

# **CONSTRUCCION Y EXPLOTACION DE UN ALMACEN DE DATOS PARA EL ANALISIS DEL SISTEMA DE PRESTACIONES SOCIALES**

## **MEMORIA FINAL**

### **TFC - ALMACENES DE DATOS**

**Carlos Javier Serrano Guarinos  
Curso 2011-2012  
2º Semestre  
Ingeniería Técnica Informática  
Consultor: Xavier Plaza Sierra**

## 1. DEDICATORIAS

Este proyecto se lo dedico a mi mujer, que tanta comprensión ha tenido cuando me quedaba horas y horas estudiando asignatura tras asignatura quitando tiempo de nuestra vida en común. Ella ha sido el soporte necesario para conseguir terminar este TFC y los estudios de ITIG.

También se lo dedico a mi hija que solo su presencia me ha servido de apoyo en los momentos más duros.

Finalmente se lo dedico a todos los que han aportado su granito de arena para conseguir todos los objetivos que me propuse al principio.

## 2. RESUMEN

El objeto de este documento es la de consolidar todos los conocimientos adquiridos durante la carrera de Ingeniería Técnica en Informática de Gestión y plasmarlo en un problema del mundo real.

También se requiere capacidad de asimilación, síntesis y resolución para lograr a partir de un problema del mundo real cumplir los objetivos que se proponen en el TFC.

A nivel personal el proyecto ha de servir para ver de forma global como usar los distintos conocimientos aprendidos en mis estudios y plasmarlo en la resolución de un proyecto complejo usando las herramientas que tengo a mi alcance.

El Organismo del Análisis del Departamento del Trabajo, en adelante OADT, necesita hacer un estudio dado el considerable aumento de personas desempleadas en los tiempos de crisis en el que estamos para ver si el sistema de prestaciones sociales es sostenible o por el contrario hay que acometer alguna acción.

El problema es que debido a la gestión distribuida de los servicios sociales por las CC.AA, además de por la obligación del cumplimiento de la protección de datos, no es posible acceder a la información que gestionan los departamentos de los servicios sociales de estos organismos.

De esta manera la OADT ha decidido encargarnos, como consultora externa al organismo, la creación de un almacén de datos para poder explotar información relevante a la necesidad del OADT.

## ÍNDICE

<b>1. DEDICATORIAS .....</b>	<b>2</b>
<b>2. RESUMEN .....</b>	<b>2</b>
<b>3. OBJETIVOS</b>	
3.1. OBJETIVOS DE LA ASIGNATURA .....	5
3.2. OBJETIVOS DEL PROYECTO .....	5
3.3. ENFOQUE Y METODO SEGUIDO .....	6
<b>4. PLANIFICACION</b>	
4.1. PLANIFICACION GLOBAL .....	7
4.2. PLANIFICACION ESPECÍFICA .....	7
4.3. DIAGRAMA DE GANNT .....	9
4.4. INCIDENCIAS Y RIESGOS .....	12
<b>5. RELACION DE PRODUCTOS OBTENIDOS .....</b>	<b>13</b>
<b>6. BREVE DESCRIPCION DEL RESTO DE APARTADOS .....</b>	<b>14</b>
<b>7. ANALISIS</b>	
7.1. DATOS DE ORIGEN .....	15
7.2. ANALISIS DE LOS DATOS DE ORIGEN .....	16
7.3. DIAGRAMA DE CASOS DE USO .....	17
7.4. DIAGRAMA DEL MODELO CONCEPTUAL .....	17
<b>8. DISEÑO</b>	
8.1. DIAGRAMA DE LA ARQUITECTURA DE HARDWARE .....	22
8.2. DIAGRAMA DE LA ARQUITECTURA DE SOFTWARE .....	22
8.3. DISEÑO DE LA BASE DE DATOS Y DIAGRAMA DEL MODELO FISICO .....	23
8.4. PROCESO ETL DE CARGA Y GENERACION DE LA BASE DE DATOS .....	29
<b>9. INFORMES: DISEÑO Y CAPTURAS DE PANTALLA .....</b>	<b>33</b>
<b>10. CONCLUSIONES .....</b>	<b>44</b>
<b>11. LINEAS DE EVOLUCION FUTURAS .....</b>	<b>44</b>
<b>12. GLOSARIO .....</b>	<b>45</b>
<b>13. BIBLIOGRAFIA .....</b>	<b>46</b>
<b>14. ANEXOS</b>	
14.1. ANEXO 1: SCRIPT DE CREACION DE BASE DE DATOS .....	47
14.2. ANEXO 2: SCRIPT DE INSERCIÓN DE DATOS DE DIMENSIONES .....	48
14.3. ANEXO 3: SCRIPT DE PROCESOS ETL .....	49

**ÍNDICE DE FIGURAS**

FIGURA 1: Diagrama de Gannt .....	10
FIGURA 2: Diagrama de Gannt .....	11
FIGURA 3: Estructura de carpetas del proyecto .....	13
FIGURA 4: Diagrama de casos de uso .....	17
FIGURA 5: Esquema del diseño conceptual .....	21
FIGURA 6: Diagrama de la arquitectura hardware .....	22
FIGURA 7: Diagrama del diseño software .....	23
FIGURA 8: Conexión a la BD Oracle .....	23
FIGURA 9: Esquema de la Base de datos .....	24
FIGURA 10: Examinar tablas con el Administrador de BD Oracle .....	25
FIGURA 11: Tabla Tiempo .....	25
FIGURA 12: Tabla Segmento .....	25
FIGURA 13: Tabla Prestación .....	26
FIGURA 14: Tabla ComunidadAutonoma .....	26
FIGURA 15: Tabla Población .....	27
FIGURA 16: Tabla Tributación .....	27
FIGURA 17: Carpeta raíz del proyecto .....	28
FIGURA 18: Carpetas ETL del proyecto .....	29
FIGURA 19: Creación del proyecto AtlasSBI .....	33
FIGURA 20: Configuración directorio virtual AtlasSBI .....	34
FIGURA 21: Creación del rol del proyecto .....	35
FIGURA 22: Creación del usuario del proyecto .....	35
FIGURA 23: Acceso al visor web .....	36
FIGURA 24: Documentos analíticos por usuario .....	36
FIGURA 25: Ejemplo presentación de un documento analítico .....	37
FIGURA 26: Informe: Análisis de tributación por año .....	37
FIGURA 27: Informe: Totales de tributación por año, ccaa y tipo de perceptor .....	38
FIGURA 28: Informe: Totales de tributación por año, ccaa y tipo de retribución .....	39
FIGURA 29: Informe: Totales análisis OLAP tributación .....	40
FIGURA 30: Informe: Evolución de las tributaciones por año .....	41
FIGURA 31: Informe: Análisis de ratios de sostenibilidad .....	42
FIGURA 32: Informe: Proyección de indicadores .....	43

### **3. OBJETIVOS**

#### **3.1 OBJETIVOS DE LA ASIGNATURA**

TFC Almacenes de datos requiere una capacidad de análisis de un problema de la vida real, su estudio y la capacidad de construir y diseñar un almacén de datos para su posterior explotación.

Concretamente los objetivos del proyecto son:

- Análisis y estudio del problema introducido en el enunciado.
- Planificación del trabajo a realizar en el calendario.
- Estudiar las posibles opciones de implementación y la viabilidad.
- Realizar la solución de implementación técnica.

Elaborar la memoria del proyecto y su presentación visual.

#### **3.2 OBJETIVOS DEL PROYECTO**

El objetivo de este proyecto es que el OADT consiga la información que necesita para la toma de decisiones que el problema necesita.

Para ello se intentaran cumplir los siguientes propósitos:

- Crear un almacén de datos a partir de los ficheros que proporciona el OADT.
- Crear informes sencillos para mostrar la información que pide el enunciado, según comunidad autónoma, según perceptor o según tipo de retribución:
  - Total retribución.
  - Total retención.
  - Retribución media.
  - % retención media.
  - Número de retribuciones medias.
  - % población por segmento.
  - El ratio de trabajadores por perceptores no activos (parados + pensionistas).
  - El ratio de trabajadores por habitantes.
  - El ratio de trabajadores por personas activas.
  - El ratio de salarios totales por el total de prestaciones.
- Finalmente se realizará una proyección de los indicadores Num. trabajadores/Num. perceptores no activos y Num. trabajadores/Habitantes para comprobar el punto en el que dos trabajadores

mantengan a un perceptor o se iguale el numero de trabajadores con la población perceptora.

### **3.3 ENFOQUE Y METODO SEGUIDO**

El método y el enfoque seguido para la realización del proyecto se ha compuesto del conjunto de capacidades aprendidas durante los años de en los diferentes estudios de ITIG, además del estudio de métodos durante el TFC para la realización de almacenes de datos y data Warehouse. Este método se puede resumir en los siguientes aspectos:

- Análisis previo
  - Establecimiento de objetivos.
  - Análisis de los datos de entrada.
  - Búsqueda de bibliografía.
  - Realización del plan de trabajo.
  
- Análisis de requerimientos
  - Análisis y documentación de los requisitos.
  - Análisis de los datos de origen
  
- Diseño
  - Diseño de modelo multidimensional.
  - Diseño del modelo ROLAP.
  - Diseño del modelo físico.
  
- Implementación
  - Construcción de la BD Oracle.
  - Creación de procedimientos de carga de los datos de origen
  - Depuración de los procesos de carga para adaptar los datos de origen a la BD.
  - Creación de informes del modelo de negocio con la herramienta AtlasSBI.

## 4. PLANIFICACION

### 4.1 PLANIFICACION GLOBAL

Las fases en las que se divide el proyecto y las fechas para las diferentes entregas son las siguientes:

PEC	CONTENIDO	FECHA INICIO	FECHA FIN
PEC 1	Plan de trabajo y análisis preliminar de requerimientos	01/03/2012	16/03/2012
PEC 2	Análisis preliminar de requerimientos y diseño técnico	17/03/2012	20/04/2012
PEC 3	Implementación	21/04/2012	25/05/2012
Entrega Final	Memoria completa y presentación	26/05/2012	13/06/2012
Debate		25/06/2012	28/06/2012

En la PEC 1 analizaremos sin entrar en detalle los datos proporcionados por el OADT y planificaremos el trabajo a realizar.

En la PEC2 se llevará a cabo el diseño de la solución y los procesos de extracción de datos.

En la PEC 3 se implantará la solución creando la estructura en la BD a partir del modelo de datos del diseño.

### 4.2 PLANIFICACION ESPECÍFICA

La planificación se ha pensado teniendo en cuenta un ritmo de trabajo de entre 2 y 3 horas al día, pudiendo disponer del siguiente horario aproximado:

- Lunes a viernes de 17:00 a 20:00
- Sábados de 10:00 a 13:00
- Posibilidad de Domingo de 10:00 a 13:00 aunque no está contemplado su necesidad en el horario.

<b>TAREA</b>	<b>FECHA INICIO</b>	<b>FECHA FIN</b>	<b>DIAS</b>
<b>PEC1</b>	<b>01/03/2012</b>	<b>16/03/2012</b>	<b>16</b>
Recopilación de la documentación	01/03/2012	01/03/2012	1
Lectura de la documentación	02/03/2012	04/03/2012	3
Instalación del software Openproj y familiarización con el entorno	05/03/2012	05/03/2012	1
Análisis preliminar de requerimientos	06/03/2012	08/03/2012	3
Plan de trabajo	09/03/2012	12/03/2012	4
Elaboración documento entrega PEC1	13/03/2012	16/03/2012	4
<b>PEC2</b>	<b>17/03/2012</b>	<b>20/04/2012</b>	<b>35</b>
Instalación del software recomendado	17/03/2012	17/03/2012	1
Recopilación de la documentación	18/03/2012	18/03/2012	1
Lectura de la documentación	19/03/2012	21/03/2012	3
Análisis detallado de los requerimientos	22/03/2012	28/03/2012	7
Revisión del plan de trabajo y análisis de requerimientos	29/03/2012	30/03/2012	2
Creación del modelo de datos	31/03/2012	05/04/2012	6
Diseño conceptual	06/04/2012	10/04/2012	5
Diseño técnico	11/04/2012	14/04/2012	4
Revisión de la documentación	15/04/2012	16/04/2012	2
Elaboración documentación PEC2	17/04/2012	20/04/2012	4
<b>PEC 3</b>	<b>21/04/2012</b>	<b>25/05/2012</b>	<b>35</b>
Instalación de Virtual Box y la maquina virtual con el software necesario	21/04/2012	21/04/2012	1
Familiarización con el entorno	21/04/2012	21/04/2012	1
Recopilación de la documentación	21/04/2012	21/04/2012	1
Lectura de la documentación	21/04/2012	24/04/2012	3
Creación de la base de datos	25/04/2012	01/05/2012	7
Programación de los procedimientos para la carga de datos (ETL)	02/05/2012	06/05/2012	5
Carga de los datos	07/05/2012	08/05/2012	2
Análisis de la información	09/05/2012	10/05/2012	2
Creación de los informes	11/05/2012	16/05/2012	6
Análisis de los resultados	17/05/2012	18/05/2012	2



Elaboración de la documentación PEC 3	19/05/2012	25/05/2012	7
<b>MEMORIA</b>	<b>26/05/2012</b>	<b>13/06/2012</b>	<b>19</b>
Posibles correcciones PEC3	26/05/2012	27/05/2012	2
Elaboración de la memoria	28/05/2012	08/06/2012	12
Revisión final de la memoria	09/06/2012	10/06/2012	2
Elaboración de la presentación virtual	11/06/2012	13/06/2012	3

### **4.3 DIAGRAMA DE GANNT**

A continuación muestro el diagrama de Gannt con las fechas de la planificación.



