subdivisión de los temas.

Atributos:

ID\_Subtema, Descripción

#### Recursos

Superclase que contiene la información común a todos los recursos.

Atributos:

ID\_Recurso, tipo, título, autor, palabras\_clave, país\_origen

#### Documentos

Subclase de Recursos para los documentos.

Atributos:

ID\_Recurso, soporte, tipo\_papel

#### Mapas

Subclase de Recursos para los mapas.

Atributos:

ID\_Recurso, soporte

### Libros

Subclase de Recursos para los libros.

Atributos:

ID\_Recurso, ISBN, soporte, número\_páginas

### Videos

Subclase de Recursos para los videos.

Atributos:

ID\_Recurso, soporte, duración

### Música

Subclase de Recursos para la música.

Atributos:

ID\_Recurso, soporte, duración, número pistas

### Copias

Entidad débil de Recursos

ID\_copia, edición, estado

### Retiradas

Entidad subclase de Copias que es entidad débil de Recursos, contiene las copias que no están disponibles para el uso público por que su estado no lo permite.

Atributos:

ID\_copia

### Prestables

Entidad subclase de [Copias](#) que es entidad débil de [Recursos](#), contiene las copias que están disponibles para préstamo, bien para consultar en sala o para préstamo domiciliario. La posibilidad del préstamo en sala o domiciliario depende del valor que tome el atributo préstamo, más adelante volveremos sobre este asunto y veremos como hacerlo.

Atributo:

ID\_copia, préstamo, días

### Empresas

Contiene las diferentes productoras o distribuidoras de libros, videos o música.

Atributos:

NIF, sector, nombre, teléfono, e-mail, dirección

### Nuevos Recursos

Superclase que contiene la información común a todos los recursos que no se encuentran en la entidad [Recursos](#) y que han sido solicitados por los usuarios.

Atributos:

ID\_Recurso, tipo, titulo, autor, palabras\_clave, país\_origen

### Nuevos Documentos

Subclase de [Nuevos Recursos](#) para los documentos.

Atributos:

ID\_Recurso, soporte, tipo\_papel

### Nuevos Mapas

Subclase de [Nuevos Recursos](#) para los mapas.

Atributos:

ID\_Recurso, soporte

### Nuevos Libros

Subclase de [Nuevos Recursos](#) para los libros.

Atributos:

ID\_Recurso, ISBN, soporte, número\_páginas

### Nuevos Videos

Subclase de [Nuevos Recursos](#) para los videos.

Atributos:

ID\_Recurso, soporte, duración

### Nuevos Música

Subclase de [Nuevos Recursos](#) para la música.

Atributos:

ID\_Recurso, soporte, duración, número pistas

## 4.4. Comentarios sobre el modelo propuesto

- La entidad *Copia* se ha considerado una entidad débil porque su atributo *ID\_copia* no nos permite identificar de forma total la copia, necesitamos de la interrelación “*Existe*” para permitir completar la identificación de la entidad copia.
- Las entidades *Retiradas* y *Prestables* son subclases de la entidad *Copias*, la primera recoge todas aquellas copias que han sido retiradas por ser necesario restaurarlas o por que por su mal estado deban ser repuestas y la segunda se correspondería con el catálogo de préstamos.
- La entidad *Personas* no aparece en el modelo original, donde si lo hacen *Usuarios* y *Empleados*. Esta entidad se ha creado recogiendo los atributos comunes de usuarios y empleados convirtiéndose en una superclase.
- La entidad *Empresas* recoge a las editoriales y productoras cinematográficas y musicales (diferenciadas por el atributo sector). Aparece inicialmente como un atributo de la entidad *Recursos* pero he preferido crear una entidad nueva que se relaciona con *Recursos* a través de tres interrelaciones como se puede ver en el esquema.
- La entidad *Plantas* es una entidad débil de *Edificios* ya que necesita de esta para estar plenamente identificada, lo mismo ocurre con *Sectores* que es entidad débil de *Plantas*.
- Mediante las interrelaciones “*Trabaja en*” y “*Se localiza en*” podremos localizar el lugar de trabajo de un empleado o donde se encuentra un tema que nos permitirá localizar un recurso. Podríamos haber relacionado la localización con subtemas para tener una localización más exacta, pero el enunciado habla de temas localizados en sectores. En mi opinión y en el caso de relacionar la ubicación con los subtemas debería introducirse un nuevo nivel en la localización, que podríamos llamar *Estantería*. De esta forma del recurso A podríamos decir que se encuentra en el edificio E, planta P, sector S, estantería 1, quedando bastante más localizado.
- La entidad *Subtemas* es débil de la entidad *Temas* ya que necesitan de esta para estar plenamente identificada.

- La relación “*Trata de*” nos une a un recurso dado con el tema y el subtema sobre los que versa.
- En el enunciado se indica que algunos recursos cuentan con referencia bibliográfica, esto queda recogido en la interrelación “*Referencia a*”, de tal forma que puede haber algún recurso que puede hacer referencia a otro u otros.
- La interrelación “*Préstamo*” relaciona a las entidades *Usuarios* y *Prestables*. Además de permitirnos relacionar a los usuarios con los recursos prestados, añadiendo la fecha de devolución a la relación, nos puede servir como histórico de préstamos, más adelante, en otra fase del trabajo, veremos como podemos desarrollar este aspecto de la relación.
- La relación “*Lista de espera*” añade a un usuario a una lista de espera en el caso de que el recurso no esté disponible por estar en préstamo, añade los atributos fecha y hora para poder saber que usuario ha solicitado antes un recurso y a la hora de que este esté disponible poder avisarlo.
- Las entidades *Nuevos Recursos*, *Nuevos Documentos*, *Nuevos Libros*, *Nuevos Videos* y *Nuevos Música* aparecen ante la necesidad de recoger aquellas solicitudes de recursos que no aparecen en la entidad *Recursos* y que pueden ser susceptibles de una próxima compra. Se relacionan con la entidad *Usuario* mediante la interrelación “*Solicitados*”. También estaría interrelacionado con la entidad *Empresas* tal y como ocurre con La entidad *Recursos*. La salvedad, en este caso, es que lo más probable el usuario no pueda completar la totalidad de los atributos, por desconocerlos, pero eso es un tema a tratar más adelante.

#### 4.5. Comentarios adicionales

Aun cuando deseemos que el modelo ER sea lo más extenso posible, siempre nos podemos encontrar con alguna o algunas situaciones que no podamos modelar mediante el modelo ER.

En nuestro caso hay varias situaciones de este tipo, dos de ellas son la solicitud automática de recursos y la baja automática de los socios. ¿Cómo modelar esto mediante ER? No podemos, estas dos situaciones dependen de una serie de condicionantes que irán apareciendo durante la explotación de nuestra base de datos ¿Cómo solventamos esta situación? Pues sencillamente deberemos recurrir al uso de “*triggers*” o disparadores. Por ejemplo, veamos como procederíamos en el caso de la baja automática, esta afecta a la tabla *Usuarios*, y lo que debemos hacer es un seguimiento de los cambios de valor del campo *días\_penalización* cuando el numero de días almacenados sea mayor a 180 días (6 meses). Crearíamos una tabla que almacene los registros de auditoria, luego crearíamos un disparador “*Before update*” a nivel de fila que solo se ejecutará si el valor de *días\_penalización* alcanza el valor de 180.

No solo serán estos dos casos, durante el desarrollo del proyecto pueden aparecer alguno más que requiera de la misma solución.

## 5. Diseño lógico: Del modelo ER al modelo relacional

En este apartado vamos a partir del diseño conceptual confeccionado en el apartado anterior mediante el modelo ER y transformarlo en una estructura de datos del modelo relacional.

Pero antes de empezar a realizar esta transformación, creo interesante tratar dos temas : la normalización de tablas derivadas de diagramas ER y sobre como se transforman las entidades y las interrelaciones a relaciones en el modelo relacional.

La base bibliográfica para la realización de este punto han sido las siguientes: *Bases de datos I [SIS02]* y *Bases de datos II [SIS03]*.

### 5.1. Normalización de tablas derivadas de un modelo ER

La normalización de las posibles tablas derivadas de un modelo ER se puede realizar mediante el análisis de las dependencias funcionales asociadas con estas tablas: dependencias funcionales explícitas derivadas de los requerimientos del análisis de la base de datos, dependencias funcionales derivadas del modelo ER obtenido, y dependencias funcionales derivadas de la intuición.

Las dependencias funcionales primarias representan las dependencias entre los atributos que son claves de las entidades, dicho de otra manera, las dependencias inter-entidad. De otra parte, las dependencias funcionales secundarias representan las dependencias entre los atributos que componen una entidad, o de otra forma las dependencias intra-entidad. Usualmente, las dependencias funcionales primarias derivan directamente del modelo ER obtenido mientras que las secundarias se obtienen del análisis de requerimientos.

Usualmente cada posible tabla tendrá varias dependencias funcionales primarias y secundarias únicamente asociadas a ella, lo cual determina el grado actual de normalización de la tabla. Podemos aplicar cualquiera de las técnicas conocidas, como por ejemplo la desarrollada por Bernstein ([BER01] páginas 272-298) para incrementar el grado de normalización de la tabla al grado requerido.

Cualquier tabla Y que se integre en otra tabla X puede ser posiblemente eliminada. Una Tabla Y se integra en otra tabla X cuando todos los atributos de Y están contenidos en X y todas las dependencias de Y también ocurren en X. El caso más simple, cualquier tabla que contenga solo una clave compuesta y un atributo no clave se integra automáticamente en cualquier otra tabla que contenga los mismos atributos, por que la clave compuesta es la forma más débil de dependencia. Si, sin embargo, las tablas X e Y representan la superclase y la subclase, respectivamente, de entidades definidas en la generalización de la abstracción, X integrará a Y solo en el caso de que Y no tenga atributos específicos.

Una tabla también puede integrarse en la unión de otras dos tablas, cuando esto ocurre, la eliminación de la tabla puede incidir en una pérdida de eficiencia a la hora de recuperar información, aunque el coste a la hora de añadir nuevos datos o actualizarlos se vea mejorado. Este elemento de compensación deberá ser analizado durante la fase del diseño físico con respecto a los requerimientos de proceso y determinar si la eliminación de la tabla integrada es razonable.

El objetivo final es que después de la transformación del modelo ER al modelo relacional todas las relaciones se encuentren en 3FN.

### 5.2. Introducción a la transformación de entidades e interrelaciones

A estas alturas ya sabemos que los elementos básicos del modelo ER son las entidades y las interrelaciones.

Cuando pasamos o traducimos una entidad al modelo relacional origina una relación.

El caso de las interrelaciones es diferente ya que pueden ocurrir dos cosas al transformarlas: podemos obtener claves foráneas de alguna relación ya creada o pueden generar una nueva relación. Debemos tener en cuenta su grado y conectividad para poder decidir cual será la transformación adecuada:

1. De las interrelaciones binarias 1:1 y 1:N obtendremos claves foráneas
2. De las interrelaciones binarias M:N y todas las n-arias obtendremos nuevas relaciones.

El proceso de transformación lo comenzaremos transformando todas las entidades del modelo ER adecuadamente. Como ya he indicado más arriba cada entidad del modelo ER se transformará en una relación del modelo relacional, queda claro que los atributos de la entidad lo serán de la relación y, por lo tanto, la clave primaria de la entidad será la clave primaria de la relación.

Cuando tratemos la transformación de entidades débiles deberemos tener en cuenta que la clave foránea que se origina de la interrelación 1:N debe formar parte de la clave primaria de la relación correspondiente a la entidad débil.

Cuando realicemos la transformación de una generalización-especialización deberemos tener en cuenta que cada una de las entidades superclase y subclase se transformarán en una relación, la relación de la entidad superclase tendrá como clave primaria la de la entidad superclase y los atributos comunes. Las relaciones que se obtengan de las entidades subclases tendrán como clave primaria la clave de la superclase y los atributos específicos de la subclase.

Terminada la transformación de las entidades en relaciones continuamos con la transformación de todas las interrelaciones en las que intervienen las entidades.

En el modelo ER propuesto podemos encontrar interrelaciones binarias del tipo 1:1, 1:N y N:M y también tenemos una interrelación recursiva.

En la siguiente tabla podemos ver un resumen de la transformación del modelo ER al modelo relacional en lo que afecta a nuestro caso:

<b>Elemento modelo ER</b>	<b>Transformación al modelo relacional</b>
Entidad	Relación
Interrelación 1:1	Clave foránea
Interrelación 1:N	Clave foránea
Interrelación M:N	Relación
Interrelación recursiva	Como en las no recursivas: <ul style="list-style-type: none"> <li>• Clave foránea para binarias 1:1 y 1:N</li> <li>• Relación para M:N y n-arias</li> </ul>
Entidad débil	La clave foránea de la interrelación identificadora forma parte de la clave primaria
Generalización-especialización	<ul style="list-style-type: none"> <li>• Relación para la entidad superclase</li> <li>• Relación para cada una de las entidades subclase</li> </ul>

### 5.3. Transformación al modelo relacional

En este apartado realizaremos la transformación del diseño conceptual de la base de datos Biblioteca al modelo relacional.

