

Título:

MOTOR WEB DE CUADROS NUMÉRICOS Y GRÁFICOS ESTADÍSTICOS SEGÚN PATRONES DE ESTRATEGIA DE PROGRAMACIÓN ORIENTADO A OBJETOS.

Palabras clave:

Software, Objetos, Diseño, Patrones, Motor, Cuadros Numéricos, Cuadros Gráficos, Estadísticas, Web, Internet, XML, Estándares, Calidad.

Autores:

Ricardo de Alfonso López
Subdirector General de Estadística de los Servicios.
Inspección General del Ministerio de Economía y Hacienda

Javier Bueno Infantes
Consejero. Subdirección General de Estadística de los Servicios.
Inspección General del Ministerio de Economía y Hacienda

Francisco José Jiménez Ibarra
Analista Programador. Subdirección General de Estadística de los Servicios.
Inspección General del Ministerio de Economía y Hacienda

1.- Antecedentes**1.1.- Antecedentes Legislativos/Administrativos**

El Real Decreto 1552/2004, de 25 de junio, por el que se desarrolla la estructura orgánica básica del Ministerio de Economía y Hacienda, determina en sus Artículo 22 apartados c) y d) que la Inspección General del Ministerio de Economía y Hacienda, órgano directivo con rango de dirección general, ejercerá entre otras las siguientes funciones:

c) La inspección de los servicios prevista en el artículo 37.2 de la Ley 21/2001, de 27 de diciembre, por la que se regulan las medidas fiscales y administrativas del nuevo sistema de financiación de las Comunidades Autónomas de régimen común y Ciudades con Estatuto de Autonomía, así como la coordinación de la alta inspección referente a la aplicación de los sistemas fiscales concertados o convenidos.

d) La elaboración, análisis y explotación de la información en materia de gestión y actuación de los servicios que resulte precisa para el ejercicio de las funciones que le corresponden

1.2.- Antecedentes Tecnológicos.

La generación de cuadros y gráficas a partir de los datos suministrados por las Comunidades Autónomas se ha venido realizando mediante la herramienta Excel de Microsoft. Los datos se copiaban a ciertas celdas de cada cuadro, para que a su vez los valores calculados se actualizaran. Este sistema es propenso a los errores debido al elevado volumen de datos y de cuadros, y a la facilidad con que se puede confundir el lugar adecuado para pegar los datos copiados. Aparte de la lentitud inherente a este sistema, cualquier modificación obligaba a realizar de nuevo todo o parte del proceso.

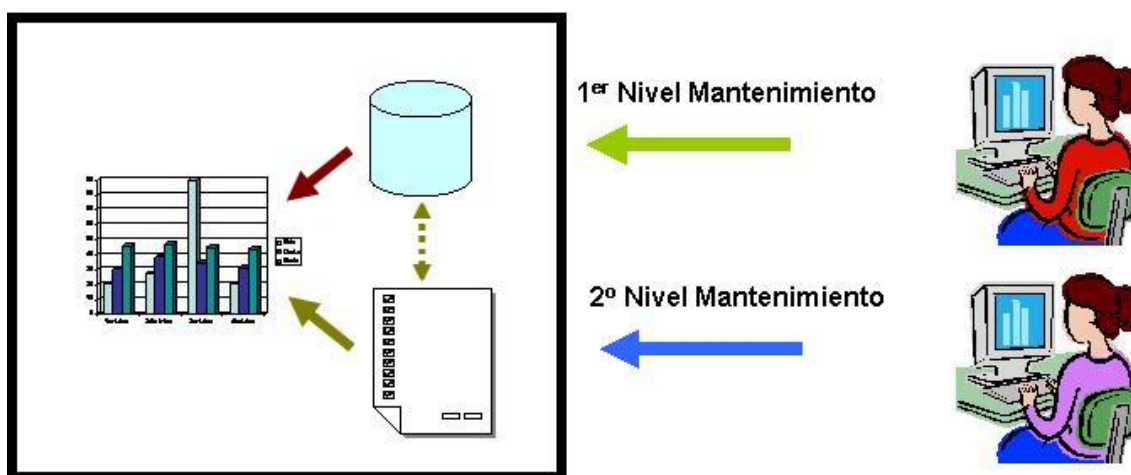
2.- Solución propuesta

A raíz de la introducción de un nuevo sistema de envío de datos basado en una aplicación Web con autenticación y firma digital de la información de gestión de los Tributos Cedidos por parte de las Comunidades Autónomas, se detectó la posibilidad de generar las gráficas y cuadros, según se van completando los envíos.