Figura 1. Diagrama de Gannt

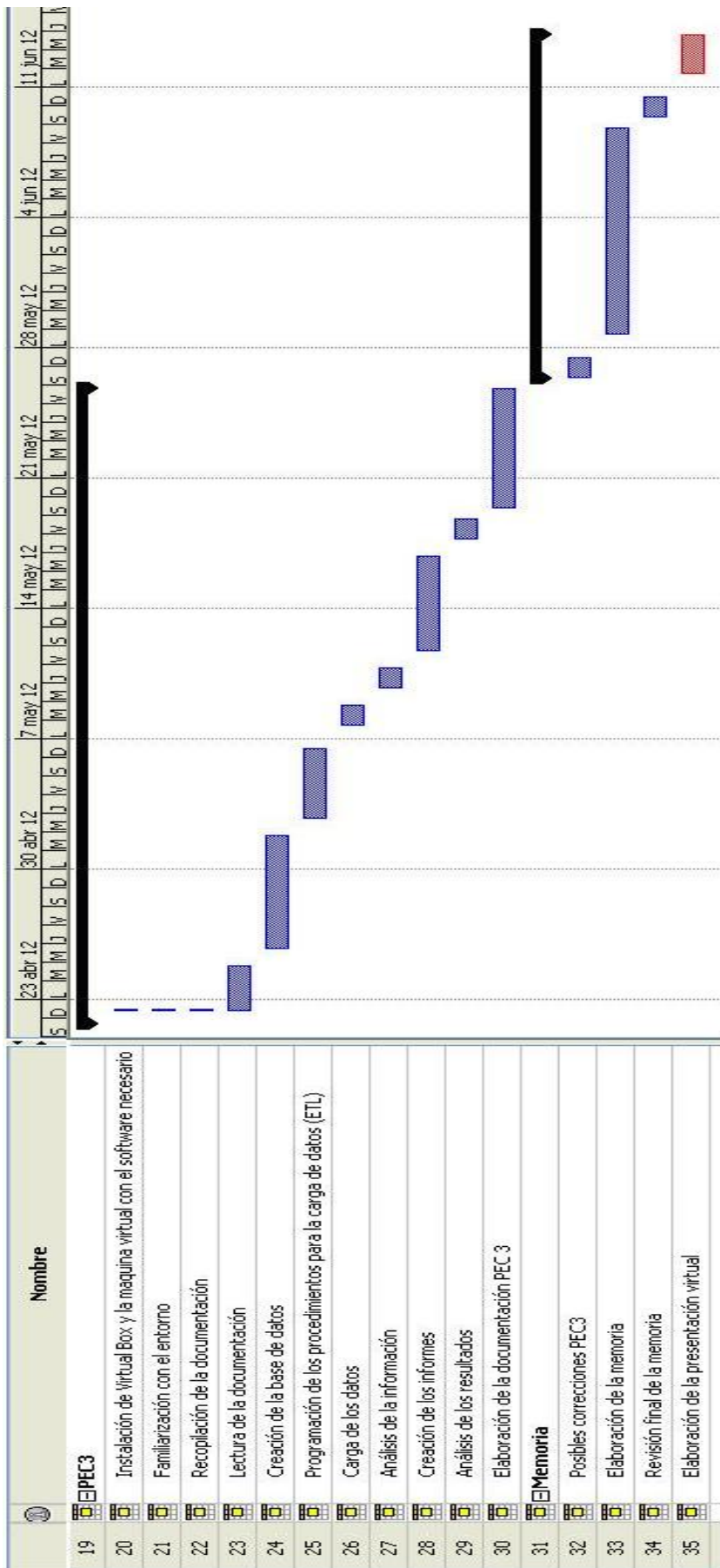


Figura 2. Diagrama de Gantt

## 4.4 INCIDENCIAS Y RIESGOS

Durante el proyecto pueden suceder ciertas incidencias que alteren el curso de la planificación por eso hay que tenerlas en cuenta para tener una mejor solución posible ante tales imprevistos, tenemos los siguientes:

- **Equipo informático:** Se pueden producir problemas técnicos en el equipamiento para llevar a cabo el proyecto, trabajo con un portátil Toshiba de 4 años de antigüedad que aunque no me ha dado problemas y tiene capacidad suficiente de memoria, no esta libre de que se produzcan fallos de hardware o de sistema operativo.

**Solución:** para que el portátil esté funcionando lo más rápido posible se va a realizar una imagen ghost después de la instalación del software necesario de la asignatura, además de realizar copias de seguridad en dos pendrives para mayor contingencia. Si el fallo es de hardware se dispone de otro portátil con menor capacidad pero suficiente para sustituir al equipo principal.

- **Motivos laborales:** Debido a que mi trabajo depende muchas veces de inesperadas cargas de trabajo es posible que mi horario habitual se vea reducido de lunes a viernes.

**Solución:** En este caso habría que hacer un esfuerzo añadido aumentando más horas de trabajo para recuperar tiempo, por ejemplo fines de semana, o dedicando días laborales libres de los que dispongo.

- **Motivos de salud:** Es la contingencia mas difícil de evaluar por lo imprevista que puede llegar a ser.

**Solución:** En este caso como en el anterior habría que hacer un esfuerzo añadido aumentando más horas de trabajo para recuperar tiempo.

El objetivo de la planificación se ha cumplido casi en su totalidad, únicamente un inesperado viaje en Semana Santa por motivos familiares obligó a aumentar las horas de trabajo varios días después.

## 5. RELACION DE PRODUCTOS OBTENIDOS

A continuación paso a enumerar los documentos y productos desarrollados y entregados durante el TFC:

- Documento PEC1 con el análisis preliminar y la planificación del trabajo.
- Documento PEC2 con el diseño conceptual y técnico y el análisis de los requisitos.
- Documento PEC3 con la implementación de los procesos ETL para la carga de los datos proporcionados por el cliente en la BD.
- Documento PEC3 con la construcción del almacén de datos y el acceso a la BD y la herramienta BI.
- Documento PEC3 con las pantallas de los informes solicitados.
- Memoria final con el resumen de todo el trabajo realizado en el TFC.
- Presentación virtual mediante un video que resume el trabajo realizado durante las etapas del proyecto.
- Maquina virtual con el almacén de datos implementado y los datos proporcionados por el cliente cargados en la BD. También con el acceso a la herramienta BI para visualizar los informes creados.

La maquina virtual se entrega con la herramienta AtlasSBI para la generación de informes, configurada y con una serie de informes instalados solicitados por el cliente.

Se entregan también en la maquina virtual, los scripts de generación de la BD y los scripts de carga de los datos en el almacén de datos.

Todos ellos se encuentran en la carpeta TFC dentro de la unidad E:\.

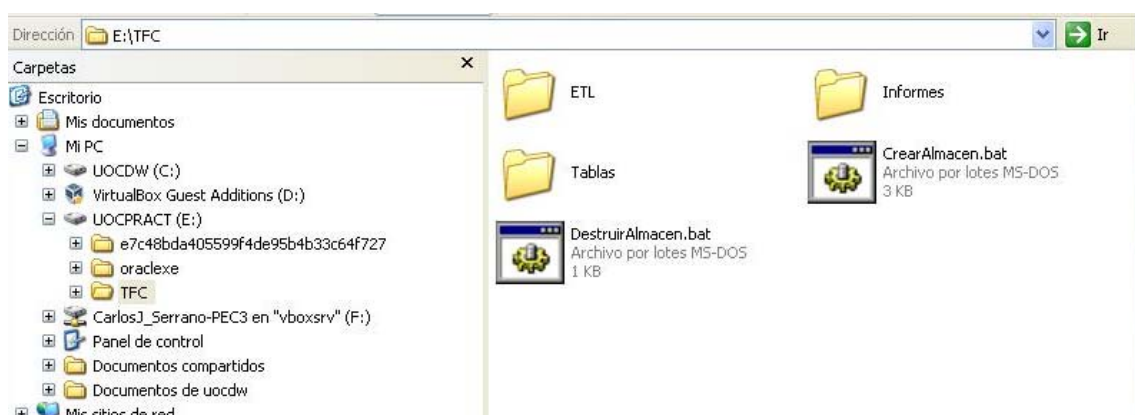


Figura 3. Estructura de carpetas del proyecto.

- **Carpetas:**
  - ETL: contiene todos los scripts necesarios para realizar la carga de los datos csv proporcionados por el cliente. Se verá con detenimiento en el apartado *Proceso ETL de carga y generación del almacén de datos*.
  - Tablas: contiene los scripts para la creación de la BD Oracle y los inserts para las tablas dimensión. También tiene los scripts para borrar las tablas de la BD por si fuera necesario.
  
- **Ficheros:**
  - CrearAlmacen.bat: fichero que llama a los scripts incluidos en la carpeta *Tablas* que crean las tablas de la BD y cargan de datos las tablas dimensión.
  - DestruirAlmacen.bat: fichero que llama a los scripts de borrado de las tablas de la BD.

## 6. BREVE DESCRIPCION DEL RESTO DE APARTADOS

En el resto de capítulos de la memoria se entra en el desarrollo de los apartados en los que hemos dividido el trabajo:

- **Análisis:** análisis de los requisitos entregados por la OADT, se identifica el hecho, medidas y atributos a emplear.
- **Diseño e implementación:** diseño lógico, conceptual y físico de la BD, procesos de carga y generación de la BD.
- **Desarrollo de informes:** se muestran los informes realizados con capturas de pantallas.

Finalmente incluiremos un apartado a modo de conclusión y otro a modo de indicaciones para la evolución del almacén de datos.

## 7. ANALISIS

### 7.1 DATOS DE ORIGEN

Los datos proporcionados por la OADT se encuentran almacenados en varios ficheros csv y son de tres tipos:

- Fichero población por ccaa xxxx.csv: datos de población de cada CC.AA, uno por año, empezando por el año 2005 hasta el año 2011. Se pueden deducir las siguientes columnas:
  - Año.
  - Comunidad Autónoma.
  - Varones.
  - Mujeres.
  
- Fichero Porc Población Activa: información de porcentaje de población activa por CC.AA y año. Entendemos por población activa la que tiene un empleo o lo está buscando. Es decir, la población activa esta compuesta por toda persona en edad laboral que o bien trabaja en un empleo remunerado (población ocupada) o bien se haya en plena búsqueda de empleo (desempleados). Se pueden deducir las siguientes columnas:
  - Comunidad Autónoma.
  - Año.
  - Población Activa.
  
- Fichero tributacion xxxx.csv: información de retribuciones de las personas de las distintas Comunidades Autónomas por cada año. Uno por CC.AA, exceptuando Navarra y el País Vasco que tienen un régimen foral propio de prestaciones sociales.

Además, los datos de tributación de Ceuta y Melilla están unificados en un fichero. Las columnas de estos ficheros son más difíciles de deducir por el formato del fichero.

En el fichero se muestran desglosadas por cada tipo de retribución (salarios, pensiones y desempleo), las retribuciones que cobra cada segmento de perceptor.

Existen tres tipos de perceptores, asalariados, pensionistas y desempleados. En un año pueden darse varios estados en los que puede encontrarse una persona: activo (cobra un salario); activo y

desempleado; activo y pensionista; activo, pensionista y desempleado; desempleado; pensionista; desempleado y pensionista.

En algunos ficheros aparecen algunos datos y en otros no, concretamente retribución total, retribución media, tipo de retención y tipo de retención medio. Más adelante veremos las posibles soluciones a este problema cuando hagamos el diseño.

Generalizando, se pueden deducir los siguientes campos de forma global:

- Año
- Prestación
- Perceptor
- Retribuciones totales (el campo no aparece en todos los ficheros).
- Retribuciones medias (el campo no aparece en todos los ficheros).
- Retenciones totales (el campo no aparece en todos los ficheros).
- Tipo medio retención (el campo no aparece en todos los ficheros).
- Personas

## **7.2 ANÁLISIS DE LOS DATOS DE ORIGEN**

El principal problema que nos podemos encontrar en el análisis de los datos es que la exportación de los datos se ha extraído de diferentes sistemas y por lo tanto hay tres posibles formatos diferentes en la separación de los campos de los ficheros, por coma, por tabulador y por punto y coma. En el proceso de ETL hay que tener en cuenta esta circunstancia a la hora de cargar los datos en la base de datos.

No se aprecian errores en el valor de los datos, dado que estos provienen de sistemas informáticos y no manuales. Sólo tenemos que tener en cuenta los tipos de datos con numeración científica para las cifras muy grandes y los tipos de datos decimales que no tienen sentido en los datos poblacionales y datos monetarios.

También nos encontramos con diferencias en cuanto a la temporalidad de los distintos ficheros, si bien aunque la temporalidad esta medida en años, en los ficheros de tributación no aparece la misma temporalidad que en los ficheros poblacionales, por lo tanto tendremos en cuenta para los informes los años que aparecen en los ficheros de tributación, aunque grabaremos todos los años en tablas para futuras cargas de datos.



### 7.3 DIAGRAMA DE CASOS DE USO

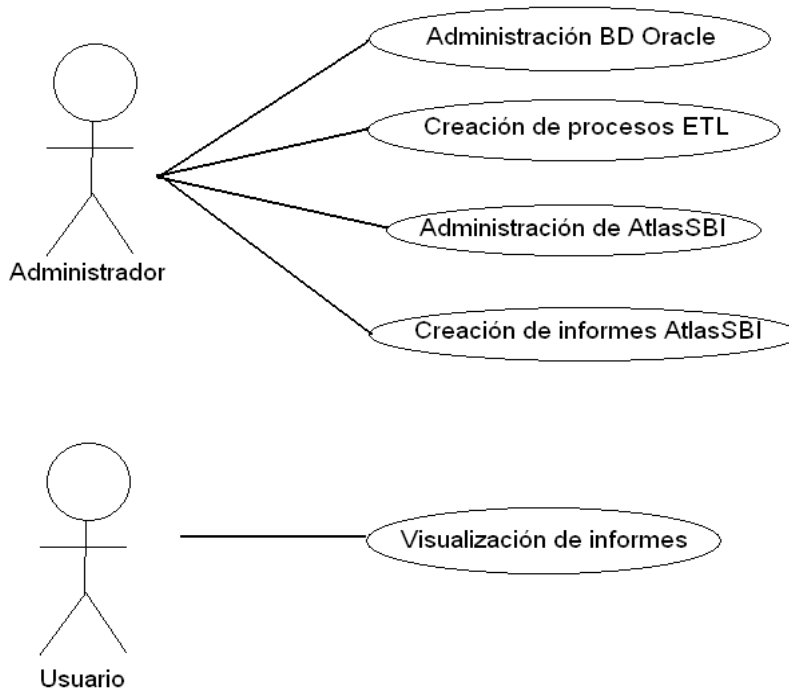


Figura 4. Diagrama de casos de uso.

### 7.4 DIAGRAMA DEL MODELO CONCEPTUAL

#### ESTABLECER EL HECHO

Primeramente establecemos el hecho o hechos que motivan el núcleo del esquema de data Warehouse que queremos construir.

En un primer análisis establecí una sola tabla de hechos con las tributaciones y los perceptores de cada tributación.

Pero dado que la información que se quiere obtener también implica en algunos informes el número de habitantes y la población activa he decidido que hayan dos tablas de hechos: una tabla Tributacion en la que estén reflejadas las retribuciones y retenciones de los perceptores, y otra tabla de hechos en la que se guarden los datos de población de cada comunidad.

Cuando hay varias tablas de hechos el esquema se llama constelación de hechos.

Los motivos de usar este tipo de esquema han sido varios: aumenta la flexibilidad, contribuye a la reutilización de las dimensiones por parte de la tabla de hechos, también existen ciertos informes que se pueden desarrollar

solamente con los datos de la tabla Tributación por lo tanto mejora el rendimiento al tener que recorrer una tabla más pequeña.

Este esquema es más sensible a cambios cuando aumentan las tablas vinculadas, pudiendo aumentar su complejidad, pero dado que no hay previsión de ello nos hemos decidido por este esquema.

## **ENCONTRAR LA GRANULARIDAD ADECUADA**

La OADT nos proporciona la información de los ficheros a nivel de año y nos no pide un detalle mayor, con lo que no es necesario entrar en una granularidad mayor dado los requisitos del problema.

En cuanto al detalle que podemos encontrar en la tabla de tributaciones, un apunte por año, CCAA, segmento y prestación parece una granularidad que no supondría una carga excesiva para el sistema y supondría el nivel de detalle suficiente para resolver algunas consultas que podamos necesitar.

## **DEFINIR LAS DIMENSIONES A EMPLEAR**

Una vez definido el hecho que determina el objetivo del Organismo de Análisis del Departamento de Trabajo, el sistema de prestaciones, debemos de analizar las dimensiones que definan las medidas a valorar dentro de las tablas de hechos.

Necesitamos unas dimensiones que contengan descriptores para las medidas que queremos valorar. La intersección de las dimensiones definen las medidas.

En la fase de planificación, en una primera aproximación, contemplé los atributos año, población y población activa dentro de la dimensión Comunidad Autónoma. Después de un análisis más detallado he concluido que población y población activa son indicadores, ya que cambian cada año y los he incorporado a la tabla de hechos Población.