Lo primero a realizar es la transformación de todas las entidades en relaciones y todas las relaciones 1:1 y 1:N en claves foráneas de estas relaciones:

**Empresas** (NIF, Sector, Nombre, Teléfono, E\_mail, Dirección)

**Personas** (ID, Nombre, Apellidos, Teléfono, E\_mail, Dirección, Fecha\_Nacimiento)

**Empleados** (ID, ID\_Ubicacion)

{ID } clave foránea hacia Personas(ID).

{ID\_Ubicacion} clave foránea hacia Sectores(ID\_Ubicacion).

**Usuarios** (ID, Estado, Días\_Penalización)

{ID } clave foránea hacia Personas(ID).

**Sectores** (ID\_Ubicacion, Edificio, Planta, Sector)

**Temas** (ID\_Tema, Descripción, ID\_Ubicacion)

{ID\_Ubicacion} clave foránea hacia Sectores(ID\_Ubicacion).

**Subtemas** (ID\_Tema, ID\_Subtema, Descripción)

{ ID\_Tema } clave foránea hacia Temas(ID\_Tema).

**Recursos** (ID\_Recurso, Tipo, Título, Autor, Palabras\_Clave, País\_Origen, ID\_Tema, ID\_Subtema)

{ ID\_Tema, ID\_Subtema } clave foránea hacia Subtemas(ID\_Tema, ID\_Subtema).

**Documentos** (ID\_Recurso, Soporte, Tipo\_Papel)

{ID\_Recurso} clave foránea hacia Recursos(ID\_Recurso).

**Mapas**(ID\_Recurso, Soporte)

{ID\_Recurso} clave foránea hacia Recursos(ID\_Recurso).

**Libros**(ID\_Recurso, ISBN ,Soporte, Número\_Páginas, NIF)

{ID\_Recurso} clave foránea hacia Recursos(ID\_Recurso).

{NIF} clave foránea hacia Empresas(NIF).

**Videos** (ID\_Recurso, Soporte, Duración, NIF)

{ID\_Recurso} clave foránea hacia Recursos(ID\_Recurso).

{NIF} clave foránea hacia Empresas(NIF).

**Música**(ID\_Recurso, Soporte, Duración, Número\_Pistas, NIF)

{ID\_Recurso} clave foránea hacia Recursos(ID\_Recurso).

{NIF} clave foránea hacia Empresas(NIF).

**Copias** (ID\_Recurso, ID\_Copia, edición, estado)

{ID\_Recurso} clave foránea hacia Recursos(ID\_Recurso).

**Retiradas** (ID\_Recurso, ID\_Copia)

{ID\_Recurso, ID\_Copia} clave foránea hacia Copias(ID\_Recurso, ID\_Copia).

**Prestables** (ID\_Recurso, ID\_Copia, préstamo, días)

{ID\_Recurso, ID\_Copia} clave foránea hacia Copias(ID\_Recurso, ID\_Copia).

**Nuevos\_Recursos** (ID\_Recurso, Tipo, Título, Autor, Palabras\_Clave, País\_Origen, Tema, Subtema)

{Tema, Subtema} clave foránea hacia Subtemas(Tema, Subtema).

**Nuevos\_Documentos** (ID\_Recurso, Soporte, Tipo\_Papel)

{ID\_Recurso} clave foránea hacia Nuevos\_Recursos(ID\_Recurso).

**Nuevos\_Mapas**(ID\_Recurso, Soporte)

{ID\_Recurso} clave foránea hacia Recursos(ID\_Recurso).

Nuevos\_Libros(ID\_Recurso, ISBN ,Soporte, Número\_Páginas)  
{ID\_Recurso} clave foránea hacia Nuevos\_Recurros(ID\_Recurso).

Nuevos\_Videos (ID\_Recurso, Soporte, Duración)  
{ID\_Recurso} clave foránea hacia Nuevos\_Recurros(ID\_Recurso).

Nuevos\_Música(ID\_Recurso, Soporte, Duración, Número\_Pistas)  
{ID\_Recurso} clave foránea hacia Nuevos\_Recurros(ID\_Recurso).

A continuación procedemos a transformar las interrelaciones M:N que tenemos en nuestro modelo en relaciones:

Referencia (ID\_Recurso, ID\_Recurso\_Referenciado)  
{ID\_Recurso} clave foránea hacia Recursos(ID\_Recurso).  
{ID\_Recurso\_Referenciado} clave foránea hacia Recursos(ID\_Recurso).

Lista\_Espera (ID\_Usuario, ID\_Recurso, Fecha\_Hora\_Solicitud, Dias\_Espera)  
{ ID\_Usuario } clave foránea hacia Usuarios(ID).  
{ID\_Recurso } clave foránea hacia Recursos(ID\_Recurso).

Préstamos (ID\_Usuario, ID\_Recurso, ID\_Copia, Fecha\_Devolución)  
{ ID\_Usuario } clave foránea hacia Usuarios(ID).  
{ID\_Recurso, ID\_Copia} clave foránea hacia Copias(ID\_Recurso, ID\_Copia).

Solicitados (ID\_Usuario, ID\_Recurso, Fecha\_Solicitud)  
{ ID\_Usuario } clave foránea hacia Usuarios(ID).  
{ID\_Recurso } clave foránea hacia Nuevos\_Recurros(ID\_Recurso).

Para elegir las claves primarias se ha tenido en cuenta la conectividad de las interrelaciones.

#### **5.4. Definición del modelo de datos**

En el apartado anterior hemos obtenido todas las relaciones que van a formar parte de nuestro modelo, así como los atributos que forman parte de ellas. El siguiente trabajo ha realizar será el determinar de que tipo es cada uno de los atributos que forman las entidades (numéricos, char, varchar2, data,...), una vez realizado este trabajo estaremos en disposición de realizar un cálculo del espacio que vamos a necesitar para nuestra base de datos, pero esto es algo que realizaremos en el siguiente punto.

Para la realización de esta tarea seguiremos el orden del apartado anterior, de esta forma veremos de que clase son los atributos de la tabla empresa y que guarda cada uno de ellos para luego continuar con el resto de las relaciones.

##### **Empresas**

- NIF
  - Campo alfanumérico de longitud 9.
  - Descripción: NIF de la empresa.
- Sector
  - Campo alfanumérico de longitud 1
  - Valores posibles:
    1. E (Editorial (libros)).
    2. A (Audiovisual (música y películas)).
    3. C (Cinematográfico (videos)).
    4. M (Musical (música)).
  - Descripción: Sector al que pertenece la empresa.
- Nombre
  - Campo alfanumérico de longitud 30
  - Descripción: Razón social de la empresa

- Teléfono
  - Campo numérico de longitud 15
  - Descripción: Teléfono de la empresa
- E\_mail
  - Campo alfanumérico de longitud 25
  - Descripción: Correo electrónico de la empresa
- Dirección
  - Campo alfanumérico de longitud 80
  - Descripción: Dirección postal de la empresa  
(Podría dividirse la dirección en dirección (calle y número), código postal, población y provincia, apareciendo nuevos atributos y nuevas relaciones en nuestro modelo)

#### Personas

- ID
  - Campo alfanumérico de longitud 9.
  - Descripción: identificador de la persona (DNI, pasaporte, ..).
- Nombre
  - Campo alfanumérico de longitud 15
  - Descripción: Nombre de la persona.
- Apellidos
  - Campo alfanumérico de longitud 30
  - Descripción: Apellidos de la persona
- Fecha Nacimiento
  - Campo de tipo date
  - Descripción: Fecha de nacimiento de la persona.
- Teléfono
  - Campo numérico de longitud 15
  - Descripción: Teléfono de la Persona
- E\_mail
  - Campo alfanumérico de longitud 25
  - Descripción: Correo electrónico de la persona
- Dirección
  - Campo alfanumérico de longitud 80
  - Descripción: Dirección postal de la persona  
(Al igual que ocurre con Empresas, podría dividirse la dirección en dirección (calle y número), código postal, población y provincia, apareciendo nuevos atributos y nuevas relaciones en nuestro modelo)

#### Empleados

- ID
  - Campo alfanumérico de longitud 9.
  - Descripción: identificador del empleado (DNI).
- ID\_Ubicación
  - Campo alfanumérico de longitud 7.
  - Descripción: Ubicación en el que trabaja el empleado.

#### Usuarios

- ID
  - Campo alfanumérico de longitud 9.
  - Descripción: identificador del usuario (DNI, pasaporte, ..).
- Estado
  - Campo alfanumérico de longitud 1
  - Valores posibles:
    1. A (Alta)
    2. B (Baja)
  - Descripción: estado en el que se encuentra un usuario
- Días\_Penalización
  - Campo numérico de longitud 3
  - Descripción: Acumulador de los días de penalización de un usuario (cada día de penalización equivale a un día de retraso en la devolución de algún recurso).

### Sectores

- ID\_Ubicación
  - Campo alfanumérico de longitud 7.
  - Descripción: Identificación de un sector dado..
- Edificio
  - Campo alfanumérico de longitud 15.
  - Descripción: Edificio en el que trabaja el usuario.
- Planta
  - Campo numérico de longitud 2.
  - Descripción: Planta del edificio.
- Sector
  - Campo alfanumérico de longitud 1.
  - Descripción: sector de la planta.

### Temas

- ID\_Tema
  - Campo numérico de longitud 5
  - Descripción: nombre del tema
- Descripción
  - Campo alfanumérico de longitud 30
  - Descripción: nombre del tema
- ID\_Ubicación
  - Campo alfanumérico de longitud 7.
  - Descripción: Ubicación donde podemos encontrar el tema.

### Subtemas

- ID\_Tema
  - Campo numérico de longitud 5
  - Descripción: nombre del tema
- ID\_Subtema
  - Campo numérico de longitud 5
  - Descripción: nombre del tema
- Descripción
  - Campo alfanumérico de longitud 40
  - Descripción: nombre del subtema

### Recursos

- ID\_Recurso
  - Campo numérico de longitud 7
  - Descripción: Identifica de manera única al recurso.
- Tipo
  - Campo alfanumérico de longitud 2
  - Valores posibles
    1. DO (Documento)
    2. MA (Mapa)
    3. LI (Libro)
    4. VI (Vídeo)
    5. MU (Música)
  - Descripción: Subtipo al que pertenece el recurso
- Titulo
  - Campo alfanumérico de longitud 80
  - Descripción: título del recurso
- Autor
  - Campo alfanumérico de longitud 250
  - Descripción: Autor o autores del recurso.
- Palabras\_clave
  - Campo alfanumérico de longitud 250
  - Descripción: palabras que identifican al recurso

- País\_Origen
  - Campo alfanumérico de longitud 60
  - Descripción: País de origen del recurso.
- ID\_Tema
  - Campo numérico de longitud 5
  - Descripción: nombre del subtema
- ID\_Subtema
  - Campo numérico de longitud 5
  - Descripción: nombre del subtema

#### Documentos

- ID\_Recurso
  - Campo numérico de longitud 7
  - Descripción: Identifica de manera única al recurso.
- Soporte
  - Campo alfanumérico de longitud 25
  - Descripción: soporte en el que se encuentra el documento (papel, CD, etc.)
- Tipo\_Papel
  - Campo alfanumérico de longitud 25
  - Descripción: Si el documento está en papel tipo del mismo.

#### Mapas

- ID\_Recurso
  - Campo numérico de longitud 7
  - Descripción: Identifica de manera única al recurso.
- Soporte
  - Campo alfanumérico de longitud 25
  - Descripción: soporte en el que se encuentra el mapa (papel, CD, etc.)

#### Libros

- ID\_Recurso
  - Campo numérico de longitud 7
  - Descripción: Identifica de manera única al recurso.
- ISBN
  - Campo numérico de 14 posiciones
  - Descripción: ISBN que identifica al libro
- Soporte
  - Campo alfanumérico de longitud 25
  - Descripción: soporte en el que se encuentra el libro (papel, CD, etc.)
- Número\_Páginas
  - Campo numérico de longitud 4
  - Descripción: páginas que contiene el libro
- NIF
  - Campo alfanumérico de longitud 9.
  - Descripción: NIF de la editorial.

#### Videos

- ID\_Recurso
  - Campo numérico de longitud 7
  - Descripción: Identifica de manera única al recurso.
- Soporte
  - Campo alfanumérico de longitud 25
  - Descripción: soporte en el que se encuentra el video (cinta, dvd, etc.)
- Duración
  - Campo numérico de 3 posiciones
  - Descripción: duración en minutos del video
- NIF

- Campo alfanumérico de longitud 9.
- Descripción: NIF de la productora.

### Música

- ID\_Recurso
  - Campo numérico de longitud 7
  - Descripción: Identifica de manera única al recurso.
- Soporte
  - Campo alfanumérico de longitud 25
  - Descripción: soporte en el que se encuentra la música (vinilo, casete, CD, etc.)
- Duración
  - Campo numérico de 3 posiciones
  - Descripción: duración en minutos del recurso musical
- Número\_Pistas
  - Campo numérico de 3 posiciones
  - Descripción: número de piezas musicales que componen el recurso
- NIF
  - Campo alfanumérico de longitud 9.
  - Descripción: NIF de la productora.