También se pensó como valor añadido que las comunidades obtuvieran sus estadísticas con respecto a datos referencia de orden nacional de forma inmediata, eliminando el periodo de elaboración previo. Estos datos podrían ser exportados a Excel desde Internet.

Se quería generar un elevado número de cuadros, mantener una alta capacidad para modificar los diseños y añadir nuevos productos con facilidad para el cambiante mundo de los tributos.

Por todo esto se pensó en un motor para la generación de los cuadros con dos niveles distintos e independientes de mantenimiento de respuesta inmediata. El primero utilizaría unas tablas de la base de datos de donde se obtienen literales, estructura de cuadros, tipo de columnas y “queries” a la base de datos. El mantenimiento del primer nivel lo ejecutan auxiliares con cualificación básica. El segundo emplea la técnica de “Patrones de Diseño de Software Orientado a Objetos”, en concreto el patrón denominado “de Estrategia”. El mantenimiento de segundo nivel lo ejecutan programadores con cualificación básica.



3.- Patrón de Diseño de Software Orientado a Objetos. Patrones de Comportamiento. Patrón de Estrategia.

Un Patrón de Diseño es una solución a un problema de diseño no trivial que es efectiva (ya se resolvió el problema satisfactoriamente en ocasiones anteriores) y re-usable (se puede aplicar a diferentes problemas de diseño en distintas circunstancias). Esta definición es plenamente aplicable al Software Orientado a Objetos.

Entre los Patrones de Diseño están los de Comportamiento que estudian las relaciones entre llamadas entre los diferentes objetos, normalmente ligados con la dimensión temporal. Y dentro de estos están los llamados Patrones de Estrategia que permiten algoritmos conectables.

El Patrón de Estrategia se basa en encapsular los aspectos del diseño con más posibilidades de cambio en clases que heredan de una clase abstracta común. Cualquier cambio posterior en los requisitos se puede incorporar de manera sencilla y con un impacto mínimo sobre el resto del sistema, ya que es posible:

- Crear nuevas clases para implementar nuevos comportamientos.
- Modificar las clases que ya existen sin que el resto de las clases se vean afectadas.
- Cambiar el comportamiento en tiempo de ejecución.

4.- Aplicación del Patrón de Diseño de Software Orientado a Objetos en la solución propuesta. 2º Nivel de Mantenimiento.

El diseño que se ha realizado utiliza el Patrón denominado “de Estrategia” que permite aprovechar las ventajas de la Programación Orientada a Objetos de una forma sistemática. De esta manera se pueden añadir nuevas operaciones y modificar las existentes sin afectar al resto del sistema (gracias al bajo acoplamiento del diseño resultante) y sin necesidad de conocer los detalles de cómo están implementadas las demás operaciones (ocultación).

A continuación se describe sucintamente cómo se ha aplicado este Patrón de Diseño en este caso.

Se definen las siguientes clases:

Dentro de la parte probablemente “fija” del diseño:

- Una clase que se encarga de administrar la creación de cuadros, denominada “SQLCuadroAdmin”.
- Una clase “Cuadro” que representa cada tabla de datos o gráfico. Tiene como atributos, entre otros, un identificador y un título. Ofrece un conjunto de columnas.

- Una clase “Columna”. Cada columna tiene uno o varios títulos de cabecera y una serie de datos, que pueden ser obtenidos por lectura directa de la base de datos (tabla Columna_Query) o como resultado de alguna operación entre datos de otras columnas del mismo cuadro (tabla Columna_Calculada).