En cuanto al año entiendo que es conveniente crear una dimensión tiempo que contenga los años en el que se contabiliza la población y la población activa.

Además la dimensión tiempo se compartiría se usaría también como descriptor de tiempo en la tabla de hechos de Tributación como medida de tiempo de la tributación.

La dimensión Segmento contiene todas las combinaciones que se pueden dar en un año de perceptores. He considerado crear una tabla con las combinaciones sin usar jerarquía. Es decir no he creado una tabla Perceptor que se use como nivel anterior a esta. Directamente incluiremos en la carga del fichero de tributación los distintos segmentos de perceptores que se pueden dar en un año.

Las dimensiones que se han identificado son las siguientes:

- Dimensión Tiempo: año en el que se contabiliza la tributación de los perceptores y también año del cálculo de la población de cada CCAA.
- Dimensión Prestación: contiene los tipos de prestación: salarios, pensiones y desempleo.
- Dimensión Comunidad Autónoma: contiene el nombre de las Comunidades Autónomas objeto del análisis.
- Dimensión Segmento: contiene todas las posibles combinaciones que se pueden dar en un año de tipo de perceptor.
  - Asalariado
  - Asalariado y pensionista
  - Asalariado y desempleado
  - Asalariado, pensionista y desempleado
  - Pensionista
  - Desempleado
  - Pensionista y desempleado

## ESTABLECER LOS ATRIBUTOS NECESARIOS

Una vez encontradas las dimensiones, pasamos a analizar cuáles son los atributos que tendrán cada dimensión.

Los valores medios no se pueden ver reflejados en la tabla de hechos dado que en una tabla de hechos se ha de trabajar con valores absolutos por lo tanto consideraremos los atributos con valores totales.

Tampoco debemos de asignar como porcentaje el valor del atributo *poblacionactiva* por la misma razón. Calcularemos el valor absoluto de la población activa.

Tenemos los siguientes atributos:

- Dimensión Tiempo:
  - id\_tiempo: código identificativo.
  - año: año del cálculo de las tributaciones y de la población. Numérico.
- Dimensión Comunidad Autónoma
  - id\_ccaa: código identificativo.
  - nombreCCAA: nombre de la Comunidad Autónoma. Alfanumérico.

- Dimensión Segmento
  - id\_segmento: código identificativo.
  - nombreSegmento: nombre del segmento según el perceptor implicado. Alfanumérico.
  
- Dimensión Prestación
  - id\_prestacion: código identificativo.
  - nombrePrestacion: nombre de la prestación. Alfanumérico.

## **TABLAS DE HECHOS**

- Población
  - id\_tiempo: código identificativo.
  - id\_ccaa: código identificativo.
  - Varones: número de varones. Numérico.
  - Mujeres: número de mujeres. Numérico.
  - totalPoblacion: campo calculado que se compone de la suma de varones y de mujeres. Numérico.
  - poblacionActiva: número de la población que es activa. Numérico.
  
- Tributación
  - id\_tiempo: código identificativo.
  - id\_ccaa: código identificativo.
  - id\_segmento: código identificativo.
  - id\_prestacion: código identificativo.
  - Personas: número de personas por segmento. Numérico.
  - retribucionTotal: cantidad total retribuida a las personas del segmento. Numérico.
  - retencionTotal: cantidad total retenida a las personas del segmento. Numérico.

## ESQUEMA DEL DISEÑO CONCEPTUAL

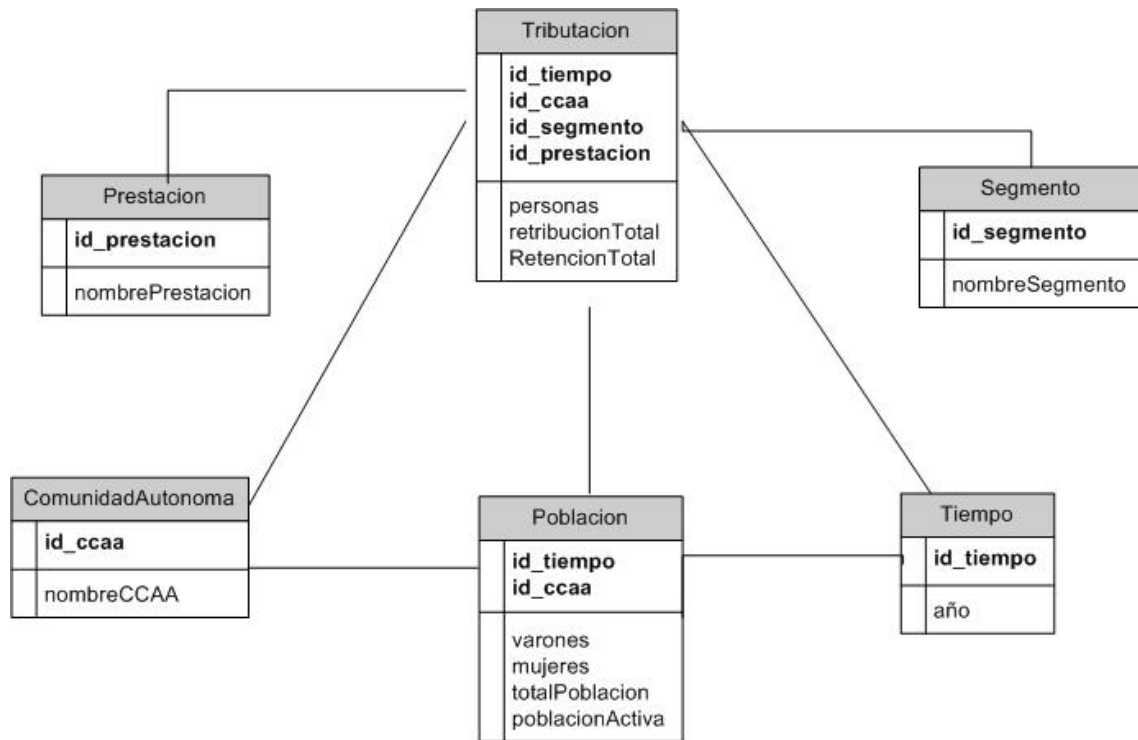


Figura 5. Esquema del diseño conceptual.

## 8. DISEÑO

### 8.1 DIAGRAMA DE ARQUITECTURA DE HARWARE

Una arquitectura mas probable es la que viene definida por un servidor central que tendría instalada la base de datos Oracle con el almacén de datos y otro servidor de aplicaciones con el software AtlasSBI Server y AtlasSBI Studio en la que tendrían acceso diferentes usuarios con permisos para ver los distintos informes.

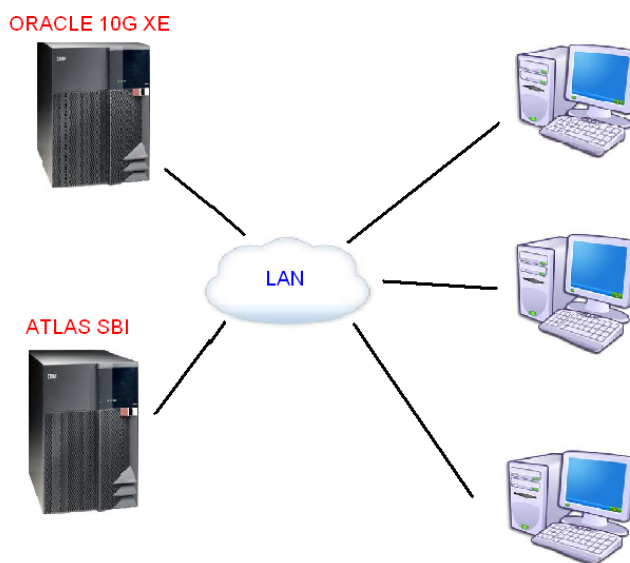


Figura 6. Diagrama de la arquitectura hardware.

### 8.2 DIAGRAMA DE ARQUITECTURA DE SOFTWARE

Para la realización del proyecto se ha utilizado una maquina virtual con Windows XP, con el software de virtualización virtualBox, con los siguientes componentes instalados:

- Oracle Database 10g XE
- SqlPlus
- Atlas SBI Studio
- Atlas SBI Server
- Microsoft SQL Server 2005 Express Edition

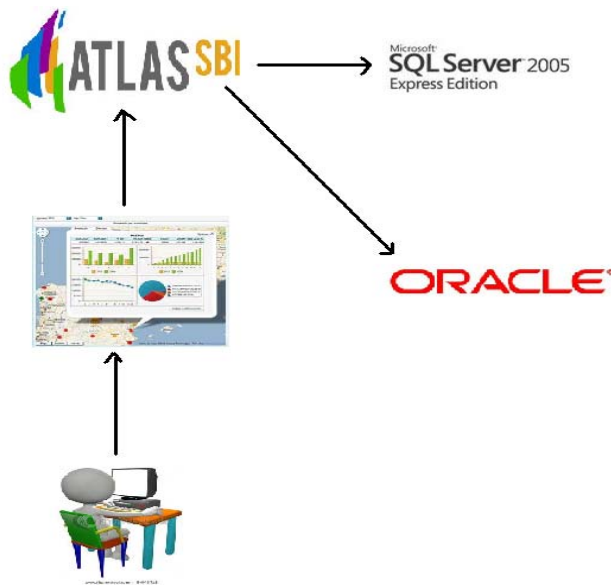


Figura 7. Diagrama del diseño software.

### 8.3 DISEÑO DE LA BASE DE DATOS Y DIAGRAMA DEL MODELO FISICO

Para montar el almacén de datos, se ha creado utilizando el usuario *SYSTEM* y contraseña proporcionada por el cliente), el usuario/esquema de base de datos *cserranogu* con contraseña “uoc” cuyo tablespace por defecto es el *USERS* y tablespace temporal *TEMP*. A este usuario se le han otorgado los roles “CONNECT” y “RESOURCE”.

La gestión de la BD se hará en Oracle Database 10 XE en cuyo entorno web podemos entrar con la contraseña antes proporcionada:



Figura 8. Conexión a la BD Oracle.

## ESQUEMA DE LA BASE DE DATOS

Basándonos en el diseño que desarrollé en la anterior fase el esquema de la BD quedaría en Oracle tal como se muestra a continuación:

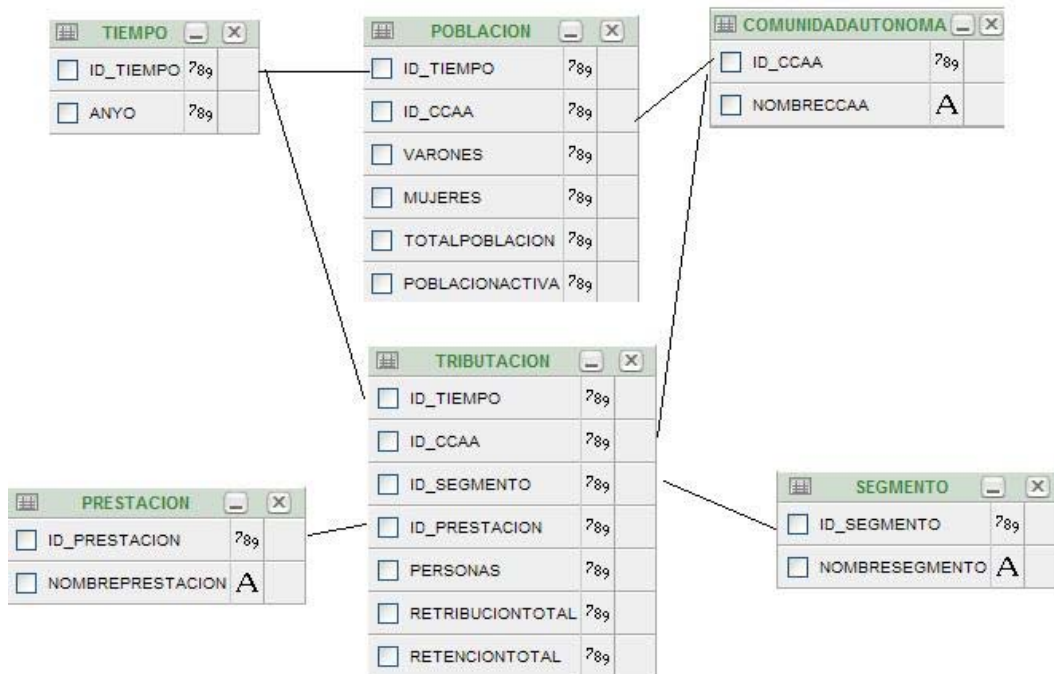


Figura 9. Esquema de la Base de datos.

## TABLAS DE LA BASE DE DATOS

Podemos ver las tablas de la BD en la opción *Explorador de objetos – examinar tablas*.





Figura 10. Examinar tablas con el Administrador de BD Oracle

A continuación paso a detallar las tablas que forman la BD:

- **TIEMPO:** tabla que guarda los años de tributación, desde el 2002 hasta el 2011, que son los años que referencian los ficheros de tributación y población.

Nombre De Columna	Tipo De Dato	Nulo	Valor Por Defecto	Clave Primaria
ID_TIEMPO	NUMBER(2,0)	No	-	1
ANYO	NUMBER(4,0)	No	-	-
				1 - 2

Figura 11. Tabla Tiempo.

ID\_TIEMPO: código identificativo del año. Clave primaria.

ANYO: año correspondiente a la tributación o a la población.

- **SEGMENTO:** tabla que guarda el segmento de perceptor que recibe una prestación.

Nombre De Columna	Tipo De Dato	Nulo	Valor Por Defecto	Clave Primaria
ID_SEGMENTO	NUMBER(2,0)	No	-	1
NOMBRESEGMENTO	VARCHAR2(30)	No	-	-
				1 - 2

Figura 12. Tabla Segmento.

ID\_SEGMENTO: código identificativo del segmento al que pertenece el perceptor. Clave primaria.

NOMBRESEGMENTO: nombre del segmento.

- **PRESTACION:** tabla que contendrá las prestaciones que reciben los perceptores.

Nombre De Columna	Tipo De Dato	Nulo	Valor Por Defecto	Clave Primaria
ID_PRESTACION	NUMBER(2,0)	No	-	1
NOMBREPRESTACION	VARCHAR2(20)	No	-	-
				1 - 2

Figura13. Tabla Prestación.

ID\_PRESTACION: código identificativo de la prestación. Clave primaria.

NOMBREPRESTACION: nombre de la prestación.

- **COMUNIDADAUTONOMA:** tabla con las Comunidades Autónomas sujetas a estudio. Todas las del Estado Español salvo País Vasco y Navarra que tienen régimen foral propio.

Nombre De Columna	Tipo De Dato	Nulo	Valor Por Defecto	Clave Primaria
ID_CCAA	NUMBER(2,0)	No	-	1
NOMBRECCAA	VARCHAR2(50)	No	-	-
				1 - 2

Figura 14. Tabla ComunidadAutonoma.

ID\_CCAA: código identificativo de la Comunidad Autónoma. Clave primaria.

NOMBRECCAA: nombre de la Comunidad Autónoma.

- **POBLACION:** tabla con la población por Comunidad Autónoma, con el total de varones, mujeres y la población activa de cada Comunidad.

Nombre De Columna	Tipo De Dato	Nulo	Valor Por Defecto	Clave Primaria
ID_TIEMPO	NUMBER(2,0)	No	-	1
ID_CCAA	NUMBER(2,0)	No	-	2
VARONES	NUMBER(8,0)	No	-	-
MUJERES	NUMBER(8,0)	No	-	-
TOTALPOBLACION	NUMBER(8,0)	No	-	-
POBLACIONACTIVA	NUMBER(8,0)	Yes	-	-
				1 - 6

Figura 15. Tabla Población.