### Copias

- ID\_Recurso
  - Campo numérico de longitud 7
  - Descripción: Identifica de manera única al recurso.
- ID\_Copia
  - Campo numérico de longitud 2
  - Descripción: número de copia del recurso.
- Edición
  - campo numérico de longitud 4
  - Descripción: año de edición de la copia.
- Estado
  - Campo alfanumérico de longitud 1
  - Valores posibles
    1. E (Excelente)
    2. B (Bueno)
    3. R (Regular)
    4. M (Malo)
    5. P (Pésimo)
  - Descripción: Estado en el que se encuentra la copia

### Retiradas

- ID\_Recurso
  - Campo numérico de longitud 7
  - Descripción: Identifica de manera única al recurso.
- ID\_Copia
  - Campo numérico de longitud 2
  - Descripción: número de copia del recurso.

### Prestables

- ID\_Recurso
  - Campo numérico de longitud 7
  - Descripción: Identifica de manera única al recurso.
- ID\_Copia
  - Campo numérico de longitud 2
  - Descripción: número de copia del recurso.
- Préstamo
  - Campo alfanumérico de longitud 1
  - Valores posibles
    1. S
    2. N

- Descripción: Indica la disponibilidad del libro para el préstamo domiciliario
- Días
  - Campo numérico de longitud 2.
  - Descripción: Si el campo anterior tiene como valor S, indica el número de días que puede prestarse un libro a domicilio.

#### Nuevos\_Recursos

- ID\_Recurso
  - Campo numérico de longitud 7
  - Descripción: Identifica de manera única al recurso.
- Tipo
  - Campo alfanumérico de longitud 2
  - Valores posibles
    1. DO (Documento)
    2. MA (Mapa)
    3. LI (Libro)
    4. VI (Vídeo)
    5. MU (Música)
  - Descripción: Subtipo al que pertenece el recurso
- Titulo
  - Campo alfanumérico de longitud 80
  - Descripción: título del recurso
- Autor
  - Campo alfanumérico de longitud 250
  - Descripción: Autor o autores del recurso.
- Palabras\_clave
  - Campo alfanumérico de longitud 250
  - Descripción: palabras que identifican al recurso
- País\_Origen
  - Campo alfanumérico de longitud 60
  - Descripción: País de origen del recurso.
- ID\_Tema
  - Campo numérico de longitud 5
  - Descripción: nombre del subtema
- ID\_Subtema
  - Campo numérico de longitud 5
  - Descripción: nombre del subtema

#### Nuevos\_Documentos

- ID\_Recurso
  - Campo numérico de longitud 7
  - Descripción: Identifica de manera única al recurso.
- Soporte
  - Campo alfanumérico de longitud 25
  - Descripción: soporte en el que se encuentra el documento (papel, CD, etc.)
- Tipo\_Papel
  - Campo alfanumérico de longitud 25
  - Descripción: Si el documento está en papel tipo del mismo.

#### Nuevos\_Mapas

- ID\_Recurso
  - Campo numérico de longitud 7
  - Descripción: Identifica de manera única al recurso.
- Soporte
  - Campo alfanumérico de longitud 25
  - Descripción: soporte en el que se encuentra el mapa (papel, CD, etc.)

#### Nuevos\_Libros

- ID\_Recurso
  - Campo numérico de longitud 7

- Descripción: Identifica de manera única al recurso.
- ISBN
  - Campo numérico de 14 posiciones
  - Descripción: ISBN que identifica al libro
- Soporte
  - Campo alfanumérico de longitud 25
  - Descripción: soporte en el que se encuentra el libro (papel, CD, etc.)
- Número\_Páginas
  - Campo numérico de longitud 4
  - Descripción: páginas que contiene el libro
- NIF
  - Campo alfanumérico de longitud 9.
  - Descripción: NIF de la editorial.

#### Nuevos\_Videos

- ID\_Recurso
  - Campo numérico de longitud 7
  - Descripción: Identifica de manera única al recurso.
- Soporte
  - Campo alfanumérico de longitud 25
  - Descripción: soporte en el que se encuentra el video (cinta, dvd, etc.)
- Duración
  - Campo numérico de 3 posiciones
  - Descripción: duración en minutos del video
- NIF
  - Campo alfanumérico de longitud 9.
  - Descripción: NIF de la productora.

#### Nuevos\_Música

- ID\_Recurso
  - Campo numérico de longitud 7
  - Descripción: Identifica de manera única al recurso.
- Soporte
  - Campo alfanumérico de longitud 25
  - Descripción: soporte en el que se encuentra la música (vinilo, casete, CD, etc.)
- Duración
  - Campo numérico de 3 posiciones
  - Descripción: duración en minutos del recurso musical
- Número\_Pistas
  - Campo numérico de 3 posiciones
  - Descripción: número de piezas musicales que componen el recurso
- NIF
  - Campo alfanumérico de longitud 9.
  - Descripción: NIF de la productora.

#### Referencia

- ID\_Recurso
  - Campo numérico de longitud 7
  - Descripción: Identifica de manera única al recurso.
- ID\_Recurso\_Referenciado
  - Campo numérico de longitud 7
  - Descripción: Identifica de manera única al recurso.

#### Lista\_Espera

- ID\_Usuario
  - Campo alfanumérico de longitud 9.
  - Descripción: identificación del usuario, se corresponde con su ID.
- ID\_Recurso

- Campo numérico de longitud 7
- Descripción: Identifica de manera única al recurso.
- Fecha\_Solicitud y Hora\_Solicitud
  - Campo de tipo TIMESTAMP
  - Descripción: Fecha y hora de solicitud del recurso no disponible, este campo se rellenará mediante la función SYSTIMESTAMP.
- Días\_Espera
  - Campo numérico de longitud 2
  - Descripción: Número de días que ha permanecido en la lista de espera.

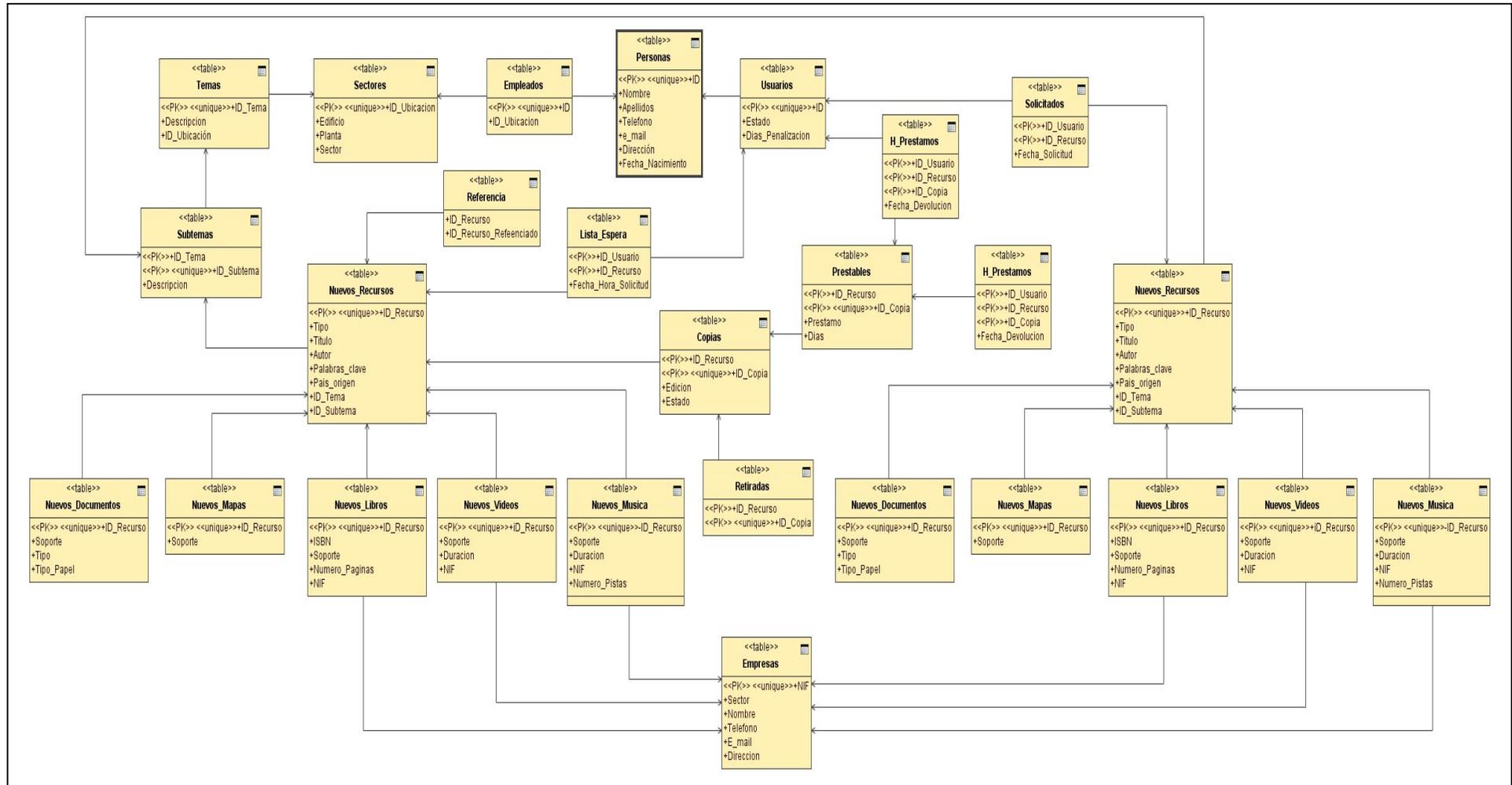
#### Préstamos

- ID\_Usuario
  - Campo alfanumérico de longitud 9.
  - Descripción: identificación del usuario, se corresponde con su ID.
- ID\_Recurso
  - Campo numérico de longitud 7
  - Descripción: Identifica de manera única al recurso.
- ID\_Copia
  - Campo numérico de longitud 2
  - Descripción: número de copia del recurso.
- Fecha\_Devolución
  - Campo de tipo DATE
  - Descripción: almacena la fecha de devolución del recurso.

#### Solicitados

- ID\_Usuario
  - Campo alfanumérico de longitud 9.
  - Descripción: identificación del usuario, se corresponde con su ID.
- ID\_Recurso
  - Campo numérico de longitud 7
  - Descripción: Identifica de manera única al recurso.
- Fecha\_Solicitud
  - Campo de tipo DATE
  - Descripción: almacena la fecha de solicitud del recurso.

### 5.5 Esquema de las tablas de la B.D.



## 6. Diseño de la base de datos Física

Una vez obtenido el diseño lógico de una base de datos debemos pasar al diseño físico de la misma, pasando por el nivel virtual. El cómo realizar esta transformación va a depender de las características propias del SGBD elegido, en este caso Oracle 9i, y las características propias de cada SGBD reflejan su entorno de funcionamiento:

- Características del servidor elegido.
- Sistema operativo y programación básica.
- Diseño del SGBD.

Cada SGBD posee un lenguaje propio, hecho a la medida por su constructor, para implementar el diseño físico de la base de datos, de acuerdo con las características del entorno, y para obtener el máximo rendimiento del servidor, del sistema operativo y del mismo gestor.

Podríamos considerar este lenguaje propio como una aplicación del lenguaje SQL estándar, con las cláusulas propias que el gestor necesitará para definir los componentes del diseño físico. Podemos comprobar fácilmente que hay un gran parecido entre muchos de los componentes de los diferentes gestores de base de datos.

Para la realización de este punto ha sido necesaria la consulta, bastante asidua, de las referencias bibliográficas [ORA01] a [ORA09].

### 6.1. Definición del tamaño de las tablas

#### 6.1.1. Algunas consideraciones previas

Para definir el tamaño de las tablas debemos tener en cuenta que los campos numéricos siempre ocuparán 4 bytes, que las fechas ocupan 8 bytes y que los campos alfanuméricos ocupan tantos bytes como caracteres se han definido.

PCTFREE es una parte del bloque de datos que no se ha rellenado con filas cuando estas han sido insertadas en una tabla, sino que se reserva para posteriores actualizaciones de las filas de dicho bloque.

PCTUSED es el porcentaje de espacio en un bloque de datos que Oracle intenta mantener lleno. Si el porcentaje cae por debajo del valor PCTUSED, entonces el bloque se añade a la lista de bloques libres en el segmento.

PCTINCREASE está diseñado para minimizar el número de extensiones en las tablas que aumentan de tamaño. Un valor diferente de cero en este parámetro puede ser muy peligroso, ya que provoca que el tamaño de cada extensión sucesiva aumente geométricamente de acuerdo con el factor PCTINCREASE especificado.

INITIAL especifica el tamaño inicial de extensión.

NEXT especifica el tamaño de la siguiente extensión.

MINEXTENTS es el número mínimo de extensiones.

MAXEXTENTS es el número máximo de extensiones, el valor más alto que le podemos dar a este parámetro es de 200.

Definiremos los valores de ocupación (PCTUSED) y libre (PCTFREE) para todas las tablas igual, estos valores serán los siguientes:

PCTFREE	5%
PCTUSED	95%

La elección de estos valores no se ha hecho de forma aleatoria sino basándome en que las tablas, en la mayor parte de los casos, se añadirá información o se modificará la ya introducida, como puede ser el cambio de estado de conservación de una copia. Debemos tener en cuenta que el borrado de datos en la mayoría de las tablas no va a estar permitido para que la integridad de la base de datos no se vea comprometida. Todo esto provocará que no exista mucha fragmentación de datos y por este motivo el PCTFREE es tan bajo.