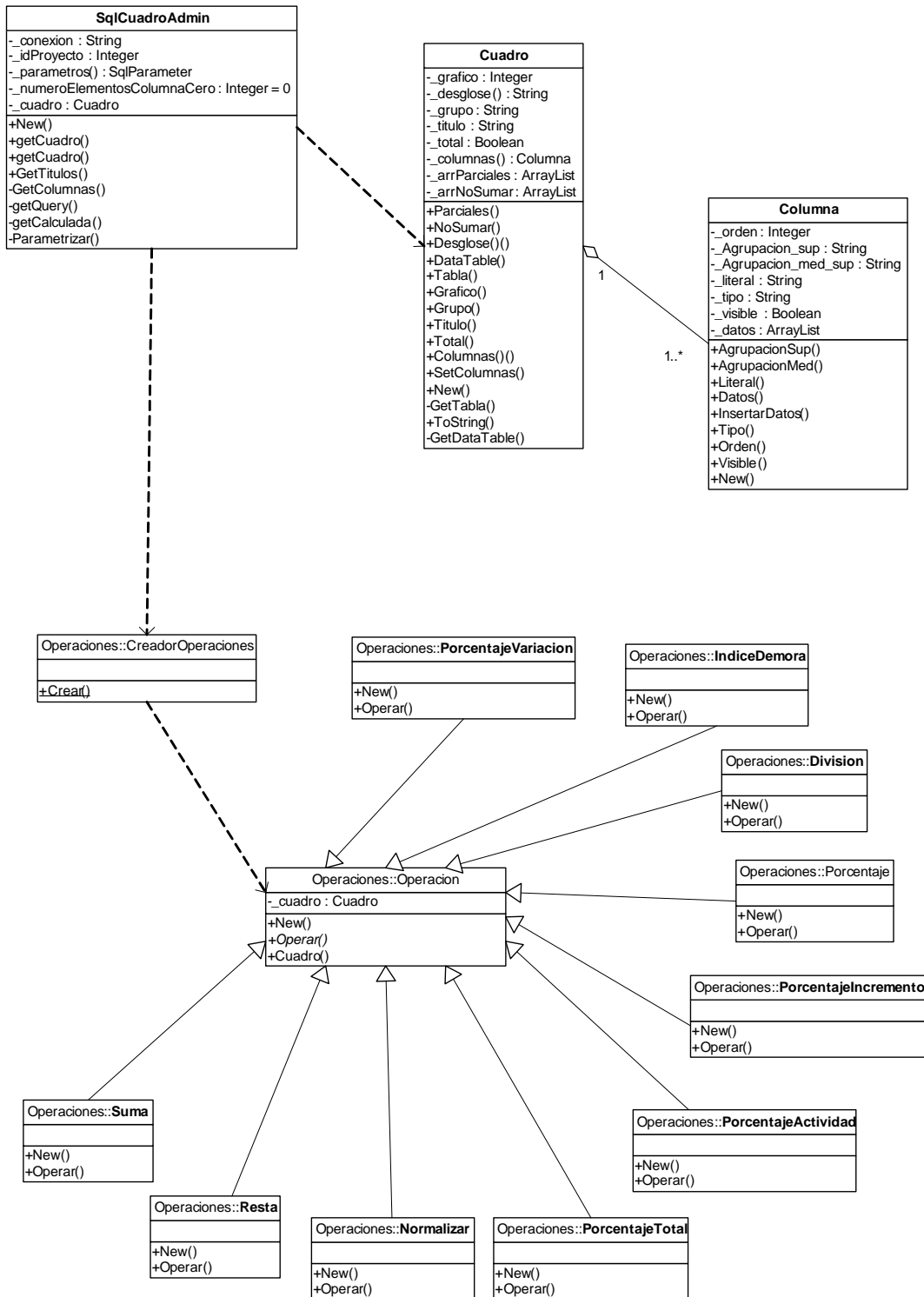
En la base de datos se han diferenciado las columnas en dos tablas distintas en función de su origen: directas y calculadas. Las del primer tipo contienen las instrucciones SQL que recuperarán la serie de datos (se parametrizan para obtener datos por años, provincias y Comunidades). En las del segundo tipo se ha incluido en la tabla la información sobre el tipo de operación, las columnas que son operandos, y el orden en que han de tomarse estos.

Dentro de la parte “variable” del diseño:

- Varias clases que se utilizan en la generación de las columnas de los cuadros y que son resultado de operaciones entre otras. Se utiliza la clase abstracta “Operacion”, de la cual heredan las distintas clases que implementan las operaciones que se han requerido para elaborar las tablas y gráficos. Estas operaciones son:
 - “Suma”
 - “Resta”
 - “División”
 - “Porcentaje”
 - “Normalizar”
 - “Índice de demora”
 - “Porcentaje Total”
 - “Porcentaje de Variación”
 - “Porcentaje de Actividad”
 - “Porcentaje de Incremento”

Se pueden añadir nuevas operaciones entre columnas de datos, por ejemplo se podría crear una nueva operación que fuese la media o la desviación típica sin más que añadir una nueva clase y aprovechando al 100% ventajas de la programación orientada a objetos: ocultación (no es necesario saber cómo se han implementado el resto de operaciones) y encapsulación (para implementar esta nueva operación no es necesario tener acceso al resto del código).