ID\_TIEMPO: código identificativo del año. Clave foránea a la tabla TIEMPO.

ID\_CCAA: código identificativo de la Comunidad Autónoma. Clave foránea a la tabla COMUNIDADAUTONOMA.

VARONES: cantidad de varones de la Comunidad Autónoma en ese año.

MUJERES: cantidad de varones de la Comunidad Autónoma en ese año.

TOTALPOBLACION: campo compuesto suma de varones y mujeres. Población total de una Comunidad Autónoma en ese año.

POBLACIONACTIVA: Porcentaje de la población con posibilidades de trabajar, que están trabajando o no.

(ID\_TIEMPO, ID\_CCAA): Forman la clave primaria.

- **TRIBUTACION:** tabla que guarda los datos de tributación, retribución y retención, recibida por los distintos segmentos de perceptores.

Nombre De Columna	Tipo De Dato	Nulo	Valor Por Defecto	Clave Primaria
ID_TIEMPO	NUMBER(2,0)	No	-	1
ID_CCAA	NUMBER(2,0)	No	-	2
ID_SEGMENTO	NUMBER(2,0)	No	-	3
ID_PRESTACION	NUMBER(2,0)	No	-	4
PERSONAS	NUMBER(7,0)	No	-	-
RETRIBUCIONTOTAL	NUMBER	Yes	-	-
RETENCIONTOTAL	NUMBER	Yes	-	-
				1 - 7

Figura 16. Tabla Tributación.

ID\_TIEMPO: código identificativo del año. Clave foránea a la tabla TIEMPO.

ID\_CCAA: código identificativo de la Comunidad Autónoma. Clave foránea a la tabla COMUNIDADAUTONOMA.

ID\_SEGMENTO: código identificativo del segmento. Clave foránea a la tabla SEGMENTO.

ID\_PRESTACION: código identificativo de la prestación. Clave foránea a la tabla PRESTACION.

PERSONAS: cantidad de personas que reciben la retribución en ese rango de segmento.

RETRIBUCIONTOTAL: cantidad de retribución que recibe el total de personas en ese rango de segmento.

RETENCIONTOTAL: cantidad total de retención de la retribución total.

(ID\_TIEMPO, ID\_CCAA, ID\_SEGMENTO, ID\_PRESTACION): Clave primaria.

## SCRIPT DE CREACION DE LA BASE DE DATOS

En la carpeta TFC de la unidad E: el fichero *CrearAlmacen.bat* genera todas las tablas de la BD además de llamar a los procesos de carga. Ejecutando este fichero crearemos todas las BD y además llamaremos a su vez al proceso ETL que carga las tablas con los datos de los ficheros csv.

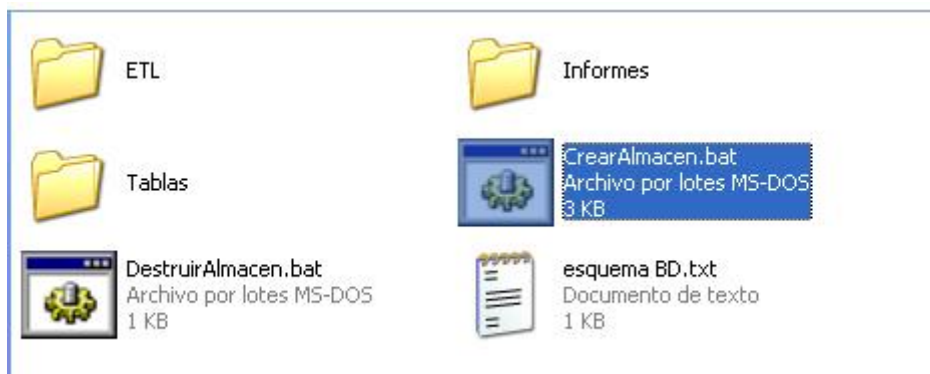


Figura 17. Carpetas raíz del proyecto.

Este fichero llama al script de creación de la BD en la carpeta tablas llamado *CreaTablas.sql*. (**Anexo 1**).

Posteriormente se ejecuta el script de inserción de datos de las dimensiones, llamado *InsertarDatos.sql*. El contenido de las dimensiones se insertará a mano dado que son datos que raramente cambian. (**Anexo 2**).

## 8.4 PROCESO ETL DE CARGA Y GENERACION DE LA BASE DE DATOS

### ESTRUCTURA DE LOS FICHEROS

Los ficheros que se usan para los procesos ETL se encuentran en la ruta E:\TFC\ETL.

La estructura de la carpeta es la siguiente:

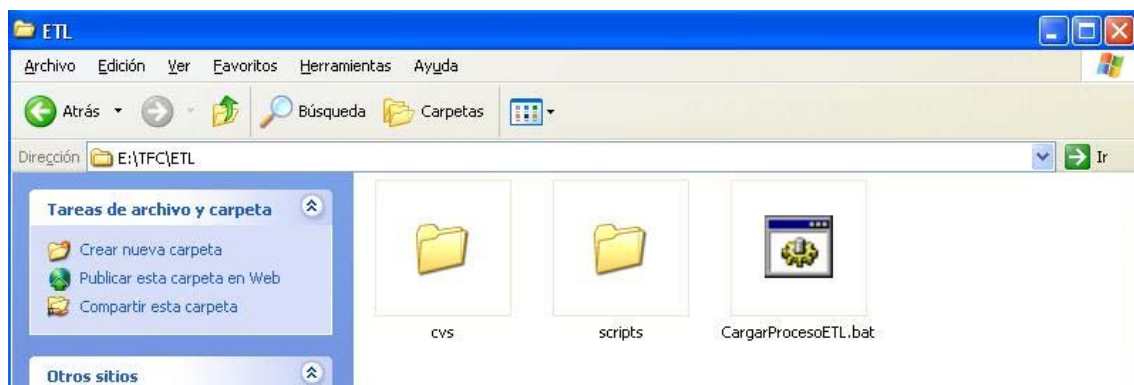


Figura 18. Carpetas ETL del proyecto.

Fichero *CargarProcesoETL.bat*: contiene las llamadas a los ficheros incluidos en la carpeta Scripts que realizan los procesos de transformación y carga en la base de datos. **(Anexo 3)**

Carpeta *csv*: contiene los ficheros de extracción que contiene la información para el almacén de datos.

Carpeta *scripts*: contiene los diversos ficheros que llaman a scripts sql y que realizan toda la carga de los datos de los ficheros csv a la base de datos mediante tablas temporales. También realizan procesos intermedios de transformación para que los ficheros datos ser filtrados e introducidos correctamente a las tablas definitivas creadas según el diseño previo.

La carpeta contiene los siguientes ficheros y carpetas:

Carpeta *logs*: contiene los logs generados por el sqlloader en la carga de los datos.

Carpeta *ctl*: contiene los ficheros extensión ctl con las instrucciones para que el sqlloader cargue los datos en las tablas temporales.

Fichero *CreaTablasETL.bat*: hace la llamada al script *CreaTablasETL.sql* que se encarga de crear todas las tablas temporales para alojar los datos de los ficheros y realizar sobre esas tablas procesos de transformación sobre los datos para cargarlos en las tablas definitivas.

Fichero *CargaDatosETL.bat*: realiza las llamadas a programa sqlloader de cada fichero csv de tributación y de los ficheros de población y población activa.

Fichero *ProcedimientosPobETL.bat*: contiene las llamadas al script *ProcedimientosPobETL.sql*, que se encarga de normalizar los nombres de las tablas temporales cargadas con los ficheros de población y población activa con respecto a los datos de la tabla temporal de COMUNIDADAUTONOMA mediante llamadas a procedimientos almacenados.

Los procedimientos tienen el nombre *normaliza\_nombre\_ccaa\_xxxx* y hay uno por año, se hace un update con el nombre normalizado en las tablas temporales población de cada año.

Posteriormente se lanza otro procedimiento por año, *insertaPoblacionxxxx*, que realiza la inserción de los datos en la tabla de POBLACION.

Fichero *ProcedimientosTribETL.bat*: hace la llamada al script *ProcedimientosTribETL.sql* que contiene los procedimientos necesarios para insertar en la tabla TRIBUTACION los datos de los ficheros de tributación por provincia.

El script *ProcedimientosTribETL.sql* va llamando uno por uno a los scripts que contienen los procedimientos, uno por comunidad autónoma, por ejemplo: *ProcTribCanETL.sql* (pertenecería a Cantabria).

Estos ficheros generan un procedimiento almacenado que se encarga de seleccionar la información necesaria de cada fichero de tributación por cada año, mandarla a una tabla temporal, recorrer la tabla temporal y cargar los datos definitivos en la tabla TRIBUTACION.

Fichero *EjecutaProcETL.bat*: realiza la llamada al script *EjecutaProcETL.sql*, que se encarga de ejecutar los procedimientos uno por uno.

Fichero *BorraObjetosETL.bat*: llama al script *BorraObjetosETL.sql* que se encarga de borrar todas las tablas temporales creadas para el proceso de ETL, para que finalmente sólo queden en la base de datos las tablas del almacén de datos usadas para los informes.

Fichero *NormalizaCantidadesTributacion.bat*: llama al script *CorrigeTributacion.sql* que multiplica por 1000 las cantidades de las retribuciones y las retenciones de todas las Comunidades Autónomas excepto Madrid, Valencia y Murcia partir del año 2006. Estas últimas ofrecen las

cantidades en medias y por ese motivo no le afecta tal anomalía. Desconozco el motivo de la reducción en 1000 de las cantidades pero es un valor que dificulta la lectura de la evolución de las retribuciones.

## **FUNCIONAMIENTO DE LOS PROCESOS ETL**

El proceso empieza con la llamada desde el fichero principal de la creación del almacén de datos *CrearAlmacen.bat*. Este fichero genera las tablas definitivas del DW y llama al fichero *CargarProcesoETL.bat* que lanza el proceso ETL.

El funcionamiento del proceso ETL es el siguiente:

Carga de datos de los ficheros *poblacion por ccaa xxxx.csv* (uno por año), *Porc Poblacion Activa.csv* y *Tributación xxxxxxxx.csv* (uno por Comunidad Autónoma).

La carga se realiza mediante el programa sqlloader dentro del fichero *CargaDatosETL.bat*. **(Anexo 3)**.

Los ficheros de tributación contienen campos que pueden variar de uno a otro, hay algunos que empiezan en el año 2002 y otros en el 2004, además de que hay algunos que contienen columnas con la suma de las retribuciones y otros no.

Al final he optado por cargar todos los campos y luego mediante procedimientos seleccionar los datos para insertar en la tabla TRIBUTACION.

Dado que la tabla temporal de tributación la he creado con 70 campos para que haya de sobra, el fichero ctl debe incluir esos 70 campos en la carga.

Transformación de datos para la introducción en las tablas definitivas:

Se crean mediante los scripts *ProcedimientosPobETL.sql* y *ProcedimientosTribETL.sql* y generan unos procedimientos almacenados para cargar en las tablas definitivas los datos de Población y de Tributación.

Procedimientos de carga en la tabla POBLACION.

En el fichero *ProcedimientosPobETL.sql* existe dos tipos de procedimientos, el primero *normaliza\_nombre\_ccaa\_xxxx* (uno por año), se encarga de modificar

los nombres de ciertas Comunidades Autónomas para que se pueda cruzar con los de la tabla COMUNIDADAUTONOMA y así poder sacar el código de Comunidad Autónoma. **(Anexo 3).**

El siguiente procedimiento se encarga de insertar los datos en la tabla de POBLACION.

Por cada año existe un procedimiento insertaPoblacionxxxx. Crea un cursor que cruza las tablas temporales de población activa y las tablas temporales de población del 2005 y del 2006 dado que el valor de la población en un año es la media de ese año con el siguiente.

En el procedimiento se realizan las operaciones para calcular la media. **(Anexo 3)**

#### Procedimientos de carga en la tabla TRIBUTACION.

En el fichero ProcedimientosPobETL.sql se hace la llamada a los ficheros donde están los procedimientos de carga en la base de datos de tributación, uno por Comunidad Autónoma. Los procedimientos se llaman tributacionxxxxxxxx (uno por Comunidad Autónoma). Como ejemplo podemos ver @@ProcTribAndETL.sql donde se crea el procedimiento tributacionAndalucia que carga en la tabla de TRIBUTACION de los datos de Andalucía.

El procedimiento crea un cursor con la selección de todos los campos de una tabla llamada TEMPORAL. Y por cada año inserta en la tabla temporal la selección de los campos de ese año. Posteriormente se recorre la tabla temporal hasta que se encuentra el valor del primer campo que sea igual a cada uno de las prestaciones: salario, desempleo y pensión. Las siguientes filas son los valores de la tributación que corresponde a cada una de las prestaciones. Una vez que se recorran todas las filas el procedimiento se termina.

Los procedimientos pueden variar un poco dependiendo si los ficheros contienen o no columna todos, que no se recoge en la tabla TRIBUTACION, o si tienen retribución media, que tampoco se guarda en la tabla. En tal caso hay que desechar estos valores.

Muestro el ejemplo del procedimiento para recoger los datos del año 2003 en Andalucía, el resto de años es igual nada más que cambiando el valor del año. **(Anexo 3)**

Finalmente se llama al script *CorrigeTributacion.sql* para corregir las cantidades de las tributaciones a partir del año 2006.



## 9. INFORMES: DISEÑO Y CAPTURAS DE PANTALLA

### CREACION DEL PROYECTO DE ATLAS SBI

Para la generación de informes y la generación de un escenario del modelo de negocio para cada usuario usamos la herramienta AtlasSBI.

AtlasSBI necesita para su funcionamiento interno una BD Sql Server Express, que se crea en la instalación. Necesitaremos las credenciales a la hora de crear nuevos proyectos.

**Nombre de la BD: AtlasSBI**

**Usuario: sa**

**Contraseña: 12345**

Para construir los informes creamos un proyecto nuevo en AtlasSBI Studio llamado OADT.

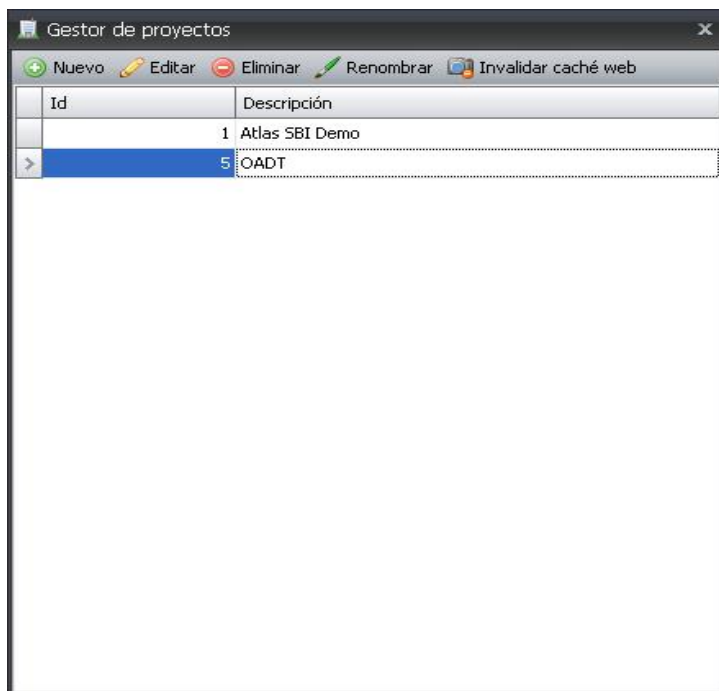


Figura 19. Creación del proyecto AtlasSBI.