Por el enunciado propuesto (Anexo I) podemos suponer que nos encontramos ante una biblioteca de un tamaño considerable, con un gran número de recursos. El número de usuarios también podemos suponerlo alto. Esto implica que habrá mucho movimiento de préstamos, que puede haber una lista de espera importante, así como una importante lista de solicitudes de nuevos recursos. Puede tratarse, por ejemplo, la Biblioteca de una Universidad.

Además de lo expuesto en el párrafo anterior me basaré también en mi propia experiencia. Por mi trabajo tengo acceso a la base de datos de la Biblioteca Municipal de Mislata (Valencia), se trata de una biblioteca que tiene en la actualidad un catalogo cercano a los 50000 recursos, la mayoría de ellos libros, quedando en minoría los documentos y videos (no existen los recursos de tipo mapa y música). El numero de usuarios está próximo a los 10000, por la naturaleza de la población residente el número de usuarios aumenta poco a lo largo del año y casi podemos decir que las altas compensan a las bajas manteniéndose más o menos estable. El incremento anual de los recursos esta próximo al 10%, por otra parte los recursos que se retiran al cabo del año por mal estado estarán entre el 2% o el 3%. De esta forma el crecimiento real estará próximo al 7%, cada año se incrementa en cerca de 3500 recursos. Hasta no hace mucho la biblioteca contaba con 4 sedes, la central, dos agencias de lectura y la biblio-metro (una agencia de lectura en una de las estaciones de metro de la población), hoy en día solo está la sede central y una agencia de lectura, la segunda esta en plena fase de reforma y biblio-metro cerró después del verano.

Tomo como punto de partida para hacer un cálculo del tamaño inicial que podemos suponer que ocuparan cada una de las diferentes tablas de nuestro modelo y una previsión de su posterior incremento lo expuesto en los dos párrafos anteriores..

### 6.1.2. Empresas

Empresas editoriales, productoras cinematográficas y productoras musicales hay muchas, incluso hoy en día hay muchas empresas que no solo se dedican a un sector. Empresas editoriales en el mundo hay más de 5000, a estas habrá que sumar las productoras cinematográficas y musicales que existan (un número bastante menor entre ambas he calculado unas 250). Podemos suponer que inicialmente este fichero consta de 600 registros, esto es un poco más del 10% del total de empresas relacionadas.

La longitud de un registro de esta tabla es de 149 bytes.

Tamaño inicial de la tabla:  $149 \times 600 = 89400$  bytes  $\approx 87,30$  Kbytes  $\Rightarrow 88$  Kb

El valor de parámetro NEXT será un 10% del valor de INITIAL.

### 6.1.3. Personas

En esta tabla, por su definición, guardamos los datos comunes a empleados y a usuarios. Supongamos que el número de empleados es de 50 (hemos supuesto que se trata de una Biblioteca grande) y que todos los matriculados en la universidad son usuarios potenciales de la biblioteca, suponemos que en el momento de poner en marcha la base de datos tendremos entre matriculados y personal docente un total de 10000 personas.

La longitud de un registro de esta tabla es de 171 bytes.

Tamaño inicial de la tabla:  $171 \cdot 10050 = 1718550$  bytes  $\approx 1,63$  Mbytes  $\Rightarrow 2$  Mb.

No podemos saber con seguridad el número de nuevos alumnos que se matriculen cada año, si damos un valor al parámetro NEXT del 10% del valor inicial puede ser que nos quedemos cortos, en este caso podemos hacer uso del parámetro PCTINCREASE al que podemos dar un valor de 2%.

#### 6.1.4. Empleados

Tabla que guarda los datos específicos de los empleados de la biblioteca, hemos supuesto inicialmente que los empleados de la biblioteca son 50.

La longitud de un registro de la tabla es de 16 bytes.

Tamaño inicial de la tabla:  $50 \cdot 16 = 800$  bytes  $\Rightarrow 1$  Kb.

Estableceremos el valor de NEXT a un 10% del valor de INITIAL.

#### 6.1.5. Usuarios

Tabla que guarda los datos específicos de los usuarios de la biblioteca, hemos supuesto inicialmente que los usuarios de la biblioteca son 10000 (todos los alumnos matriculados, más el personal docente).

La longitud de un registro de la tabla es de 14 bytes.

Tamaño inicial de la tabla:  $10000 \cdot 14 = 14000$  bytes  $\approx 13,67$  Kbytes  $\Rightarrow 14$  Kb.

Como hemos visto antes no podemos saber con seguridad el número de nuevos alumnos que se matriculen cada año y si damos un valor al parámetro NEXT del 10% del valor inicial puede ser que nos quedemos cortos, también daremos al parámetro PCTINCREASE un valor de 2%.

#### 6.1.6. Sectores

Tabla que guarda las diferentes ubicaciones que podemos encontrar en nuestra biblioteca, es una tabla que puede no variar mucho en su tamaño a lo largo del tiempo. Supongamos que la biblioteca se encuentra en 5 edificios de 3 plantas cada uno y que en cada planta tenemos 25 sectores, de esta forma los registros iniciales serán  $5 \cdot 3 \cdot 25 = 375$  posibles ubicaciones.

La longitud de un registro de la tabla es de 25 bytes.

Tamaño inicial de la tabla:  $375 \cdot 25 = 9375$  bytes  $\approx 9,15$  Kbytes  $\Rightarrow 10$  Kb.

Estableceremos el valor de NEXT a un 10% del valor de INITIAL.

#### 6.1.7 Temas

Tabla que guarda los diferentes temas en los que podemos agrupar los recursos que posee nuestra biblioteca. Temas posibles hay muchos, podemos suponer que esta tabla recogerá todos o casi todos los temas posibles y lo más que haremos será añadir los nuevos temas que puedan aparecer a lo largo de los años. No será una tabla que aumente mucho de tamaño. Supongamos que están todos los temas introducidos y que en total son 1000.

La longitud de un registro de la tabla es de 41 bytes.

Tamaño inicial de la tabla:  $1000 \cdot 41 = 4100$  bytes  $\Rightarrow$  4 Kb.

Estableceremos el valor de NEXT a un 10% del valor de INITIAL.

### 6.1.8. Subtemas

Estamos en el mismo caso que con la tabla anterior, si hemos supuesto que partimos de 1000 temas iniciales, establecemos el tamaño inicial de esta tabla en 10000 registros. Es muy posible que esta tabla crezca más deprisa que la de temas, ya que es más fácil, desde mi punto de vista, añadir una división a un tema que crear uno nuevo.

La longitud de un registro de la tabla es de 48 bytes.

Tamaño inicial de la tabla:  $10000 \cdot 48 = 480000$  bytes  $\approx$  468,75Kbytes  $\Rightarrow$  500Kb.

Estableceremos el valor de NEXT a un 10% del valor de INITIAL.

### 6.1.9. Recursos

Nos encontramos ante la tabla principal de nuestra base de datos, tabla que crecerá sin cesar, donde iremos añadiendo todos los recursos que la biblioteca vaya adquiriendo y de donde no podremos borrar nada, suponemos que nuestra biblioteca inicialmente posee un total de 100000 recursos independientemente de su tipo. Si la biblioteca adquiere todo los recursos que salen al mercado el crecimiento anual podemos suponerlo mayor del 20%. Para las siguientes tablas supondremos que el 20% de los recursos son documentos, un 5% son mapas, un 50% son libros, un 10% son videos y un 15% son música.

La longitud de un registro de la tabla es de 654 bytes.

Tamaño inicial tabla:  $100000 \cdot 654 = 65400000$  bytes  $\approx$  62.37 Mbytes  $\Rightarrow$  63 Mb.

Estableceremos el valor de NEXT a un 20% del valor de INITIAL y como puede que nos quedemos cortos, también daremos al parámetro PCTINCREASE un valor de 5%.

### 6.1.10. Documentos

Tabla dependiente de la tabla Recursos, contiene los recursos de tipo documento, deberemos establecer todos los parámetros de igual manera.

La longitud de un registro de la tabla es de 54 bytes.

Tamaño inicial de la tabla:  $20000 \cdot 54 = 1080000$  bytes  $\approx$  1 Mbytes  $\Rightarrow$  1,5 Mb.

Estableceremos el valor de NEXT a un 20% del valor de INITIAL y al parámetro PCTINCREASE le damos un valor de 5%.

### 6.1.11. Mapas

Tabla dependiente de la tabla Recursos, contiene los recursos de tipo mapa, deberemos establecer todos los parámetros de igual manera.

La longitud de un registro de la tabla es de 29 bytes.

Tamaño inicial de la tabla:  $5000 \cdot 29 = 145000$  bytes  $\approx$  141,6 Kbytes  $\Rightarrow$  145 Kb.

Estableceremos el valor de NEXT a un 20% del valor de INITIAL y al parámetro PCTINCREASE le damos un valor de 5%.

### 6.1.12. Libros

Tabla dependiente de la tabla Recursos, contiene los recursos de tipo libro, deberemos establecer todos los parámetros de igual manera.

La longitud de un registro de la tabla es de 46 bytes.

Tamaño inicial de la tabla:  $50000 \cdot 46 = 2300000$  bytes  $\approx 2,19$  Mbytes  $\Rightarrow 2,5$  Mb.

Estableceremos el valor de NEXT a un 20% del valor de INITIAL y al parámetro PCTINCREASE le damos un valor de 5%.

### 6.1.13. Videos

Tabla dependiente de la tabla Recursos, contiene los recursos de tipo video, deberemos establecer todos los parámetros de igual manera.

La longitud de un registro de la tabla es de 42 bytes.

Tamaño inicial de la tabla:  $10000 \cdot 42 = 420000$  bytes  $\approx 410,1$  Kbytes  $\Rightarrow 450$  Kb.

Estableceremos el valor de NEXT a un 20% del valor de INITIAL y al parámetro PCTINCREASE le damos un valor de 5%.

### 6.1.14. Música

Tabla dependiente de la tabla Recursos, contiene los recursos de tipo música, deberemos establecer todos los parámetros de igual manera.

La longitud de un registro de la tabla es de 46 bytes.

Tamaño inicial de la tabla:  $15000 \cdot 46 = 690000$  bytes  $\approx 673,8$  Kbytes  $\Rightarrow 700$  Kb.

Estableceremos el valor de NEXT a un 20% del valor de INITIAL y al parámetro PCTINCREASE le damos un valor de 5%.

### 6.1.15. Copias

Tabla íntimamente relacionada con la tabla Recursos y que guarda todas las copias existentes de cada uno de los recursos que hay en la biblioteca. Partimos de 100000 recursos iniciales, de los que podemos tener uno, dos, tres o más copias, será una tabla con un gran número de registros y que crecerá posiblemente más que la tabla de recursos, por poner un punto de partida, supondremos que inicialmente esta tabla contendrá 300000 registros.

La longitud de un registro de la tabla es de 13 bytes.

Tamaño inicial de la tabla:  $300000 \cdot 13 = 3900000$  bytes  $\approx 3,72$  Mbytes  $\Rightarrow 4$  Mb.

Estableceremos el valor de NEXT a un 20% del valor de INITIAL y al parámetro PCTINCREASE le damos un valor de 5%.

### 6.1.16. Retiradas

Tabla que guarda las copias retiradas, supongamos que un 5% del total de las copias están en un estado que no permite su préstamo, además, como en el caso de la tabla Recursos, deberemos establecer todos los parámetros de igual manera que lo hemos hecho en la tabla Copias.

La longitud de un registro de la tabla es de 8 bytes.

Tamaño inicial de la tabla:  $15000 \times 8 = 120000 \approx 117,2 \text{ Kbytes} \Rightarrow 118 \text{ Kb}$ .

Estableceremos el valor de NEXT a un 20% del valor de INITIAL y como puede que nos quedemos cortos, también daremos al parámetro PCTINCREASE un valor de 5%.

### 6.1.17. Prestables

Tabla que guarda todas las copias que tienen carácter de prestable, bien en sala o en domicilio, que será el 95% de las copias. Deberemos establecer todos los parámetros de igual manera que lo hemos hecho en la tabla Copias.

La longitud de un registro de la tabla es de 13 bytes.

Tamaño inicial de la tabla:  $285000 \times 13 = 3705000 \text{ bytes} \approx 3,53 \text{ Mbytes} \Rightarrow 4 \text{ Mb}$ .

Estableceremos el valor de NEXT a un 20% del valor de INITIAL y como puede que nos quedemos cortos, también daremos al parámetro PCTINCREASE un valor de 5%.

### 6.1.18. Nuevos\_Recursos

En esta tabla se irán almacenando las solicitudes de recursos que vayan haciendo los usuarios. Podemos pensar que esta tabla no crecerá demasiado ya que cada vez que se compren los recursos en ella almacenados deberán desaparecer ya que los añadiremos a la tabla Recursos. Supongamos que las compras se realizan anualmente, el problema viene a la hora de dar un número de registros iniciales a esta tabla, no podemos saber de antemano el número de peticiones que van a realizar los usuarios. Por dar un valor inicial supongamos un tamaño inicial de 1000 registros. Para las siguientes tablas procederemos igual que hemos hecho con las derivadas de recursos: supondremos que el 20% de los nuevos recursos son documentos, un 5% son mapas, un 50% son libros, un 10% son videos y un 15% son música.

La longitud de un registro de la tabla es de 654 bytes.

Tamaño inicial de la tabla:  $1000 \times 654 = 654000 \text{ bytes} \approx 638,67 \text{ Kbytes} \Rightarrow 640 \text{ Kb}$ .