Para poder visualizar los informes por el navegador necesitamos crear un directorio virtual del proyecto y configurar el acceso a la BD de Sql Server.

Para ello copiamos la carpeta *AtlasSBI* en la ruta `c:\inetpub\wwwroot` y le damos el nombre del proyecto *OADT*.

Finalmente creamos el directorio virtual en *Servicios de Internet Information Server* con nombre *OADT* y apuntamos a la ruta donde hemos puesto la carpeta.

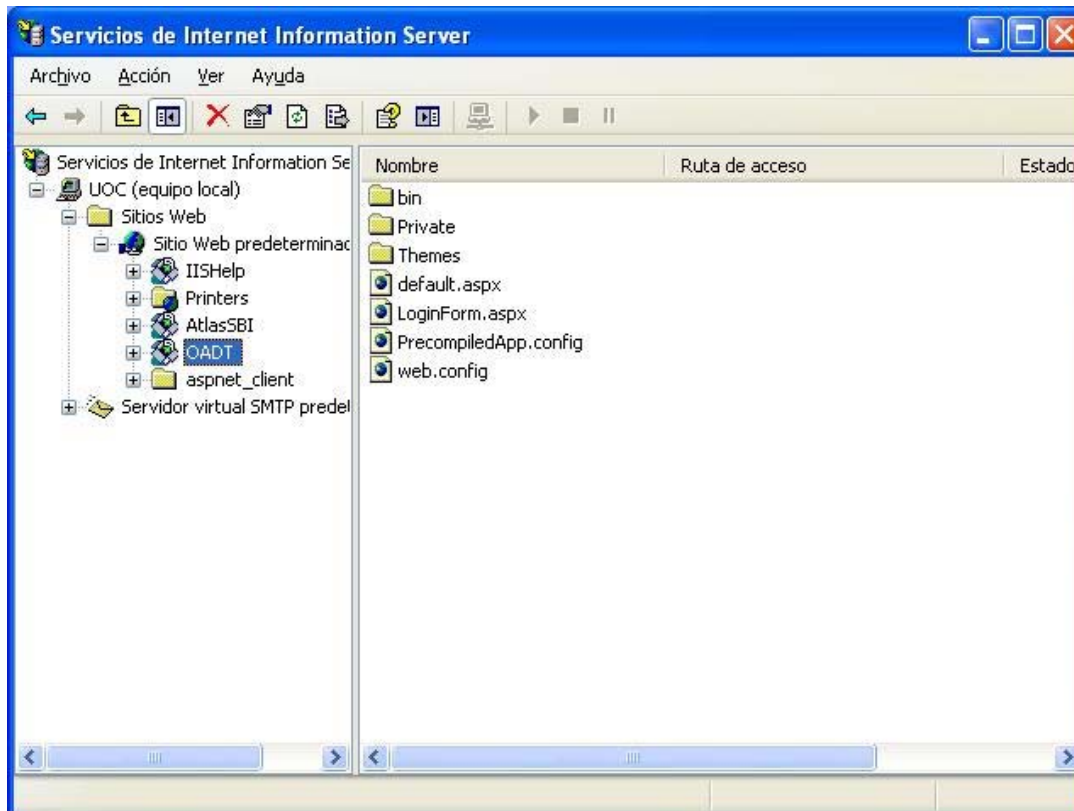


Figura 20. Configuración directorio virtual AtlasSBI.

Habr  que configurar el acceso a la base de datos de Sql Server en las propiedades del directorio virtual, en *Editar Configuraci n* de ASP.Net editando los par metros de conexi n de AtlasSBI:

```
Data Source=(local)\AtlasSBI;Initial Catalog=AtlasSBI;User
ID=sa;Password=12345
```

## CREACION DEL USUARIO PARA INFORMES

Una vez que hemos creado el proyecto y el directorio virtual para el acceso a la herramienta para ver los informes necesitamos crear un usuario para asignarle los informes.

Primero crearemos un rol en AtlasSBI Studio que llamaremos roloadt. El rol va a contener los documentos anal ticos de nuestro modelo de negocio.

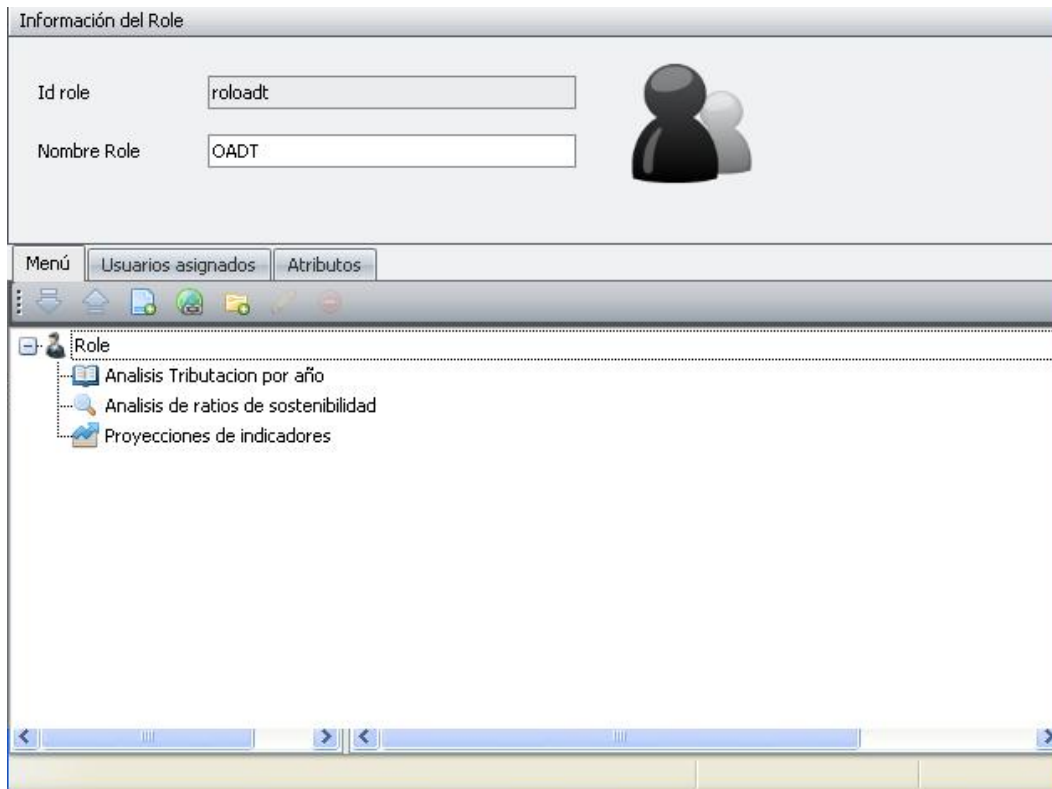


Figura 21. Creación del rol del proyecto.

A continuación crearemos el usuario con el que nos identificaremos en el visor web. El usuario debe estar asignado al rol que acabamos de crear.

**Usuario: oadt**  
**Contraseña: uoc**



Figura 22. Creación del usuario del proyecto.

## ACEESO AL ENTORNO WEB PARA VISUALIZAR LOS INFORMES

Para ver los informes tenemos que acceder a la siguiente URL:

<http://localhost/OADT>

A continuación aparecerá la ventana del visor web pidiendo el usuario y el password. Introducimos el usuario que hemos creado, *oadt* y la contraseña *uoc*.

The image shows a login form titled "Atlas SBI - Identificación". On the left, there is a logo consisting of several overlapping, colorful shapes (green, blue, yellow, red). The form contains two input fields: "Usuario" with the text "oadt" and "Contraseña" with three black dots. Below the password field is a checkbox labeled "No cerrar sesión" which is currently unchecked. At the bottom right of the form is a button labeled "Acceder".

Figura 23. Acceso al visor web.

Una vez dentro tenemos en el panel de la izquierda arriba los documentos analíticos que tenemos asignados y abajo los que hemos abierto hasta ahora.

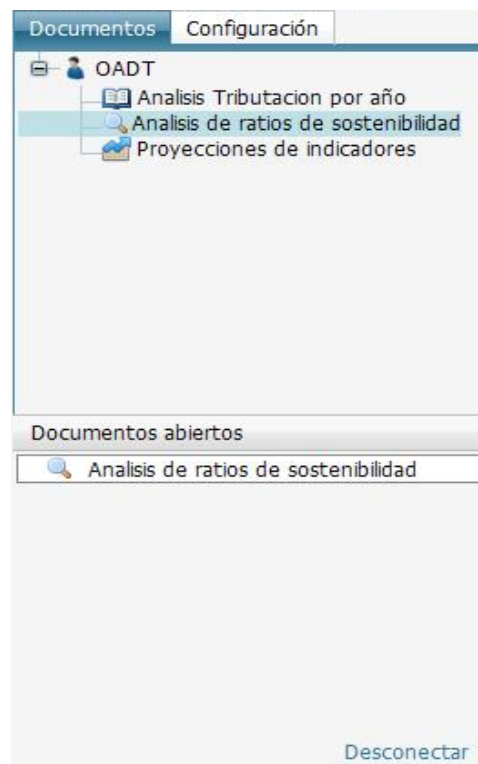


Figura 24. Documentos analíticos por usuario.

En el panel de la derecha se mostrará el documento que estemos visualizando.

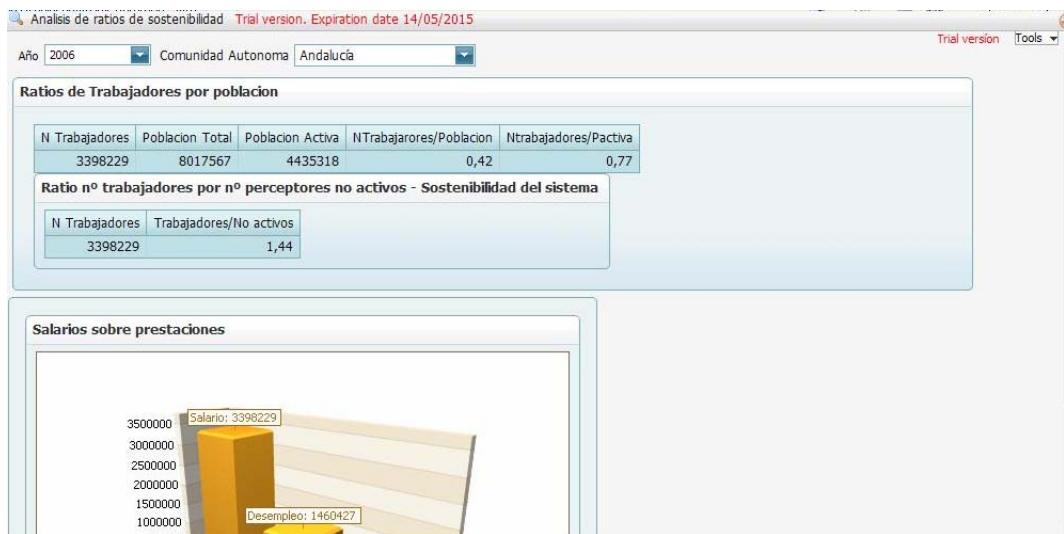


Figura 25. Ejemplo presentación de un documento analítico.

## INFORMES: CAPTURAS DE PANTALLA

- **Análisis de tributación por año**

En realidad este informe es una cabecera que recoge en un control *tab* los siguientes tres informes por que están relacionados.

Se puede elegir cualquiera de estos tres eligiendo una pestaña.

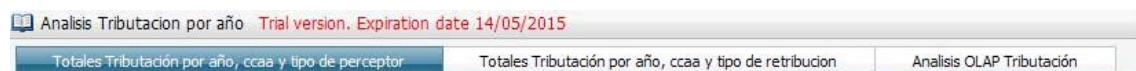


Figura 26. Informe: Análisis de tributación por año.

○ **Totales de tributación por año, ccaa y tipo de perceptor**

En este informe se muestra la información de la tributación por año y por Comunidad Autónoma. En este caso podemos elegir el tipo de perceptor.

La información que se muestra del tipo de perceptor es: la retribución total del segmento, la retención total de esa retribución, así mismo también se muestra la retribución media, la retención media total y los porcentajes de retención media y el número de retribuciones dadas.

Finalmente en el informe se muestran dos gráficos que ayudan a aclarar un poco la información requerida por la OADT, es una comparativa de retribuciones por segmento y de personas por segmento. También contiene un desglose de cada perceptor que tipo de retribución tiene asociado, así por ejemplo el perceptor asalariado solo ha recibido un salario, sin embargo el perceptor asalariado y desempleado ha podido recibir en ese año un salario y una prestación por desempleo. En el desglose se muestran las personas la retención y la retribución.

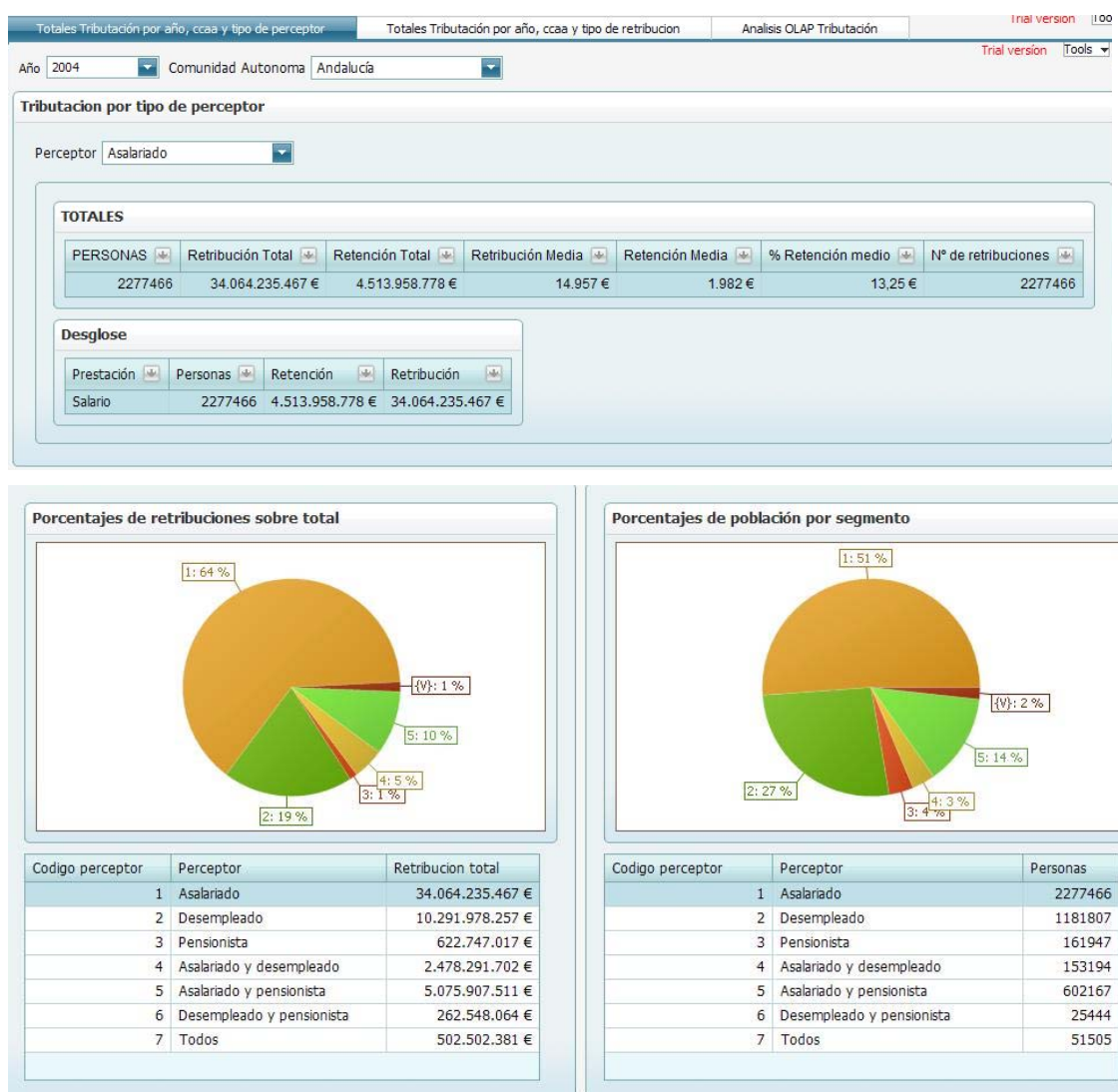


Figura 27. Informe: Totales de tributación por año, ccaa y tipo de perceptor.

○ **Totales de tributación por año, ccaa y tipo de retribución**

Este informe es igual que el anterior pero en vez de mostrar información dependiendo del segmento a que pertenece la persona lo muestra por el tipo de retribución que cobra: salario, desempleo o pensión.

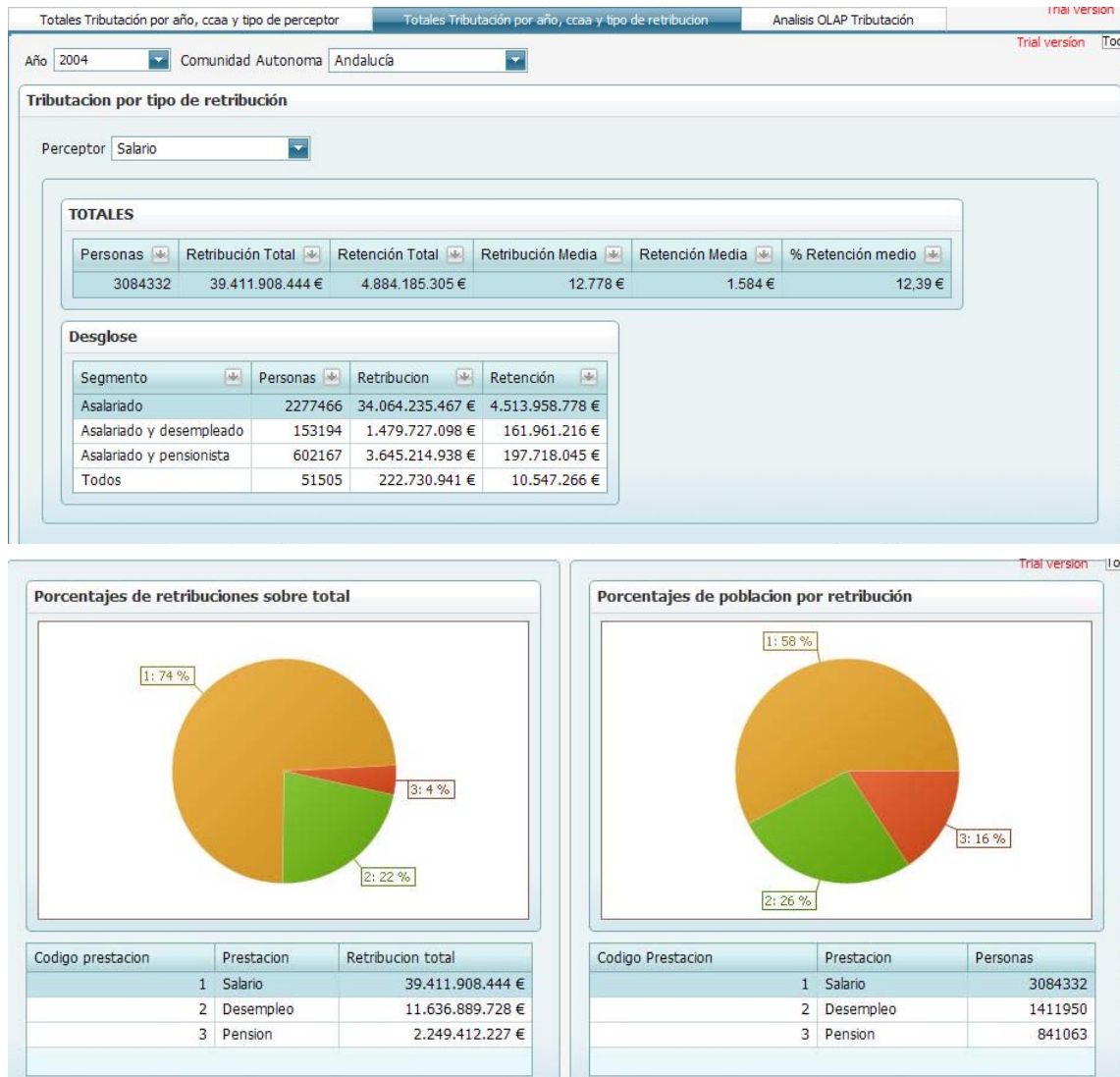


Figura 28. Informe: Totales de tributación por año, ccaa y tipo de retribución.



○ **Análisis OLAP tributación**

Este informe, muy práctico, se muestra en forma de cubos OLAP la información reflejada en la tabla de TRIBUTACION. En el muestra las dimensiones y los indicadores relacionados con las dimensiones, retribuciones, retenciones y personas.

Podemos seleccionar los años que nos interesen y ver la información de cada año dividida por perceptores y retribuciones.

Año			Total General		
Personas	Retribución	Retención	Personas	Retribución	Retención
Comunidad Autónoma	Perceptor	Prestación			
Andalucía	Asalariado	Salario	16486279	103.217.037.106,00 €	13.670.414.322,00 €
	Asalariado y desempleado	Desempleo	1272068	3.128.523.836,00 €	159.425.892,00 €
		Salario	1272068	4.469.790.745,00 €	485.486.763,00 €
	Asalariado y pensionista	Pension	4841366	4.331.872.888,00 €	5.253.075,00 €
		Salario	4841366	10.927.729.127,00 €	606.525.481,00 €
	Desempleado	Desempleo	8435830	30.845.051.740,00 €	1.129.553.943,00 €
	Desempleado y pensionista	Desempleo	201166	527.773.104,00 €	26.631.958,00 €
		Pension	201166	242.657.471,00 €	605.095,00 €
	Pensionista	Pension	1371172	1.871.111.564,00 €	6.438.761,00 €
	Todos	Desempleo	430382	508.443.796,00 €	24.861.928,00 €
Pension		430382	351.092.335,00 €	359.263,00 €	
Salario		430382	673.221.470,00 €	33.007.824,00 €	
	Asalariado	Salario	2866324	16.818.259.067,00 €	2.497.386.955,00 €
	Asalariado y desempleado	Desempleo	223540	554.738.434,00 €	37.742.672,00 €
		Salario	223540	724.599.870,00 €	90.943.002,00 €

Figura 29. Informe: Totales análisis OLAP tributación.



○ **Evolución de las tributaciones por año**

En este informe vemos la evolución que han tenido las retribuciones, retenciones y personas que han cobrado distintas prestaciones. Se puede observar que los salarios han ido creciendo a lo largo de los años hasta el 2009 que han ido disminuyendo cuando ya empezaban a notarse los efectos de la crisis a la vez que han aumentado las prestaciones por desempleo. Podemos observar el generalizado crecimiento de personas y por lo tanto retribuciones de pensionistas en el 2009. Otro factor que afecta negativamente al sistema de prestaciones.

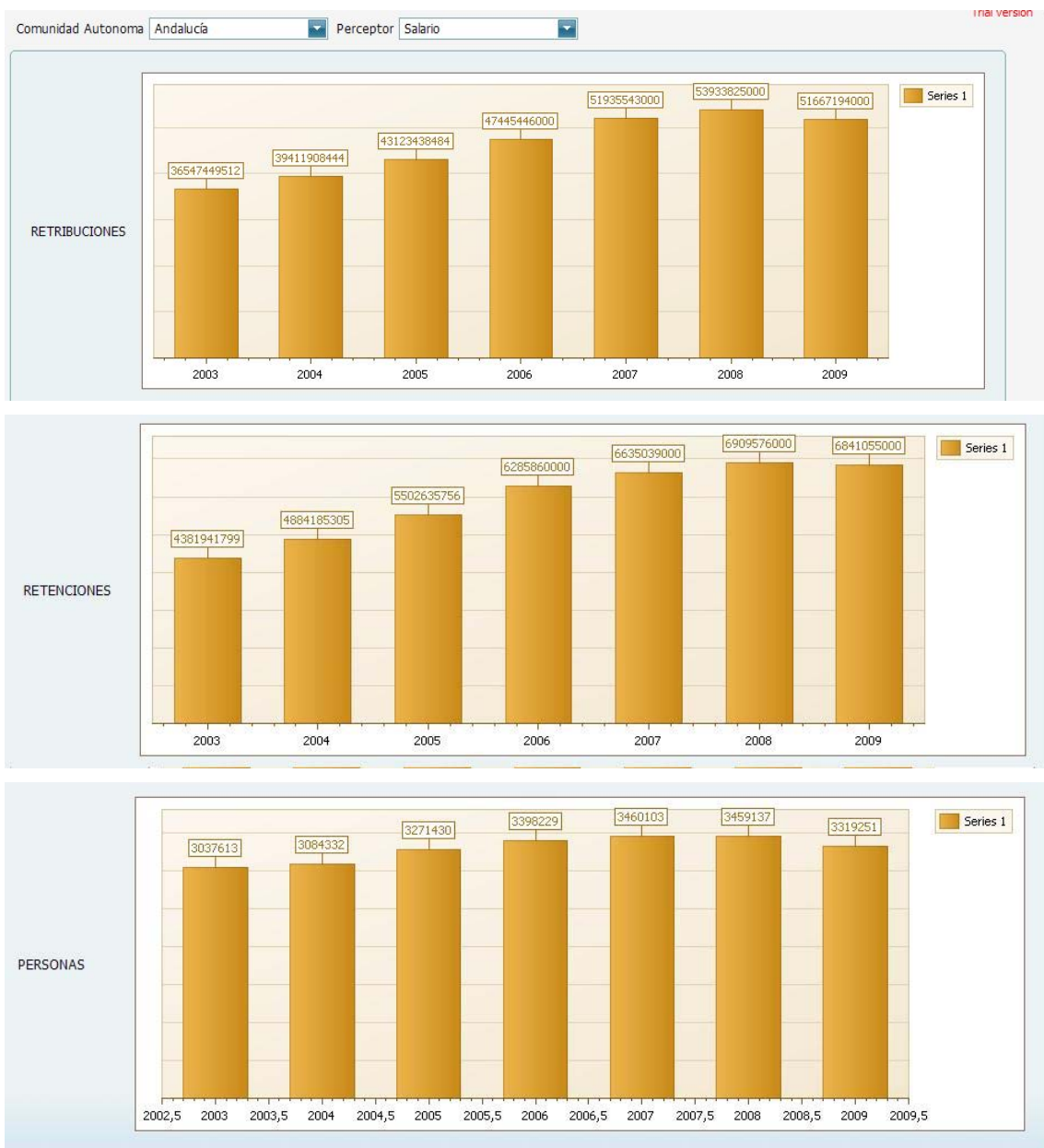


Figura 30. Evolución de las tributaciones por año.

- **Análisis de ratios de sostenibilidad**

Este informe muestra la información de los ratios para medir la sostenibilidad del sistema de prestaciones. Estos ratios miden la capacidad que tienen los trabajadores para sostener a las personas que reciben prestación por parte del Estado. Se miden por años y por Comunidad Autónoma. Los ratios que se muestran son: Número de trabajadores por habitantes, número de trabajadores por la población activa y el último es el número de habitantes por los que reciben prestación. Este último es el dato que puede servir para medir la sostenibilidad del sistema de prestaciones. Y es el dato que nos puede llevar a deducir si hay suficientes trabajadores para pagar las prestaciones.

Finalmente se muestra un gráfico de salarios sobre prestaciones.

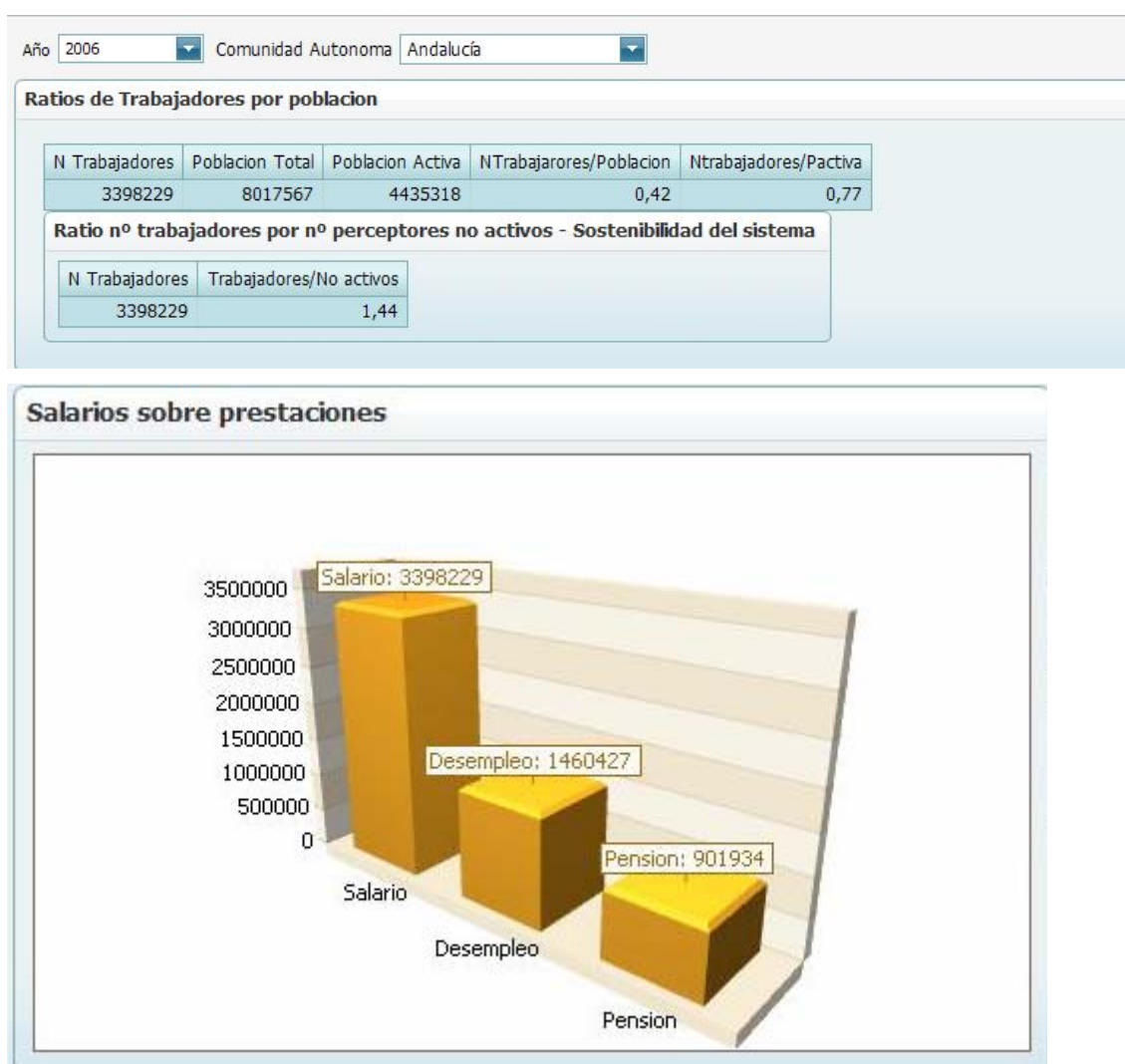


Figura 31. Informe: Análisis de ratios de sostenibilidad.