Estableceremos el valor de NEXT a un 20% del valor de INITIAL.

### 6.1.19. Nuevos\_Documentos

Tabla dependiente de la tabla Nuevos\_Recursos, contiene los recursos de tipo documento, deberemos establecer todos los parámetros de igual manera que en esta.

La longitud de un registro de la tabla es de 54 bytes.

Tamaño inicial de la tabla:  $200 \times 54 = 10800 \text{ bytes} \approx 10,55 \text{ kbytes} \Rightarrow 11 \text{ kb}$ .

Estableceremos el valor de NEXT a un 20% del valor de INITIAL.

### 6.1.20. Nuevos\_Mapas

Tabla dependiente de la tabla Nuevos Recursos, contiene los recursos de tipo mapa, deberemos establecer todos los parámetros de igual manera.

La longitud de un registro de la tabla es de 29 bytes.

Tamaño inicial de la tabla:  $50 \cdot 29 = 1450$  bytes  $\approx 1,4$  Kbytes  $\Rightarrow 2$  Kb.

Estableceremos el valor de NEXT a un 20% del valor de INITIAL.

### 6.1.21. Nuevos\_Libros

Tabla dependiente de la tabla Nuevos Recursos, contiene los recursos de tipo libro, deberemos establecer todos los parámetros de igual manera.

La longitud de un registro de la tabla es de 46 bytes.

Tamaño inicial de la tabla:  $500 \cdot 46 = 23000$  bytes  $\approx 22,46$  Kbytes  $\Rightarrow 23$  Kb.

Estableceremos el valor de NEXT a un 20% del valor de INITIAL.

### 6.1.22. Nuevos\_Videos

Tabla dependiente de la tabla Nuevos Recursos, contiene los recursos de tipo video, deberemos establecer todos los parámetros de igual manera.

La longitud de un registro de la tabla es de 42 bytes.

Tamaño inicial de la tabla:  $100 \cdot 42 = 4200$  bytes  $\approx 4,1$  Kbytes  $\Rightarrow 4,5$  Kb.

Estableceremos el valor de NEXT a un 20% del valor de INITIAL.

### 6.1.23. Nuevos\_Música

Tabla dependiente de la tabla Nuevos Recursos, contiene los recursos de tipo música, deberemos establecer todos los parámetros de igual manera.

La longitud de un registro de la tabla es de 46 bytes.

Tamaño inicial de la tabla:  $150 \cdot 46 = 6900$  bytes  $\approx 6,7$  Kbytes  $\Rightarrow 7$  Kb.

Estableceremos el valor de NEXT a un 20% del valor de INITIAL.

### 6.1.24. Referencia

Tabla que guarda las referencias entre Recursos, un recurso puede referenciar o no a otro recurso, un ejemplo podría ser la discografía de un grupo musical o las películas que haya podido dirigir un director determinado, podemos suponer que si el tamaño inicial de la tabla recursos es de 100000 registros esta podemos suponerla mayor. Como punto de partida supondremos que tiene 300000 registros y que se comportará de una manera parecida a la tabla Copias.

La longitud de un registro de la tabla es de 8 bytes.

Tamaño inicial de la tabla:  $300000 * 8 = 2400000$  bytes  $\approx 2,29$  Mbytes  $\Rightarrow 3$  Mb.

Estableceremos el valor de NEXT a un 20% del valor de INITIAL y como puede que nos quedemos cortos, también daremos al parámetro PCTINCREASE un valor de 5%.

### 6.1.25. Lista\_Espera

Tabla que guarda las solicitudes que hacen los usuarios de recursos que no están disponibles por estar en préstamo o en reparación. Aunque debiera ser una tabla de movimientos el enunciado del proyecto nos obliga a mantener los datos para luego poder obtener las estadísticas que se nos piden. Le calculamos un tamaño inicial de 1000 registros.

La longitud de un registro de la tabla es de 25 bytes.

Tamaño inicial de la tabla:  $1000 * 25 = 25000$  bytes  $\approx 24,4$  Kbytes  $\Rightarrow 25$  Kb.

Estableceremos el valor de NEXT a un 10% del valor de INITIAL.

### 6.1.26. Préstamos

Tabla que guarda los préstamos de recursos realizados y que además nos servirá como histórico de préstamos. Será una tabla que crezca continuamente, le daremos una capacidad inicial de 10000 registros. Como no podemos saber que incremento anual tendrá la tabla utilizaremos el parámetro PCTINCREASE.

La longitud de un registro de la tabla es de 25 bytes.

Tamaño inicial de la tabla:  $10000 * 25 = 250000$  bytes  $\approx 244,1$  Kbytes  $\Rightarrow 245$  Kb.

Estableceremos el valor de NEXT a un 20% del valor de INITIAL y como puede que nos quedemos cortos, también daremos al parámetro PCTINCREASE un valor de 5%.

### 6.1.27. Solicitados

Tabla que guarda los datos relacionados con las solicitudes que realizan los usuarios de recursos que no se encuentran en el catálogo y que pueden incluirse en futuras compras. Al igual que en el caso anterior le daremos un tamaño inicial de 10000 registros,

La longitud de un registro de la tabla es de 21 bytes.

Tamaño inicial de la tabla:  $10000 * 21 = 210000$  bytes  $\approx 205,1$  Kbytes  $\Rightarrow 206$  Kb.

Estableceremos el valor de NEXT a un 10% del valor de INITIAL.

## 6.2. Creación del Script de generación de la base de datos

Llegado a este punto es hora de preparar es Script que permitirá la creación de la base de datos objeto de este trabajo.

A la hora de crear la base de datos podemos optar entre usar el asistente de configuración de bases de datos que proporciona Oracle, de esta manera podremos estar seguros de que la nueva base de datos esta optimizada para funcionar con Oracle 9i. Para seguir este método

tan solo deberemos leer las instrucciones e ir pasando las diferentes pantallas hasta terminar el proceso. La segunda opción para crear la base de datos es hacerlo manualmente, como podremos ver a continuación.

Deberemos, en primer lugar, crear el directorio o carpeta que contendrá la nueva base de datos, le llamaremos TFC. Una vez realizada esta faena, copiaremos el fichero init.ora al nuevo directorio renombrándolo como init{SID}.ora, {SID} es el “Oracle System Identifier” que identifica a la instancia de base de datos que vamos a crear, en nuestro caso SID = biblios, luego crearemos el fichero initbiblios.ora.

El siguiente paso será hacer consistente este parámetro, para ello si trabajamos con el sistema operativo Windows haremos:

```
Set ORACLE_SID = biblios
```

Si trabajamos con Unix o Linux haremos:

```
Export ORACLE_SID = biblios
```

El próximo paso será iniciar SQL\*Plus “/ as sysdba” y establecer la base de datos en el modo “nomount”.

Ahora, si queremos ignorar el fichero pfile por defecto y comenzar la nueva instancia en el modo “nomount” deberemos escribir:

```
Startup nomount pfile = 'TFC\initbiblios.ora'
```

Ahora ya podemos crear la base de datos. Es recomendable que cuando vayamos a crear una base de datos escribamos el código en un fichero SQL, de esta forma tendremos una copia del comando y los parámetros utilizados en la creación así como un modo rápido de corregir posibles errores a la hora de usar el comando “create database”. El código que podríamos usar sería:

```
CREATE DATABASE biblios
Maxinstances 1
Maxloghistory 1
Maxlogfiles 10
Maxlogmembers 25
Maxdatafiles 100
Datafile 'TFC\biblios_01.dbf'
Size 300M reuse autoextend on next 15M maxsize unlimited
logfile
group 1 ('TFC\logs\redo_01.log') size 75M
group 2 ('TFC\logs\redo_02.log') size 75M
group 3 ('TFC\logs\redo_03.log') size 75M
default temporary tablespace TEMP
tempfile 'TFC\temp\temp_01.dbf'
extent management local uniform size 1M
undo tablespace UNDO_TS datafile 'TFC\undo\biblios_01.dbf'
size 125M reuse autoextend on next 15M maxsize unlimited;
```

Oracle deberá responder diciendo que la base de datos ha sido creada. Una vez hecho esto estamos preparados para cargar los scripts que son necesarios para dar soporte al producto Oracle , para ello deberemos ejecutar:

```
@'TFC\catalog.sql'
@'TFC\catproc.sql'
@'TFC\catexp.sql'
```

Realizado todo este proceso podemos decir que la base de datos está configurada. El primer trabajo que se deberá realizar es cambiar las contraseñas de *SYSTEM* y *SYS* para una mayor seguridad de nuestra base de datos.

Queda claro el método manual para crear la base de datos, aunque en este caso recurriremos a las herramientas que nos proporciona Oracle para crear nuestra base de datos.

A continuación debe crearse el espacio para las tablas (*TABLESPACE*), componente que no pertenece al diseño lógico de la base de datos y, por lo tanto, no está incluido en el estándar SQL. También tenemos dos posibilidades a la hora de crear el espacio para las tablas podemos usar el asistente de configuración de bases de datos que proporciona Oracle o mediante la ejecución de un Script. Si sumamos el tamaño inicial de todas las tablas obtenemos que estas ocuparán unos 88 Mb, así pues, de forma genérica el comando que nos va a permitir crear el espacio para las tablas (*TABLESPACE*) podría ser:

```
create tablespace bib datafile
'/ tfc/data/BIBLIOS/bib01.dbf' size 200M
default storage (
    initial 20M
    next 20M
    pctincrease 0
    minextents 2
);
```

De igual forma que en el caso anterior utilizaremos las herramientas que nos ofrece Oracle para la creación del espacio de tablas.

En la siguiente dirección podemos encontrar los pasos a seguir para crear tanto la base de datos como el espacio de tablas manualmente.

<http://www.cs.umb.edu/cs634/ora9idocs/server.920/a96521/create.htm>,

Una vez llegado a este punto seguiremos con la creación de las diferentes tablas necesarias para el funcionamiento de nuestra base de datos, aunque podemos utilizar la consola de Oracle para crear las tablas en este caso usaremos un Script para ello que puede verse completo en el Anexo II de esta memoria. El esquema de Script de creación de tablas es el siguiente:

1. Creación de los contadores necesarios mediante el comando *CREATE SEQUENCE*.
2. Creación de las tablas mediante *CREATE TABLE*.
3. Creación de las claves primarias (*PRIMARY KEY*).
4. Creación de las claves foráneas (*FOREIGN KEY*).
5. Creación de los índices de texto (*CREATE INDEX*).
6. Creación de las restricciones de las tablas (*CHECK*).

Podemos añadir un sexto punto con la creación, si vemos la necesidad, de *VIEWS* para la realización de posteriores consultas o bien realizar esto en un Script diferente.

Como ejemplo podemos ver la manera de proceder a la hora de crear la tabla *Empresas*:

```
/*
*Creación de las tablas
*/

CREATE TABLE EMPRESAS
(NIF CHAR(9) NOT NULL
,SECTOR CHAR(1) NOT NULL
,NOMBRE VARCHAR(30) NOT NULL
,TELEFONO VARCHAR(15) NOT NULL
,E_MAIL VARCHAR(25)
,DIRECCION VARCHAR(80) NOT NULL
```

```
)
PCTUSED 95
PCTFREE 5
TABLESPACE BIB
STORAGE(
INITIAL 1024
NEXT 100
PCTINCREASE 0
MINEXTENTS 1
MAXEXTENS 200);
```

.....

/\*

*\*Creación de las claves primarias*

\*/

**ALTER TABLE EMPRESAS**

ADD CONSTRAINT EMPRESAS\_PK PRIMARY KEY(NIF);

/\*

*\*Creación de restricciones de tabla*

\*/

**ALTER TABLE EMPRESAS**

ADD CONSTRAINT CK\_SECTOR CHECK(SECTOR IN('E','A','C','M'));

De igual manera procederemos con el resto de las tablas que debemos crear.

### 6.3. Operaciones de gestión básicas

Dentro de las operaciones básicas nos encontramos con todas las relacionadas con el mantenimiento de los datos como son añadir, modificar o eliminar datos de las tablas.

Así pues, una vez ejecutado el Script de creación de las tablas de nuestra base de datos nos queda la faena de introducir datos y mantenerlos. Veremos a continuación cómo y cuáles de estas operaciones podemos realizar o no en las diferentes tablas.

Ante todo debemos tener en cuenta que nos encontramos ante una base de datos donde el borrado de registros en las tablas va a estar muy restringido, solo podrá realizarse en algunas tablas. Esto es así por que debemos mantener la integridad de los datos.

Dicho lo anterior, optamos por impedir que un usuario cualquiera pueda hacer operaciones indebidas sobre nuestra base de datos mediante el uso de disparadores y que cualquier operación ha realizar deba ser realizada a través de procedimientos.

Para ellos usamos “triggers” o disparadores y la integridad referencial que hemos establecido a la hora de crear las tablas.

Un ejemplo rápido, si no bloqueáramos la posibilidad de borrado sobre la tabla de Usuarios (que sabemos está estrechamente relacionada con la de Personas) podría darse el caso de que alguien borrara un Usuario y quedara un enlace roto en la tabla Personas. Para evitar esto creamos un disparador que evite el borrado directo de registros de la tabla Usuarios, es más, por la filosofía empleada en el diseño nunca podremos borrar un Usuario.