- **Proyecciones de indicadores**

En este informe se muestran dos gráficos en el que se muestran dos proyecciones de los indicadores nº de trabajadores por nº de perceptores no activos y nº trabajadores por población. Estos indicadores son herramientas para deducir cuando se produciría el colapso del sistema de prestaciones a este ritmo. El indicador de numero de trabajadores por población desciende a un ritmo pequeño pero vemos que nº de trabajadores por nº de perceptores no activos mantiene la tendencia a producir un colapso cada vez se acerca mas a la barrera de que un trabajador mantenga a un perceptor y se podría producir para mediados del 2017 según vemos en el punto que corta la línea de colapso en el eje y la línea de la tendencia del ratio en el eje x. De esta manera vemos que la población no ha aumentado demasiado sino que es el problema del desempleo y el envejecimiento de la población la principal causa de que se pueda producir el colapso en años próximos.



Figura 32. Informe: Proyección de indicadores.

## 10. CONCLUSIONES

Después de todo un semestre de retos abordados en esta asignatura he llegado al punto en el que sin duda alguna he conseguido el objetivo que me propuse al apuntarme a este TFC: aprender como funciona un almacén de datos.

Ha sido el trabajo de transformación y carga en la BD lo que más quebraderos de cabeza me ha dado, ya que no dominaba el trabajo con el pl/sql y tenía conocimientos básicos de SQL, con el que he tenido que trabajar horas y horas. Además por circunstancias de falta de tiempo he tenido que aprender el funcionamiento de la herramienta AtlasSBI para la realización de informes, sin poder profundizar en ella lo suficiente.

Finalmente, he de decir, que considero cumplido el objetivo de la ODAT, construir un almacén de datos que pueda ser útil en el propósito de la organización, comprobar la evolución que va a tomar el sistema de prestaciones sociales en el futuro y así poder tomar decisiones que puedan mejorar el sistema actual y que visto el resultado de los informes no lleva a una situación demasiado buena.

Con todo, la experiencia adquirida en este proyecto, espero que me sirva para saber conllevar otros trabajos destinados a realizar bases de datos para el análisis y la toma de decisiones que hoy en día son tan útiles en las empresas.

## 11. LINEAS DE EVOLUCION FUTURAS

Este proyecto se ha tenido en cuenta dentro de una temporalidad desde el 2004 hasta el 2009, dado que los ficheros de tributación de las algunas Comunidades Autónomas lo contemplaban de esa manera. Los ficheros de población y de población activa ampliaban el abanico de años hasta el 2011.

Para dejar la puerta abierta a futuras ampliaciones de los años de tributación, sería necesario modificar los scripts de carga simplemente añadiendo los nuevos años en los procedimientos de población, población activa y tributación de la misma manera que están hechos los anteriores.

También sería necesario a su vez, insertar los nuevos años en la tabla de tiempo.

El resto de tablas de dimensiones es muy difícil que cambie su situación dado que son datos muy estáticos. En el caso de cambio de algún dato de la tabla de Comunidades Autónomas, prestaciones o retribuciones simplemente habrá que añadir o modificar el dato mediante un script en la BD.

## 12. GLOSARIO

**Almacén de datos (Data Warehouse):** colección de datos orientada a un determinado ámbito (empresa, organización, etc.), integrado, no volátil y variable en el tiempo, que ayuda a la toma de decisiones en la entidad en la que se utiliza.

**Business Intelligence:** o inteligencia empresarial, conjunto de estrategias y herramientas enfocadas a la administración y creación de conocimiento mediante el análisis de datos existentes en una organización.

**OLAP:** solución utilizada en la Business Intelligence (BI) para agilizar la consulta de grandes cantidades de datos integrada en los data Warehouse. Para ello utiliza estructuras multidimensionales (o Cubos OLAP) que contienen datos resumidos de grandes almacenes de datos.

**Hecho:** actividad de la organización objeto del análisis.

**Medida:** indicador determinante sobre el hecho, valor con el que queremos trabajar. Es uno de los atributos del cubo.

**Dimensión:** son elementos que contienen atributos (o campos) que se utilizan para restringir y agrupar los datos almacenados. Estructura que representa una de las caras del cubo.

**Tabla de hechos:** la tabla central de un esquema dimensional que contiene los valores de las medidas de negocio.

**Tabla de dimensiones:** tabla que contienen los atributos dimensiones para la organización de las medidas a valorar.

**Constelación de hechos:** esquema dimensional que contiene múltiples tablas de hechos. Con esta solución las tablas de dimensiones pueden estar compartidas entre más que una tabla de hechos.

**Procesos ETL:** (Extract, Transform and Load). Procesos que facilitan el mover datos desde múltiples fuentes, reformatearlos y limpiarlos, y cargarlos en otra base de datos para analizar la información.

**Oracle:** sistema de gestión de base de datos relacional fabricado por Oracle Corporation.

## 13. BIBLIOGRAFIA

[http://sistemas.uniandes.edu.co/~isis2304/dokuwiki/lib/exe/fetch.php?media=tutoriales:tutorial\\_general\\_sqloader.pdf](http://sistemas.uniandes.edu.co/~isis2304/dokuwiki/lib/exe/fetch.php?media=tutoriales:tutorial_general_sqloader.pdf)

<http://www.dataprix.com/data-warehousing-y-metodologia-hefesto/arquitectura-del-data-warehouse/34-datawarehouse-manager#x1-220002.8.1>

<http://es.wikipedia.org/>

[http://members.fortunecity.com/2horasdejazz/soporte/curso\\_loader.html](http://members.fortunecity.com/2horasdejazz/soporte/curso_loader.html)

[http://wwwdi.ujaen.es/~molina/bd1/apuntes\\_prac.pdf](http://wwwdi.ujaen.es/~molina/bd1/apuntes_prac.pdf)

<http://www.oracle.com/technetwork/index.html>

<http://www.oracleutilities.com/OSUtil/sqlldr.html>

[http://www.orafaq.com/wiki/SQL\\*Loader\\_FAQ](http://www.orafaq.com/wiki/SQL*Loader_FAQ)

<http://www.proyectosfindecarrera.com/>

<http://elvex.ugr.es/idbis/db/docs/intro/F%20Modelo%20multidimensional.pdf>

<http://infolab.stanford.edu/infoseminar/Archive/FallY99/russakovskii-slides/index.htm>

<http://www.adictosaltrabajo.com/tutoriales/tutoriales.php?pagina=datawarehouse3#2.13.2.1.Tabla%20de%20Hechos%20%28Tablas%20Fac%29outline>

<http://kbase.atlassbi.com/>

## 14. ANEXOS

### ANEXO 1: SCRIPT DE CREACION DE BASE DE DATOS

```

CREATE table "SEGMENTO" (
    "ID_SEGMENTO"    NUMBER(2) NOT NULL,
    "NOMBRESEGMENTO" VARCHAR2(30) NOT NULL,
    constraint "SEGMENTO_PK" primary key ("ID_SEGMENTO")
);

CREATE table "TIEMPO" (
    "ID_TIEMPO"    NUMBER(2) NOT NULL,
    "ANYO"         NUMBER(4) NOT NULL,
    constraint "TIEMPO_PK" primary key ("ID_TIEMPO")
);

CREATE table "PRESTACION" (
    "ID_PRESTACION"    NUMBER(2) NOT NULL,
    "NOMBREPRESTACION" VARCHAR2(20) NOT NULL,
    constraint "PRESTACION_PK" primary key ("ID_PRESTACION")
);

CREATE table "COMUNIDADAUTONOMA" (
    "ID_CCAA"    NUMBER(2) NOT NULL,
    "NOMBRECCAA" VARCHAR2(50) NOT NULL,
    constraint "COMUNIDADAUTONOMA_PK" primary key ("ID_CCAA")
);

CREATE table "POBLACION" (
    "ID_TIEMPO"    NUMBER(2) NOT NULL,
    "ID_CCAA"      NUMBER(2) NOT NULL,
    "VARONES"     NUMBER(8) NOT NULL,
    "MUJERES"     NUMBER(8) NOT NULL,
    "TOTALPOBLACION" NUMBER(8) NOT NULL,
    "POBLACIONACTIVA" NUMBER(8) NULL,
    CONSTRAINT "POBLACION_PK" primary key
("ID_TIEMPO", "ID_CCAA"),
    CONSTRAINT "POBLACION_CCAA_FK" FOREIGN KEY ("ID_CCAA")
REFERENCES "COMUNIDADAUTONOMA" ("ID_CCAA"),
    CONSTRAINT "POBLACION_TIEMPO_FK" FOREIGN KEY ("ID_TIEMPO")
REFERENCES "TIEMPO" ("ID_TIEMPO")
);

CREATE table "TRIBUTACION" (
    "ID_TIEMPO"    NUMBER(2) NOT NULL,

```

```

"ID_CCAA"          NUMBER(2) NOT NULL,
"ID_PRESTACION"   NUMBER(2) NOT NULL,
"ID_SEGMENTO"     NUMBER(2) NOT NULL,
"PERSONAS"        NUMBER(7) NOT NULL,
"RETRIBUCIONTOTAL" NUMBER,
"RETENCIONTOTAL"  NUMBER,
CONSTRAINT "TRIBUTACION_PK" primary key ("ID_TIEMPO", "ID_CCAA",
"ID_SEGMENTO", "ID_PRESTACION"),
CONSTRAINT "TRIBUTACION_TIEMPO_FK" FOREIGN KEY ("ID_TIEMPO")
REFERENCES "TIEMPO" ("ID_TIEMPO"),
CONSTRAINT "TRIBUTACION_CCAA_FK" FOREIGN KEY ("ID_CCAA")
REFERENCES "COMUNIDADAUTONOMA" ("ID_CCAA"),
CONSTRAINT "TRIBUTACION_PRESTACION_FK" FOREIGN KEY
("ID_PRESTACION") REFERENCES "PRESTACION" ("ID_PRESTACION"),
CONSTRAINT "TRIBUTACION_SEGMENTO_FK" FOREIGN KEY ("ID_SEGMENTO")
REFERENCES "SEGMENTO" ("ID_SEGMENTO")
);

```

## ANEXO 2: SCRIPT DE INSERCIÓN DE DATOS DE DIMENSIONES

```

-- Insertamos en la tabla Tiempo los valores de los años
INSERT INTO TIEMPO VALUES (1, 2002);
INSERT INTO TIEMPO VALUES (2, 2003);
INSERT INTO TIEMPO VALUES (3, 2004);
INSERT INTO TIEMPO VALUES (4, 2005);
INSERT INTO TIEMPO VALUES (5, 2006);
INSERT INTO TIEMPO VALUES (6, 2007);
INSERT INTO TIEMPO VALUES (7, 2008);
INSERT INTO TIEMPO VALUES (8, 2009);
INSERT INTO TIEMPO VALUES (9, 2010);
INSERT INTO TIEMPO VALUES (10, 2011);

-- Insertamos los valores de las Comunidades Autonomas
INSERT INTO COMUNIDADAUTONOMA VALUES (1, 'Andalucía');
INSERT INTO COMUNIDADAUTONOMA VALUES (2, 'Aragón');
INSERT INTO COMUNIDADAUTONOMA VALUES (3, 'Principado de
Asturias');
INSERT INTO COMUNIDADAUTONOMA VALUES (4, 'Islas Baleares');
INSERT INTO COMUNIDADAUTONOMA VALUES (5, 'Canarias');
INSERT INTO COMUNIDADAUTONOMA VALUES (6, 'Cataluña');
INSERT INTO COMUNIDADAUTONOMA VALUES (7, 'Cantabria');

```



```
INSERT INTO COMUNIDADAUTONOMA VALUES (8, 'Castilla - La
Mancha');
INSERT INTO COMUNIDADAUTONOMA VALUES (9, 'Castilla y León');
INSERT INTO COMUNIDADAUTONOMA VALUES (10, 'Comunidad
Valenciana');
INSERT INTO COMUNIDADAUTONOMA VALUES (11, 'Extremadura');
INSERT INTO COMUNIDADAUTONOMA VALUES (12, 'Galicia');
INSERT INTO COMUNIDADAUTONOMA VALUES (13, 'La Rioja');
INSERT INTO COMUNIDADAUTONOMA VALUES (14, 'Comunidad de
Madrid');
INSERT INTO COMUNIDADAUTONOMA VALUES (15, 'Región de Murcia');
INSERT INTO COMUNIDADAUTONOMA VALUES (16, 'Ceuta');
INSERT INTO COMUNIDADAUTONOMA VALUES (17, 'Melilla');

-- Insertamos las prestaciones
INSERT INTO PRESTACION VALUES (1, 'Salario');
INSERT INTO PRESTACION VALUES (2, 'Desempleo');
INSERT INTO PRESTACION VALUES (3, 'Pension');

-- Insertamos todos los segmentos posibles de los receptores de
prestacion
INSERT INTO SEGMENTO VALUES (1, 'Asalariado');
INSERT INTO SEGMENTO VALUES (2, 'Desempleado');
INSERT INTO SEGMENTO VALUES (3, 'Pensionista');
INSERT INTO SEGMENTO VALUES (4, 'Asalariado y desempleado');
INSERT INTO SEGMENTO VALUES (5, 'Asalariado y pensionista');
INSERT INTO SEGMENTO VALUES (6, 'Desempleado y pensionista');
INSERT INTO SEGMENTO VALUES (7, 'Todos');
```

## ANEXO 3: SCRIPTS PROCESOS ETL

### Fichero *CargarProcesoETL.bat*:

```
ECHO OFF
CD ./scripts

rem ---- llamamos al fichero que borra todos los objetos
temporales de la BD
call BorraObjetosETL.bat
```

```
rem ---- llamamos al fichero que crea las tablas temporales
para la carga de datos
call CreaTablasETL.bat

rem ---- llamamos al fichero que carga las tablas temporales de
datos de los ficheros
call CargaDatosETL.bat

rem ---- llamamos al fichero que crea los procedimientos para
la entrada de datos
rem ---- de la tabla de Poblacion
call ProcedimientosPobETL.bat

rem ---- llamamos al fichero que crea los procedimientos para
la entrada de datos
rem ---- de la tabla de Tributacion
call ProcedimientosTribETL.bat

rem ---- llamamos al fichero que ejecuta todos los
procedimientos
call EjecutaProcETL.bat

rem ---- llamamos al fichero que borra todos los objetos
temporales de la BD
call BorraObjetosETL.bat
```

### **Fichero *CargaDatosETL.bat*:**

Carga del fichero de *poblacion por ccaa xxxx.csv*.

```
sqlldr cserranogu/uoc@XE control=.\ctl\poblacion2005.ctl
log=.\logs\poblacion2005.log

sqlldr - programa ejecutable
cserranogu/uoc@XE - usuario de la base de datos
control=.\ctl\poblacion2005.ctl - ubicación del fichero de
control en que están los comandos que el sqlloader interpreta
para la carga de los datos
log=.\logs\poblacion2005.log - ubicación del fichero de log con
los resultados de la importación
```

El fichero ctl de esa línea es el siguiente:

```
OPTIONS (SKIP=2)
LOAD DATA
INFILE '..\cvs\poblacion por ccaa 2005.csv'
BADFILE '..\logs\poblacion2005.bad'
REPLACE
INTO TABLE "POBLACION_CCAA_2005_tmp"
FIELDS TERMINATED BY "," OPTIONALLY ENCLOSED BY ''
(CCAA,
VARONES,
MUJERES)
```

Carga desde el fichero *población por ccaa 2005.csv*, eliminando las dos primeras líneas (OPTIONS (SKIP=2)). Carga los ficheros en la tabla temporal *POBLACION\_CCAA\_2005\_tmp*, en los campos *CCAA*, *VARONES* Y *MUJERES*.