Dicho esto, en la mayor parte de las tablas he optado por controlar mediante disparadores la inserción (*INSERT*), modificación (*UPDATE*) y borrado (*DELETE*) de registros de forma que no se pueden hacer directamente mediante sentencias SQL y es necesaria la ejecución de algún procedimiento, En el caso de las tablas Recursos, Nuevos\_recursos, Temas, Subtemas y Personas controlamos mediante disparadores únicamente la inserción, permitimos modificar y el borrado queda controlado por la integridad relacional.

De la tabla Empresas solo nos interesan aquellos registros que estén relacionadas con algún recurso, estos no podrán borrarse ya que nos lo impedirá la integridad relacional que hay establecida. Pero la modificación y la adición de registros esta abierta.

Los disparadores creados se encuentran recogidos en el Anexo III de la presente memoria.

Para finalizar en la siguiente tabla podemos ver que restringen los disparadores sobre las diferentes tablas de nuestra base de datos:

Tabla	Restringido		
	Update	Delete	Insert
Personas			X
Empleados	X	X	X
Usuarios	X	X	X
Sectores	X	X	X
Temas			X
Subtemas			X
Recursos			X
Documentos	X	X	X
Mapas	X	X	X
Libros	X	X	X
Videos	X	X	X
Música	X	X	X
Empresas			
Nuevos_Recursos			X
Nuevos_Documentos	X	X	X
Nuevos_Mapas	X	X	X
Nuevos_Libros	X	X	X
Nuevos_Videos	X	X	X
Nuevos_Música	X	X	X
Lista_Espera	X	X	X
Copias			X
Prestables	X	X	X
Retiradas	X	X	X
Referencia	X	X	X
Prestamos	X	X	X
Histórico Prestamos	X	X	X
Solicitados	X	X	X
Lista_Espera	X	X	X

Si nos fijamos en la lista de tablas veremos que hay una que no aparece en el esquema inicial, se trata de la tabla Histórico\_Prestamos, esta tabla tiene la misma estructura que la tabla Préstamos y en ella volcaremos, bien de forma automática o mediante una llamada explicita a un procedimiento, aquellos registros de la tabla Préstamos que a una fecha dada los recursos que referencian ya hayan sido devueltos. El tratamiento del histórico lo veremos en el punto siguiente.

Otra tabla que recibe un tratamiento especial en lo que a disparadores se refiere es la de usuarios. En el enunciado se indica que cuando los días de retraso acumulados por un usuario superen los seis meses debemos darlo de baja inmediatamente. Para este caso he creado un disparador que controla la actualización del campo Dias\_Penalización y cuando supera los 180 días llama al procedimiento que cambia el estado del usuario de alta a baja.

#### 6.4. Packages (Procedimientos, funciones y consultas)

En este apartado trataremos la gestión diaria de nuestra base de datos. Todo lo que debemos hacer estará recogido en procedimientos y funciones que se encuentran agrupadas en paquetes siguiendo, o intentado seguir, un criterio de dependencia entre las tablas.

A la hora de realizar el trabajo he partido de la suposición de que el programa que manejen los usuarios de este aplicativo estará desarrollado en un lenguaje de programación, como puede ser Java o PHP, y que será desde las diferentes clases (si trabajamos en Java) desde donde se llamará a los procedimientos pasando los parámetros necesarios para obtener los resultados deseados.

Como puede verse en el código las funciones definidas siempre son llamadas desde los procedimientos, en su mayoría sirven de comprobación de la existencia de algún recurso o registro en alguna tabla de la base de datos.

La parte correspondiente a las consultas pedidas y a la información estadística esta recogida toda en dos paquetes (consultas y estadística) y las operaciones que no sean las triviales sobre las tablas están recogidas en un paquete denominado gestión. El resto de los paquetes se centran el mantenimiento de las tablas principales y están agrupadas en seis paquetes.

A continuación veremos todos los paquetes creados y una breve explicación de su contenido. El código de los mismos se puede consultar en el Anexo IV de esta memoria.

#### **6.4.1 Package Sectores**

Paquete que gestiona la tabla sectores en la que tenemos las diferentes ubicaciones de nuestra biblioteca.

##### **6.4.1.1. PROCEDURE INS\_SECTOR**

Nos permite añadir nuevas sedes a nuestra biblioteca, deberemos pasar como parámetros el edificio, la planta y el sector.

#### **6.4.2. Package Temas**

Paquete que gestiona la tabla temas.

##### **6.4.2.1. FUNCTION COMPROBAR\_TEMAS**

Comprueba si un tema ya ha sido dado de alta.

##### **6.4.2.2. PROCEDURE INS\_TEMA**

Inserta un tema nuevo en la base de datos, previamente ejecuta la función *Comprobar\_Temas* para asegurarse de que no haya sido dado de alta previamente.

#### **6.4.3. Package Subtemas**

Paquete que gestiona la tabla subtemas.

##### **6.4.3.1. FUNCTION COMPROBAR\_SUBTEMAS**

Comprueba si un subtema ya ha sido dado de alta.

##### **6.4.3.2. FUNCTION COMPROBAR\_NUMERO**

Comprueba y devuelve el último número de subtema asignado dentro de un tema determinado.

##### **6.4.3.3. PROCEDURE INS\_SUBTEMA**

Inserta un subtema nuevo en la base de datos, previamente ejecuta la función *Comprobar\_Subtemas* para asegurarse de que no haya sido dado de alta previamente, si no existe ejecuta la función *Comprobar\_Numero* y le suma uno para añadirlo al campo *ID\_SUBTEMA*.

#### **6.4.4. Package Personas**

Paquete que gestiona la tabla personas y sus dependientes, empleados y usuarios.

##### **6.4.4.1. PROCEDURE DEL\_EMPLEADO**

Nos permite dar de baja un empleado. Implica borrar la fila de la tabla *Empleados* y *Personas*.

#### **6.4.4.2. PROCEDURE I\_EMPLEADO**

Añade un nuevo empleado a la base de datos, esta operación implica añadir el registro correspondiente a la tabla *Personas* además de a la tabla *Empleados*.

#### **6.4.4.3. PROCEDURE I\_USUARIO**

Como en el caso anterior nos permite añadir un usuario a la base de datos, implica igualmente añadir la fila a las tablas *Usuarios* y *Personas*. A diferencia del caso anterior jamás podremos borrar un usuario añadido.

#### **6.4.4.4. PROCEDURE MOD\_UBICA\_EM**

Nos permite cambiar el lugar de trabajo de un empleado. La nueva ubicación debe existir en la base de datos, en caso contrario dará un error.

#### **6.4.4.5. PROCEDURE MOD\_ESTADO\_U**

Aunque el cambio de estado de un usuario se realice automáticamente cuando supera los 180 días de penalización de alta a baja, puede darse el caso de que el cambio se haya realizado por error, mediante este procedimiento podríamos subsanarlo. Del mismo modo puede servirnos para dar de baja a un usuario que así lo desee.

#### **6.4.4.6. PROCEDURE MOD\_D\_PENA\_U**

Nos permitirá acceder a los días de penalización para actualizarlos. Igual que en el caso anterior la penalización se gestiona automáticamente. Este procedimiento nos permitirá actualizar la "pena" hasta liquidarla y poder obtener recursos.

### **6.4.5. Package Recursos**

Paquete para la gestión de la tabla recursos y sus dependientes, así como la de copias y prestables (entiendo que todo recurso nuevo dado de alta pasa a estar disponible en el catálogo de la biblioteca).

#### **6.4.5.1. FUNCTION COMPROBAR\_RECURSO**

Comprueba si un recurso ya ha sido dado de alta.

#### **6.4.5.2. FUNCTION COMPROBAR\_COPIAS**

Comprueba el número de copias que existen de un recurso.

#### **6.4.5.3. PROCEDURE INS\_DOCUMENTO**

Nos permite insertar un nuevo recurso de tipo documento en la base de datos, previamente ejecuta la función *Comprobar\_Recurso* para ver si existe, en caso de que no exista lo añadirá a las tablas *Recursos*, *Documentos*, *Copias* y *Prestables*. En el caso de que ya exista solo lo añadirá a la tabla de copias y prestables. Para obtener el número de copia que le corresponde ejecuta la función *Comprobar\_Copias* y le suma uno.

#### **6.4.5.4. PROCEDURE INS\_MAPAS**

El funcionamiento es el mismo que en el caso anterior pero este caso con recursos de tipo mapas.

#### **6.4.5.5. PROCEDURE INS\_LIBRO**

El funcionamiento es el mismo que en el apartado 6.4.5.3. pero este caso con recursos de tipo libro.

#### **6.4.5.6. PROCEDURE INS\_VIDEO**

El funcionamiento es el mismo que en el apartado 6.4.5.3. pero este caso con recursos de tipo video.

#### **6.4.5.7. PROCEDURE INS\_MUSICA**

El funcionamiento es el mismo que en el apartado 6.4.5.3. pero este caso con recursos de tipo música.

#### **6.4.5.8. PROCEDURE INS\_COPIA**

Añade una nueva copia de un recurso existente a la tabla copias y prestables. Ejecuta la función *Comprobar\_Copias* para obtener el número de la última copia y le suma uno.

#### **6.4.6. Package Nuevos Recursos**

Paquete que gestiona las tablas relacionadas con los nuevos recursos solicitados.

##### **6.4.6.1. FUNCTION COMPROBAR\_RECURSO**

Comprueba si un recurso ya ha sido dado de alta como nuevo recurso.

##### **6.4.6.2. PROCEDURE INS\_DOCUMENTO**

Nos permite insertar un recurso de tipo documento en la base de datos, previamente ejecuta la función *Comprobar\_Recurso* para ver si existe, en caso de que no exista lo añadirá a las tablas *Nuevos\_Recursos*.

##### **6.4.6.3. PROCEDURE INS\_MAPAS**

El funcionamiento es el mismo que en el apartado 6.4.6.2. pero este caso con recursos de tipo mapas.

##### **6.4.6.4. PROCEDURE INS\_LIBRO**

El funcionamiento es el mismo que en el apartado 6.4.6.2. pero este caso con recursos de tipo libro.

##### **6.4.6.5. PROCEDURE INS\_VIDEO**

El funcionamiento es el mismo que en el apartado 6.4.6.2. pero este caso con recursos de tipo video.

##### **6.4.6.6. PROCEDURE INS\_MUSICA**

El funcionamiento es el mismo que en el apartado 6.4.6.2. pero este caso con recursos de tipo música.

#### **6.4.7. Package Gestión**

Este paquete puede ser considerado el corazón de todo el sistema, en él están recogidos todos los procedimientos y funciones que van a posibilitar el funcionamiento diario de nuestra biblioteca, veamos cuales son y una pequeña descripción de lo que hacen.

##### **6.4.7.1. FUNCTION COMPROBAR\_LISTA**

Comprueba si un recurso esta en la lista de solicitados (tabla *Lista\_Espera*) y devuelve en su caso el *ID* del usuario solicitante

##### **6.4.7.2. FUNCTION COMPROBAR\_TIEMPO\_MEDIO**

Comprueba el tiempo medio que un recurso esta en la lista de solicitados (tabla *Lista\_Espera*) y devuelve su *ID* si el tiempo es mayor de 30 días y el recurso es de tipo libro.

##### **6.4.7.3. FUNCTION OBTENER\_FP**

Obtiene y devuelve la fecha en la que se ha realizado un préstamo

##### **6.4.7.4. FUNCTION OBTENER\_FUD**

Obtiene y devuelve la fecha en la que se ha realizado el último préstamo un usuario.

##### **6.4.7.5. FUNCTION COMPROBAR\_PRESTAMOS**

Comprueba si un recurso esta en la tabla *Prestamos* devolviéndonos si es el caso el *ID* del recurso.

##### **6.4.7.6. FUNCTION COMPROBAR\_RETIRADA**

Comprueba si un recurso esta en la tabla *Retiradas* devolviéndonos si es el caso el *ID* del recurso.

##### **6.4.7.7. FUNCTION OBTENER\_PR**

Para un recurso determinado comprueba si puede ser prestada o consultada en sala..

#### **6.4.7.8. FUNCTION OBTENER\_DP**

Para un recurso determinado obtiene y devuelve los días que puede permanecer en préstamo.

#### **6.4.7.9. FUNCTION OBTENER\_ESTADO**

Para un usuario determinado obtiene y devuelve su estado , alta o baja

#### **6.4.7.10. FUNCTION OBTENER\_PENALIZACION**

Para un usuario determinado obtiene y devuelve los días de penalización que tiene.

#### **6.4.7.11. PROCEDURE INS\_LISTA**

Inserta un recurso solicitado por un usuario y que no se encuentra disponible en ese momento en la tabla *Lista\_Espera*.

#### **6.4.7.12. PROCEDURE TIEMPO\_MEDIO**

Inserta una petición de libro en las tablas *Nuevos\_Recursos* y *Nuevos\_libros* si el tiempo medio de espera en lista (tabla *Lista\_Espera*) es mayor de 30 días

#### **6.4.7.13. PROCEDURE INS\_PRESTAMO**

Inserta un recurso prestado a un usuario en la tabla de prestamos

#### **6.4.7.14. PROCEDURE DEV\_PRESTAMO**

Gestiona la devolución de los recursos prestados, comprueba si se ha producido retraso en la devolución para penalizar al usuario y, además, comprueba si ha sido solicitado por algún otro usuario y nos avisa de ello. Y finalmente, si estaba en lista de espera, comprueba el tiempo de espera medio para incluirlo en las nuevas peticiones de recursos.