Carga del fichero *Porc Poblacion Activa.csv* es:

```
sqlldr cserranogu/uoc@XE control=.\ctl\pactiva.ctl
log=.\logs\pactiva.log
```

El fichero ctl de esa línea es el siguiente:

```
OPTIONS (SKIP=4)
LOAD DATA
INFILE '..\cvs\Porc Poblacion Activa.csv'
BADFILE '..\logs\pactiva.bad'
REPLACE
INTO TABLE "POBLACION_ACTIVA_tmp"
FIELDS TERMINATED BY "," OPTIONALLY ENCLOSED BY ''
(CCAA CHAR,
ANYO2010 CHAR,
ANYO2009 CHAR,
ANYO2008 CHAR,
ANYO2007 CHAR,
ANYO2006 CHAR,
ANYO2005 CHAR)
```

Fichero ejemplo de una Comunidad para un fichero de tributación, por ejemplo *Tributación Andalucía.csv* es:

```
sqlldr cserranogu/uoc@XE control=.\ctl\tributacionAndalucia.ctl  
log=.\logs\tributacionAndalucia.log
```

Y el fichero ctl de esta línea es el siguiente:

```
LOAD DATA  
INFILE '..\cvs\Tributación Andalucía.csv'  
BADFILE '..\logs\tributacionAndalucia.bad'  
REPLACE  
INTO TABLE "TRIBUTACION_ANDALUCIA_tmp"  
FIELDS TERMINATED BY "," OPTIONALLY ENCLOSED BY '' '  
TRAILING NULLCOLS  
(CAMPO1 CHAR,  
CAMPO2 CHAR,  
CAMPO3 CHAR,  
CAMPO4 CHAR,  
CAMPO5 CHAR,  
CAMPO6 CHAR,  
CAMPO7 CHAR,  
CAMPO8 CHAR,  
CAMPO9 CHAR,  
CAMPO10 CHAR,  
CAMPO11 CHAR,  
CAMPO12 CHAR,  
CAMPO13 CHAR,  
CAMPO14 CHAR,  
CAMPO15 CHAR,  
CAMPO16 CHAR,  
.....  
CAMPO70 CHAR)
```

## Procedimientos de carga en la tabla POBLACION

### Ejemplo procedimiento normaliza\_nombre\_ccaa\_xxxx

```

CREATE PROCEDURE normaliza_nombre_ccaa_2005
IS
BEGIN
    update "POBLACION_CCAA_2005_tmp" set ccaa = 'Principado de Asturias' where ccaa = 'Asturias (Principado de)';
    update "POBLACION_CCAA_2005_tmp" set ccaa = 'La Rioja' where ccaa = 'Rioja (La)';
    update "POBLACION_CCAA_2005_tmp" set ccaa = 'Comunidad de Madrid' where ccaa = 'Madrid (Comunidad de)';
    update "POBLACION_CCAA_2005_tmp" set ccaa = 'Región de Murcia' where ccaa = 'Murcia (Región de)';
    update "POBLACION_CCAA_2005_tmp" set ccaa = 'Islas Baleares' where ccaa = 'Balears (Illes)';
    update "POBLACION_CCAA_2005_tmp" set ccaa = 'Ceuta' where ccaa = 'Ciudad autónoma de Ceuta';
    update "POBLACION_CCAA_2005_tmp" set ccaa = 'Melilla' where ccaa = 'Ciudad autónoma de Melilla';
END;

```

### Ejemplo procedimiento insertaPoblacionxxxx:

```

CREATE PROCEDURE insertaPoblacion2005
IS
BEGIN
    DECLARE
        CURSOR p IS
            SELECT C.ID_CCAA, A.CCAA, A.VARONES, A.MUJERES, D.VARONES, D.MUJERES, B.ANYO2005
            FROM "POBLACION_CCAA_2005_tmp" A
            INNER JOIN "POBLACION_ACTIVA_tmp" B
            ON A.CCAA=B.CCAA
            INNER JOIN "COMUNIDADAUTONOMA" C
            ON A.CCAA=C.NOMBRECCAA
            INNER JOIN "POBLACION_CCAA_2006_tmp" D
            ON A.CCAA=D.CCAA
            ORDER BY C.ID_CCAA;

    id number(2);

```

```
nombrecaa varchar2(50);
-- recogemos en las variables string
varones2005 varchar2(10);
mujeres2005 varchar2(50);
varones2006 varchar2(50);
mujeres2006 varchar2(50);
pob_activa varchar2(50);
-- guardamos en las variables de tipos de la tabla POBLACION
varones number;
mujeres number;
total number;
pactiva number;

BEGIN
-- convertimos los tipos al que va a tener la tabla POBLACION.
-- abrimos el cursor
OPEN p;
LOOP
-- guardamos en las variables la primera fila del cursor
  FETCH p into id, nombrecaa, varones2005, mujeres2005,
varones2006, mujeres2006, pob_activa;
  EXIT WHEN p%NOTFOUND;
-- calculamos el valor del campo varones y mujeres del año
2005 como media del año 2005 a 1 de Enero
-- y 2006 a 1 de Enero,
  varones := (TO_NUMBER(varones2005) +
TO_NUMBER(varones2006))/2;
  mujeres := (TO_NUMBER(mujeres2005) +
TO_NUMBER(mujeres2006))/2;
  varones := ROUND(varones);
  mujeres := ROUND(mujeres);
-- calculamos el campo poblacion activa de la tabla
poblacion en total de personas y no en porcentaje
-- con la siguiente formula: poblacion total *
pob_activa/100
  total := varones + mujeres;
  pactiva := total * (to_number(pob_activa)/100);
  pactiva := ROUND(pactiva);
-- insertamos en la tabla Poblacion con el id de tiempo
que es el 4
  INSERT INTO POBLACION VALUES (4,id, varones, mujeres,
total, pactiva);
END LOOP;
```

```
CLOSE p;  
END;  
END;
```

## Procedimientos de carga en la tabla TRIBUTACION

Ejemplo procedimiento tributacionxxxxxxxx:

```
CREATE PROCEDURE tributacionAndalucia  
IS  
    CURSOR c IS  
    SELECT * FROM "TEMPORAL";  
    cabecera varchar(50);  
    asal varchar(50);  
    pens varchar(50);  
    des varchar(50);  
    asal_pen varchar(50);  
    asal_des varchar(50);  
    asal_pen_des varchar(50);  
    pen_des varchar(50);  
    -- datos de personas  
    perasal varchar(50);  
    perpens varchar(50);  
    perdes varchar(50);  
    perasal_pen varchar(50);  
    perasal_des varchar(50);  
    perasal_pen_des varchar(50);  
    perpen_des varchar(50);  
    -- datos de retribuciones  
    retrasal varchar(50);  
    retrpens varchar(50);  
    retrdes varchar(50);  
    retrasal_pen varchar(50);  
    retrasal_des varchar(50);  
    retrasal_pen_des varchar(50);  
    retrpen_des varchar(50);  
    -- datos de retenciones  
    retenasal varchar(50);
```

```
retenpens varchar(50);
retendes varchar(50);
retenasal_pen varchar(50);
retenasal_des varchar(50);
retenasal_pen_des varchar(50);
retenpen_des varchar(50);
ccaa number;
anyo number;
seg number;
BEGIN

-- La CCAA va a ser siempre la misma
ccaa := 1;
-----
-- insertamos los datos en la tabla temporal del año 2003
-----
anyo := 2;

INSERT INTO "TEMPORAL"
  SELECT CAMPO1,
  CAMPO3,CAMPO4,CAMPO5,CAMPO6,CAMPO7,CAMPO8,CAMPO9 FROM
  "TRIBUTACION_ANDALUCIA_tmp";

-- Abrimos el cursor para recorrer la tabla temporal
OPEN c;
LOOP
  -- comprobamos que la cabecera es SALARIOS
  EXIT WHEN c%NOTFOUND;
  FETCH c INTO cabecera, asal, pens, des, asal_pen, asal_des,
  pen_des, asal_pen_des;

  IF (cabecera = 'SALARIOS') THEN
    seg := 1;
    FETCH c INTO cabecera, asal, pens, des, asal_pen,
    asal_des, pen_des, asal_pen_des;
    IF cabecera = 'Personas' THEN
      perasal := asal;
      perasal_pen := asal_pen;
      perasal_des := asal_des;
      perasal_pen_des := asal_pen_des;
      FETCH c INTO cabecera, asal, pens, des, asal_pen,
      asal_des, pen_des, asal_pen_des;
```



```
END IF;
IF cabecera = 'Retribuciones' THEN
    retrasal := asal;
    retrasal_pen := asal_pen;
    retrasal_des := asal_des;
    retrasal_pen_des := asal_pen_des;
    FETCH c INTO cabecera, asal, pens, des, asal_pen,
asal_des, pen_des, asal_pen_des;
END IF;
IF cabecera = 'Retenciones' THEN
    retenasal := asal;
    retenasal_pen := asal_pen;
    retenasal_des := asal_des;
    retenasal_pen_des := asal_pen_des;
    FETCH c INTO cabecera, asal, pens, des, asal_pen,
asal_des, pen_des, asal_pen_des;
END IF;
-- insertamos en la tabla definitiva los datos de la
prestacion salario de todos los segmentos que tengan salario
INSERT INTO "TRIBUTACION" VALUES (anyo, ccaa, seg, 1,
to_number(perasal), to_number(retrasal), to_number(retenasal));
INSERT INTO "TRIBUTACION" VALUES (anyo, ccaa, seg, 4,
to_number(perasal_pen), to_number(retrasal_pen),
to_number(retenasal_pen));
INSERT INTO "TRIBUTACION" VALUES (anyo, ccaa, seg, 5,
to_number(perasal_des), to_number(retrasal_des),
to_number(retenasal_des));
INSERT INTO "TRIBUTACION" VALUES (anyo, ccaa, seg, 7,
to_number(perasal_pen_des), to_number(retrasal_pen_des),
to_number(retenasal_pen_des));
END IF;

IF (cabecera = 'PENSIONES') THEN
    seg := 2;
    FETCH c INTO cabecera, asal, pens, des, asal_pen,
asal_des, pen_des, asal_pen_des;
    IF cabecera = 'Personas' THEN
        perpens := pens;
        perasal_pen := asal_pen;
        perpen_des := pen_des;
        perasal_pen_des := asal_pen_des;
        FETCH c INTO cabecera, asal, pens, des, asal_pen,
asal_des, pen_des, asal_pen_des;
    END IF;
END IF;
```

```
        IF cabecera = 'Retribuciones' THEN
            retrpens := pens;
            retrasal_pen := asal_pen;
            retrpen_des := pen_des;
            retrasal_pen_des := asal_pen_des;
            FETCH c INTO cabecera, asal, pens, des, asal_pen,
            asal_des, pen_des, asal_pen_des;
        END IF;
        IF cabecera = 'Retenciones' THEN
            retenpens := pens;
            retenasal_pen := asal_pen;
            retenpen_des := pen_des;
            retenasal_pen_des := asal_pen_des;
            FETCH c INTO cabecera, asal, pens, des, asal_pen,
            asal_des, pen_des, asal_pen_des;
        END IF;
        -- insertamos en la tabla definitiva los datos de la
        -- prestacion salario de todos los segmentos que tengan salario
        INSERT INTO "TRIBUTACION" VALUES (anyo, ccaa, seg, 2,
        to_number(perpens), to_number(retrpens), to_number(retenpens));
        INSERT INTO "TRIBUTACION" VALUES (anyo, ccaa, seg, 4,
        to_number(perasal_pen), to_number(retrasal_pen),
        to_number(retenasal_pen));
        INSERT INTO "TRIBUTACION" VALUES (anyo, ccaa, seg, 6,
        to_number(perpen_des), to_number(retrpen_des),
        to_number(retenpen_des));
        INSERT INTO "TRIBUTACION" VALUES (anyo, ccaa, seg, 7,
        to_number(perasal_pen_des), to_number(retrasal_pen_des),
        to_number(retenasal_pen_des));
    END IF;

    IF (cabecera = 'DESEMPLEO') THEN
        seg := 3;
        FETCH c INTO cabecera, asal, pens, des, asal_pen,
        asal_des, pen_des, asal_pen_des;
        IF cabecera = 'Personas' THEN
            perdes := des;
            perasal_des := asal_des;
            perpen_des := pen_des;
            perasal_pen_des := asal_pen_des;
            FETCH c INTO cabecera, asal, pens, des, asal_pen,
            asal_des, pen_des, asal_pen_des;
        END IF;
    END IF;
```

```
        IF cabecera = 'Retribuciones' THEN
            retrdes := des;
            retrasal_des := asal_des;
            retrpen_des := pen_des;
            retrasal_pen_des := asal_pen_des;
            FETCH c INTO cabecera, asal, pens, des, asal_pen,
            asal_des, pen_des, asal_pen_des;
        END IF;
        IF cabecera = 'Retenciones' THEN
            retendes := des;
            retenasal_des := asal_des;
            retenpen_des := pen_des;
            retenasal_pen_des := asal_pen_des;
            FETCH c INTO cabecera, asal, pens, des, asal_pen,
            asal_des, pen_des, asal_pen_des;
        END IF;
        -- insertamos en la tabla definitiva los datos de la
        -- prestacion salario de todos los segmentos que tengan salario
        INSERT INTO "TRIBUTACION" VALUES (anyo, ccaa, seg, 3,
        to_number(perdes), to_number(retrdes), to_number(retendes));
        INSERT INTO "TRIBUTACION" VALUES (anyo, ccaa, seg, 5,
        to_number(perasal_des), to_number(retrasal_des),
        to_number(retenasal_des));
        INSERT INTO "TRIBUTACION" VALUES (anyo, ccaa, seg, 6,
        to_number(perpen_des), to_number(retrpen_des),
        to_number(retenpen_des));
        INSERT INTO "TRIBUTACION" VALUES (anyo, ccaa, seg, 7,
        to_number(perasal_pen_des), to_number(retrasal_pen_des),
        to_number(retenasal_pen_des));
    END IF;

    END LOOP;
    CLOSE c;
    -- truncamos los datos de la tabla para añadir los datos de
    -- otro año
    EXECUTE IMMEDIATE 'TRUNCATE TABLE "TEMPORAL"';
    -----
    ---- fin de año 2003 -----
    -----
```