#### **6.4.7.15. PROCEDURE HIS\_PRESTAMO**

Pasa de la tabla de prestamos a la tabla histórica de prestamos (*H\_Prestamos*), los recursos que pasen deben haber sido devueltos.

#### **6.4.7.16. PROCEDURE UPD\_COPIAS**

Actualiza el estado en el que se encuentra una copia, dependiendo del nuevo estado la copia deberá estar en prestables o retiradas. No podremos cambiar el estado de una copia que se encuentre en préstamo.

### **6.4.8. Package Consultas**

Este paquete recoge todas las consultas que se piden en el enunciado del presente TFC.

#### **6.4.8.1. PROCEDURE CONS\_1**

Nos permite consultar recursos por uno de los siguientes ítems o la combinación de varios: tema, subtema, tipo de recurso, palabras clave, título, autor, país de origen.

#### **6.4.8.2. PROCEDURE CONS\_2**

Nos permite obtener una lista de recursos más solicitados, en general, por tema, por palabras clave y por tipo de recurso.

#### **6.4.8.3. PROCEDURE CONS\_3**

Nos permite obtener una lista de socios con retrasos de devolución en el momento en el que se ejecute la consulta.

#### **6.4.8.4. PROCEDURE CONS\_4**

Nos permite consultar la disponibilidad de un recurso y, además, saber si está disponible, en caso de que no lo estuviera indica la fecha en la que lo estará.

#### **6.4.8.5. PROCEDURE CONS\_5**

Nos permite consultar la referencia bibliográfica (libros) de un recurso así como la disponibilidad de éstas en la biblioteca.

#### **6.4.8.6. PROCEDURE CONS\_6**

Obtenemos una lista con los usuarios que están a la espera de un determinado recurso, ordenada por fecha de solicitud.

#### **6.4.8.7. PROCEDURE CONS\_7**

Obtenemos un listado con la información de los empleados que trabajan en un edificio, planta y sector determinados.

#### **6.4.9. Package Estadística**

A la hora de enfocar este punto he tenido en cuenta la experiencia personal, vuelvo a basarme en la Biblioteca Municipal de Mislata (Valencia). Entiendo que la biblioteca deberá sacar una memoria anual de gestión, por lo tanto las estadísticas se ejecutarán de forma anual principalmente, aunque esto no quita que alguna pueda ejecutarse de forma esporádica antes de la realización de la memoria anual.

##### **6.4.9.1. PROCEDURE EST\_1**

Dado un recurso concreto, obtenemos un listado con los recursos que solicitaron otros socios que también solicitaron el primer recurso, ordenados en forma decreciente por número de solicitudes.

##### **6.4.9.2. PROCEDURE EST\_2**

Lo mismo que antes pero en lugar de ser un recurso concreto, puede ser un tema, título, autor.

##### **6.4.9.3. PROCEDURE EST\_3**

Obtenemos un listado dividido en las siguientes franjas de edad: [0..12], [13..18], [19..25], [26..35], [36..45], [46..55], [56..]. Por cada franja listamos los 100 recursos más solicitados según uno o la combinación de los criterios: tipo de recurso, tema, subtema, palabras clave, título, autor, país de origen.

##### **6.4.9.4. PROCEDURE EST\_4**

Agrupamos los préstamos por meses y los sumamos para obtener los meses en los que se solicitan más recursos según: tipo de recurso, tema, subtema, palabras clave, título, autor.

##### **6.4.9.5. PROCEDURE EST\_5**

Dado un recurso, obtenemos un listado con los recursos lo referencian.

##### **6.4.9.6. PROCEDURE EST\_6**

El resultado es obtener un listado con aquellos recursos que han tenido lista de espera mayor a dos durante el último año, además calcula para cada recurso el tiempo medio de espera por el mismo.

## 7. Valoración de costes del proyecto

La valoración económica del proyecto, a parte de software sobre el que hemos trabajado, también deberá tener en cuenta el hardware sobre el cual estará corriendo dicho software. A modo de ejemplo a continuación puede verse un ejemplo de lo que puede ser el servidor de base de datos Oracle para un correcto funcionamiento de todos los componentes informáticos que se han propuesto en este estudio.

A continuación se valorarán en sus respectivos apartados de manera económica el software y el hardware.

### 7.1. Valoración del Hardware

Un equipo con las siguientes características podría servir como servidor de Base de datos Oracle (fuente: www.hp.es):

#### FICHA TECNICA:

##### HP ProLiant ML570 G2

- Bastidor
- 2 x Xeon MP 2.7 GHz
- RAM 1 GB
- HD: ninguno/a
- CD
- LAN EN, Fast EN
- Monitor : ninguno
- 7 U



El servidor HP ProLiant ML570 G2 es un producto concebido para grandes corporaciones y clientes empresariales, que requieren un servidor escalable para soportar entornos de empresa en crecimiento, ya que ofrece una capacidad extremadamente alta de almacenamiento para disponer de acceso rápido a datos centralizados y una elevada disponibilidad para proporcionar el máximo tiempo de funcionamiento. Este sistema proporciona a las empresas protección de su inversión y soporte para crecimiento futuro, una fácil implantación y gestión remota para soportar a centros de datos distribuidos y un gran número de usuarios. El HP ProLiant ML570 G2 es un servidor adecuado para aplicaciones críticas a un coste razonable.

#### ESPECIFICACIONES PRINCIPALES:

**Garantía del fabricante:** 3 años de garantía

**Dimensiones (Ancho x Profundidad x Altura):** 44.4 cm x 65 cm x 32.5 cm

**Procesador:** 2 x Intel Xeon MP 2.7 GHz

**Disco duro:** Ninguno/a

**Conexión de redes:** Adaptador de red - PCI - Ethernet, Fast Ethernet

**Alimentación:** CA 110/230 V ( 50/60 Hz )

**Memoria RAM:** 1 GB (instalados) / 32 GB (máx.) - DDR SDRAM - ECC - 200 MHz - PC1600

**Tipo:** Servidor

**Localización:** Europa

**Almacenamiento óptico:** CD-ROM

**Monitor:** Ninguno

**Caché por procesador:** 2 MB

**Controlador gráfico:** PCI - ATI RAGE IIC - 8 MB

**Unidad de disquete:** Disquete de 3,5" de 1,44 MB

**Factor de forma:** Montaje en bastidor - 7 U

**Memoria caché:** 4 MB L3

**Descripción del producto:** HP ProLiant ML570 G2 - Xeon MP 2.7 GHz

**Controlador de almacenamiento:** SCSI ( Ultra160 SCSI ) - PCI 64

#### ESPECIFICACIONES AMPLIADAS:

##### General

Tipo: Servidor

Anchura: 44.4 cm  
Profundidad: 65 cm  
Altura: 32.5 cm  
Localización: Europa  
Uso recomendado: Empresa  
Factor de forma del producto: Montaje en bastidor - 7 U  
Cantidad de compartimentos frontales: 4  
Cantidad de compartimentos de intercambio rápido (hot-swap): 12

### Expansión / Conectividad

Interfaces:

- 1 x paralelo - IEEE 1284 (EPP/ECP) - D-Sub de 25 espigas (DB-25) !
- 2 x serial - RS-232 - D-Sub de 9 espigas (DB-9)
- 1 x ratón -genérico - mini-DIN de 6 espigas (estilo PS/2)
- 1 x teclado -genérico - mini-DIN de 6 espigas (estilo PS/2)
- 1 x pantalla / vídeo -VGA - HD D-Sub de 15 espigas (HD-15)
- 2 x USB - 4 PIN USB tipo A
- 1 x red - Ethernet 10Base-T/100Base-TX - RJ-45
- 1 x almacenamiento - Ultra160 SCSI - 68 PIN VHDCI (Mini-Centronics)

Total ranuras de expansión (libres):

- 4 ( 2 ) x procesador - Enchufe 603
- 8 ( 6 ) x memoria - DIMM de 184 espigas
- 4 ( 4 ) x PCI-X hot-plug a 100 MHz
- 3 ( 3 ) x PCI-X / 100 MHz
- 2 ( 1 ) x placa memoria

Total compartimentos de expansión (libres):

- 12 ( 12 ) x hot-swap - 3.5" x 1/3H
- 1 ( 0 ) x accesible a la parte frontal - 3.5"
- 3 ( 2 ) x accesible a la parte frontal - 5.25" x 1/2H

### Garantía del fabricante

Servicio y mantenimiento: 3 años de garantía

Detalles de Servicio y Mantenimiento: Garantía limitada - piezas y mano de obra - in situ - 3 años

### Diverso

Características: Contraseña de administrador, contraseña de encendido, contraseña teclado, bloqueo configuración disco, control E/S disquete, control E/S puerto paralelo, control de E/S del puerto serie, bloqueo rápido

Cumplimiento de normas: ACPI 2.0

### Procesador

Tipo: Intel Xeon MP 2.7 GHz

Tipo conjunto de chips: ServerWorks Grand Champion HE

Cantidad instalada: 2

Velocidad bus de datos: 400 MHz

Cantidad máxima soportada: 4

Capacidad de actualización: Actualizable

### Memoria RAM

Tecnología: DDR SDRAM - ECC

Factor de forma: DIMM de 184 patillas

Características: Registrado

Tamaño instalado: 1 GB / 32 GB (máx.)

Norma de actualización: 2 módulos a la vez

Conforme a la especificación de memoria PC1600

Velocidad de memoria: 200 MHz

### Almacenamiento

Disco duro: Ninguno/a

Unidad de disquete: Disquete de 3,5" de 1,44 MB

### Conexión de redes

Características: Wake on LAN (WoL)

Conexión de redes: Adaptador de red - PCI - integrado

Protocolo de interconexión de datos: Ethernet, Fast Ethernet

Cumplimiento de normas: IEEE 802.3, IEEE 802.3U

### Alimentación

Voltaje necesario: CA 110/230 V ( 50/60 Hz )

Tipo de dispositivo: Fuente de alimentación - conectable en caliente / redundante

Cantidad instalada: 3 (instalados) / 3 (máx.)

Potencia suministrada: 600 vatios

### Controlador de almacenamiento

Tipo: 1 x SCSI - integrado - PCI 64

Tipo de controlador interfaz: Ultra160 SCSI

Nº canales: 2

### Memoria caché

Tipo: L3

Tamaño instalado: 4 MB

Caché por procesador: 2 MB

### Almacenamiento óptico

Tipo: CD-ROM - IDE

Velocidad de lectura: 48x

### Sistemas operativos / Software

Software incluido:

- Controladores y utilidades
- Compaq SmartStart
- Compaq Management Agents
- Compaq Insight Manager 7
- Compaq Survey Utility
- Compaq ROMPaq

OS certificado:

- SCO OpenServer
- UnixWare, SunSoft Solaris x86
- Novell NetWare
- Microsoft Windows 2000 Server
- Microsoft Windows 2000 Advanced Server
- Red Hat Linux
- IBM OS/2 Warp Server para negocios electrónicos
- Caldera OpenLinux
- SuSE Linux Enterprise Server
- Microsoft Windows NT Server Enterprise Edition
- Microsoft Windows Server 2003 Enterprise Edition
- Microsoft Windows Server 2003 Standard Edition
- Red Hat Enterprise Linux

### Monitor

Tipo de monitor: Ninguno

### Controlador de almacenamiento (2º)

Tipo: 1 x IDE - integrado

Tipo de controlador interfaz: IDE/ATA

### Controlador gráfico

Tipo: PCI - integrado

Memoria de vídeo: SDRAM

Tamaño instalado: 8 MB

Resolución máxima (externa): 1600 x 1200 / 16 bits (64K colores)

Procesador gráfico/ fabricante: ATI RAGE IIC

La valoración económica de este servidor es aproximadamente de 7.800 €

El servidor no incluye discos duros para poder almacenar datos, con lo que se propone la compra de discos duros de tipo SCSI para poder formar un RAID y tener la seguridad de que el fallo físico de un disco no provoca la pérdida de información.

Las características de los discos duros será la siguiente:

#### FICHA TÉCNICA:

HP - Disco duro - 72.8 GB - hot-swap - 3.5" - Ultra320 SCSI - 10000 rpm



El disco duro que usted elija determinará en gran medida el rendimiento y la confiabilidad general de su sistema. El disco duro es un componente fundamental de su sistema de almacenamiento primario, que contiene toda la información, desde sus datos de importancia crítica hasta el sistema operativo del que depende su equipo.

Los discos Compaq se someten a pruebas y procesos de calificación para garantizar el mayor nivel de confiabilidad, compatibilidad e interoperabilidad. ¿Por qué confiar sus recursos vitales a unidades que no sean discos duros Compaq? Esta unidad de disco duro se ha diseñado para servidores ProLiant, sistemas de almacenaje ProLiant, AlphaServers, gabinetes HP StorageWorks, con capacidad de Ultra 320, Ultra3 o Ultra2.

#### ESPECIFICACIONES PRINCIPALES:

**Descripción del producto:** HP disco duro - 72.8 GB - Ultra320 SCSI

**Factor de forma:** 3.5" x 1/3H

**Tipo:** Disco duro - hot-swap

**Velocidad de transferencia de datos:** 320 MBps

**Tipo de interfaz:** Ultra320 SCSI DBV

**Capacidad:** 72.8 GB

**Tiempo de búsqueda medio:** 4.9 ms

**Velocidad del eje:** 10000 rpm

**Dimensiones (Ancho x Profundidad x Altura)** 10.2 cm 2.5 cm

#### ESPECIFICACIONES AMPLIADAS:

##### General

Anchura: 10.2 cm

Altura: 2.5 cm

Compatibilidad: PC

Tipo de dispositivo: Disco duro - hot-swap

##### Expansión / Conectividad

Interfaces: 1 x Ultra320 SCSI

Compartimentos compatibles: 1 x hot-swap - 3.5" x 1/3H

##### Diverso

Accesorios incluidos: Bandeja de la unidad de disco duro

##### Parámetros de entorno

Temperatura mínima de funcionamiento: 10 °C

Temperatura máxima de funcionamiento: 35 °C

#### **Disco duro**

Factor de forma: 3.5" x 1/3H

Tipo de interfaz: Ultra320 SCSI

Homologado S.M.A.R.T.

Capacidad: 72.8 GB

Tipo de señalización: SCSI Diferencial de bajo voltaje (DBV)

#### **Prestación**

Índice de transferencia de la unidad: 320 MBps (externo)

Tiempo de búsqueda: 4.9 ms (media) / 10 ms (máx.)

Velocidad del eje: 10000 rpm

El precio de estos discos duros está aproximadamente 310 € por unidad. Si queremos tener un raid 5 con replicación de datos deberemos usar como mínimo 7 discos, por lo que deberemos gastar 2.170 €.

En total deberemos invertir casi 11.000 € en la compra del hardware.

## **7.2. Valoración del software**

En la valoración del software no solo deberemos tener en cuenta el importe de las licencias de Oracle que vayamos a necesitar, además deberemos tener en cuenta el software desarrollado para el funcionamiento de la aplicación.

En lo referente al Oracle, adquiriremos la versión 9i Enterprise Edition y, por seguir suponiendo, vamos a necesitar 20 licencias para nuestra biblioteca. Basándonos en los precios obtenidos de la página web de Oracle podemos decir que el importe de la adquisición rondará los 7.500 €.

La parte referente al desarrollo es la más difícil de valorar, va a depender y mucho de que empresa de software desarrolle el proyecto. En mi caso cuento con el precio de hora de la empresa TAO, perteneciente al grupo GEDAS, en este caso la empresa cobra a 104 € la hora de trabajo por persona implicada, independientemente de que sea analista o programador. Supongamos que para la realización del proyecto necesitamos un analista y 4 programadores, supongamos el analista necesita 36 horas para realizar el estudio y redactar el proyecto y los programadores trabajan durante 20 días a 7 horas por día, esto es 140 horas, sumamos las dos cantidades y tenemos un total de 176 horas, redondeamos a 200 horas y multiplicamos, el total asciende a 20.800 €.

El importe puede parecer una exageración, pero pensemos que la empresa puede desarrollar el software para que se pueda utilizar en más de una biblioteca, con lo cual el precio final de la aplicación puede ser un tercio de la cantidad calculada, es decir, unos 7.000 €. Con 3 instalaciones ya ha recuperado la inversión hecha, el resto serán ganancias.

Sería de locos, a mi entender, el intentar realizar todo el proyecto a medida y una sola persona. Siempre habrá algo que retocar, así que puede considerarse atado de por vida al desarrollo realizado.

## 8. Conclusiones

Llegado a este punto solo cabe ya hacer una reflexión sobre el trabajo que ha sido realizado en la presente memoria, no solo en el aspecto técnico o académico, sino además en el aspecto personal, esto es, todo lo referente a los nuevos conocimientos que me ha aportado el presente desarrollo.

He de decir que este ha sido mi primer contacto con la base de datos Oracle, hasta ahora había trabajado únicamente con Informix. Los cambios a nivel de lenguaje SQL no han sido tan grandes, en cambio todo lo referente a la creación de bases de datos así como el lenguaje de programación PL/SQL ha sido una experiencia totalmente nueva.

Puede ser, desde mi punto de vista, que el tener conocimientos básicos sobre Oracle y PL/SQL no sea suficiente para poder llevar a cabo un desarrollo óptimo del trabajo encomendado. No me cabe la menor duda de que no funciona todo a la primera y sin fallos, aunque los diferentes paquetes hayan sido compilados y corregidos de errores. No es algo nuevo en este trabajo que después del proceso de desarrollo llega un periodo de “tuning”, las famosas versiones beta, para dejar el aplicativo en condiciones óptimas para su funcionamiento.

Desde mi punto de vista falta mucho camino por recorrer y poder dar por terminada la aplicación, todo el trabajo de programación está pendiente y en la integración de lo desarrollado con el lenguaje de programación que se haya elegido para desarrollar la interfaz que manejará el usuario pueden surgir nuevos problemas i/o errores que hasta el momento no eran visibles o no se habían producido. Además creo que el proyecto puede verse enriquecido con la introducción de nuevos procesos que automaticen más la gestión de la biblioteca. También pienso que ya que Oracle nos da la posibilidad de acceder mediante web, cabría la posibilidad de dotar al sistema de esta posibilidad de forma que los usuarios pudieran realizar reservas de recursos disponibles desde su domicilio sin tener que desplazarse a ninguna de las sedes de la biblioteca. Estas dos nuevas ideas son solo una pequeña parte de lo que podemos ampliar nuestro sistema de gestión.

A nivel de diseño, aunque tal como está funciona correctamente, yo hubiera preferido trabajar con un nivel más detallado en las tablas, por ejemplo que el campo dirección utilizado en Personas y Empresas estuviera relacionado con otras tablas como podrían ser, códigos postales (que incluiría los diferentes códigos postales de las poblaciones y un enlace a las tablas poblaciones y provincias) esto por pover un ejemplo.

Personalmente he tenido que usar mucho la bibliografía correspondiente a Oracle, aunque no sé si habrán sido muchas más las consultas realizadas en INTERNET con el fin de conseguir ejemplos o documentación para poder seguir con el trabajo, el desarrollo del presente trabajo hubiera sido imposible de no haber contado con este medio. Es de agradecer la ayuda prestada por Gladys, consultora asignada como tutora a mi trabajo final de carrera, sin ella me hubiera quedado atascado más de una vez.

En definitiva, pienso que sería conveniente introducir algún seminario sobre Oracle, en modo curso de verano o especialización, de forma que podamos adquirir conocimientos con un buen fundamento.

Creo haber conseguido todos los puntos que se piden en el enunciado del TFC, los he probado todos dentro de mis posibilidades y los resultados obtenidos han sido los deseados. Hubiera sido muy interesante el poder introducir un número de datos mucho mayor de los que he manejado pero el tiempo corre y no puedes pararte en muchos detalles. A partir de este punto, en los Anexos, puede verse todo el trabajo realizado para la realización de este proyecto.



## 9. Bibliografía

- (1) [ORA01]: *Oracle9i: Guía de aprendizaje* (Oracle Press).
- (2) [ORA02]: *Oracle9i: Manual de referencia* (Oracle Press)
- (3) [ORA03]: *Oracle9i: Programación PL/SQL* (Oracle Press)
- (4) [ORA04]: *Introduction to Oracle9i: PL/SQL. Student guide.* ORACLE®
- (5) [ORA05]: *Oracle 9i. SQL\*PLUS. User's Guide and Reference. Release 9.2, March 2002.*
- (6) [ORA06]: *Oracle 9i. SQL Reference. Release 2 (9.2), October 2002*
- (7) [ORA07]: *Oracle 9i. Supplied PL/SQL Packages and Types Reference. Release 2 (9.2), March 2002.*
- (8) [BAR01]: *El modelo entidad relación. CASE\*METHOD.* Richard Barker.
- (9) [BAT01]: *Diseño conceptual de bases de datos. Un enfoque de entidades relaciones.* Batini y otros.
- (10)[TEO01]: *Database modelling and design.* Teoray T.J.
- (11)[ELM01]: *Fundamentals of Database Systems.* Redwood City. ELMASRI, N., y NAVATHE, S.B. (1989). The Benjamin/Cummings Pub.Co.
- (12)[BER01]: *Sistemas de bases de datos orientadas a objetos.* BERTINO, E. y MARTINO, L. (1995). Wilmington. Addison-Wesley/Díaz de Santos.
- (13)[SIS01]: *Disseny de bases de dades.* Jaume Sistac i Planas. EDIUOC.
- (14)[SIS02]: *Bases de datos I.* Jaume Sistac i Planas. UOC
- (15)[SIS03]: *Bases de datos II.* Jaume Sistac i Planas. UOC.

## Anexo I. Enunciado del TFC

### Titulo

Diseño e implementación de un sistema de gestión e inteligente para una biblioteca.

### Descripción

Se trata de realizar el diseño y la implementación de un sistema que gestione todos los recursos de una biblioteca, así como realice cálculos inteligentes para la toma de decisiones en lo que tiene que ver con la reposición y adquisición de nuevos recursos. Dicha biblioteca posee como recursos disponibles al público: documentos, mapas, videos, música y libros.

Para cada recurso se almacena la siguiente información: tipo de recurso, título, autor/es, edición, palabras clave, tema, subtema, número de copias, estado de la copia, disponibilidad de préstamo en sala o a domicilio, país de origen, duración máxima del préstamo si procede, ubicación física en la biblioteca. En caso de ser un documento se tiene información sobre el tipo de soporte (papel, digital). En caso de ser un libro se conoce además el número total de páginas, la editorial, la edición. En caso de ser video se conoce el tipo de soporte (cinta o DVD), así como información sobre el director, la productora. Si es música se conoce también el tipo de soporte: cinta, disco de vinilo, CD, así como información sobre la discográfica.

Algunos recursos cuentan con referencia bibliográfica.

Se dispone también de información sobre los socios de la biblioteca: nombre, apellido, teléfono de contacto, correo electrónico, dirección postal, histórico con los recursos utilizados de la biblioteca, así como su estado actual. Asimismo se acumulará por cada socio los eventuales retrasos generados a lo largo de su relación de socio con la biblioteca. Este número se explicará más adelante.

La biblioteca está distribuida en varios edificios. Cada edificio está organizado en plantas. Cada planta se divide en sectores que se corresponden a cada tema de los mencionados que posee cada recurso.

El sistema deberá permitir operaciones de gestión básicas como el ingreso y eliminación de material de la biblioteca, el alta, baja y modificación de los socios y la relación de préstamos entre socios y recursos, alta, baja y modificación de empleados.

Cuando un recurso no está disponible el socio tiene la posibilidad de apuntarse en una lista de espera.

Por cada día de retraso de devolución de un recurso por parte de un socio se le aplicará una penalización igual al retraso en días. Durante éste período no podrá obtener préstamos. Los retrasos contados en días se van acumulando por socio.

Se mantiene información de los libros que han sido solicitados y no estaban en catálogo para futuras compras. Asimismo se mantiene información de los libros en mal estado. Se los retira del catálogo de préstamos, hasta ser reparados o repuestos por otro nuevo.

Se pide que el sistema sea capaz de ofrecer información respecto a lo siguiente:

- Consultar recursos por uno de los siguientes ítems o la combinación de varios: tema, subtema, tipo de recurso, palabras clave, título, autor, país de origen.
- Consulta de lista de recursos más solicitados, en general, por tema, por palabras clave y por tipo de recurso.
- Consulta de lista de socios con retrasos de devolución en el momento actual.
- Consultar disponibilidad de un recurso. Indicar si está disponible y si no estuviera, decir cuando debiera estarlo
- Consultar la referencia bibliográfica de un recurso así como la disponibilidad de éstas en la biblioteca.
- Consultar lista de espera por un recurso.

- Si un socio tiene acumulada un retraso superior a 6 meses, será automáticamente dado de baja.
- Consultar información de los empleados de un edificio, planta, sector.

El sistema contará además con cierta información estadística que será de utilidad para la toma de decisiones relacionadas con la nueva adquisición, renovación y reubicación de recursos:

- Dado un recurso concreto, listar que otros recursos solicitaron otros socios que también solicitaron el primer recurso, ordenarlos en forma decreciente por número de solicitudes.
- Lo mismo que antes pero en lugar de ser un recurso concreto, puede ser un tema, título y autor.
- Indicar por franja de edades [0..12], [13..18], [19..25], [26..35], [36..45], [46..55], [56..] la consulta de los 100 recursos más solicitados según uno o la combinación de los criterios: tipo de recurso, tema, subtema, palabras clave, título, autor, país de origen.
- Indicar los períodos del año que se solicitan más recursos según: tipo de recurso, tema, subtema, palabras clave, título, autor.
- Dado un recurso, conocer que otros recursos lo referencian.
- Indicar que recurso ha tenido lista de espera mayor a dos durante el último año. Indicar también para cada recurso el tiempo medio de espera por el mismo.

Por último se pide automáticamente se generen peticiones de libros nuevos que se agregarán a los solicitados (no deberá haber repeticiones) cada vez que el sistema detecte que un libro tiene un tiempo medio de espera superior a 1 mes (no se contabilizan los retrasos).

#### Objetivos

- Poner en práctica los conocimientos adquiridos en las asignaturas de Bases de Datos I y Bases de Datos II
- Haber cursado las asignaturas Bases de Datos I y II
- Conocimientos mínimos de Oracle.
- Conocimientos de SQL y PL/SQL.

#### Requerimientos de software

Oracle 8i o superior.