Adicionalmente, es posible modificar una operación ya existente sin que esto tenga ningún efecto sobre las demás, por ejemplo modificar la operación de normalización para adoptar nuevas constantes.



4.- Aplicación a la Solución del Primer Nivel de Mantenimiento.

Es muy difícil explicar con orden lógico esta parte del código si no se ha descrito el Segundo Nivel de Mantenimiento. La razón es que las clases utilizan esta parte del diseño como elementos de información. El diseño de cuadros se ha realizado de forma que toda la información relativa a cómo se componen físicamente, que puede ser de tipo numérico o gráfico, se encuentra en tablas de la base de datos:

Tabla Cuadros:

ID Cuadro	Grupo	Título	Total	Desglose	Tipo
1	MEDIOS PERSONALES	Personal total afecto a la gestión de los Tributos Cedidos en las comunidades que la han asumido	1		TABLA
2	MEDIOS PERSONALES	Medios personales existentes a fin de año en los Servicios Territoriales. Número de efectivos: personal funcionario y contratado	1		TABLA
3	MEDIOS PERSONALES	Distribución de los medios personales existentes a fin de año en los Servicios Territoriales por Comunidades Autónomas y grupos	1		TABLA
4	MEDIOS PERSONALES	Distribución de los medios personales existentes a fin de año en los Servicios Territoriales por Comunidades Autónomas y funciones	1		TABLA
5	MEDIOS PERSONALES	Personal de empresas de servicios y Oficinas Liquidadoras	1		TABLA
6	MEDIOS PERSONALES	Ingresos aplicados por efectivo afecto a la gestión de los Tributos Cedidos en miles de €	1		TABLA
7	MEDIOS PERSONALES	Evaluación de los medios personales en los Servicios Territoriales de las Comunidades Autónomas	0		BARRAS
8	MEDIOS PERSONALES	Distribución del personal afecto a los Servicios Territoriales de las Comunidades Autónomas	0		BARRAS
9	MEDIOS PERSONALES	Distribución porcentual del personal afecto a los Servicios Territoriales de las Comunidades Autónomas	0		COLUMNAS100

Tabla Columnas:

Mediante los campos de Agrupación superior, media y literal se logra agrupar los tres posibles niveles de cabeceras de las columnas según sean los valores (iguales o no):

ID Cuadro	Orden Columna	Tipo	Agrupación superior	Agrupación Media	Literal
1	0	QUERY			Comunidades Autónomas
1	1	QUERY	Oficinas gestoras	Personal del Servicio Territorial	Personal funcionario
1	2	QUERY	Oficinas gestoras	Personal del Servicio Territorial	Personal interino
1	3	QUERY	Oficinas gestoras	Personal del Servicio Territorial	Personal contratado
1	4	CALCULADA	Oficinas gestoras	Personal del Servicio Territorial	Suma
1	5	QUERY	Oficinas gestoras		Personal empresas de servicios
1	6	CALCULADA	Oficinas gestoras		Total
1	7	QUERY			Oficinas liquidadoras
1	8	CALCULADA			Total
2	0	QUERY			Servicio Territorial
2	1	QUERY		Funcionarios del Estado	[@Anio - 1]
2	2	QUERY		Funcionarios del Estado	[@Anio]
2	3	QUERY		Funcionarios de la C. Autónoma	[@Anio - 1]
2	4	QUERY		Funcionarios de la C. Autónoma	[@Anio]

Tabla Columna_Calculada:

ID Cuadro	Orden Columna	Operación
1	4	SUMA
1	6	SUMA
1	8	SUMA
2	7	SUMA
2	8	SUMA
2	11	SUMA

2	12	SUMA
3	11	SUMA
3	23	SUMA
3	25	SUMA
4	17	SUMA
5	5	SUMA
5	6	SUMA
6	1	RESTA
6	3	DIVISION
6	6	DIVISION
6	8	SUMA
6	9	DIVISION
8	3	SUMA
13	2	PORCENTAJE_INCREMENTO
13	4	PORCENTAJE_INCREMENTO
13	5	DIVISION
13	6	PORCENTAJE_INCREMENTO
14	2	PORCENTAJE_INCREMENTO

Tabla Columna_Query:

ID Cuadro	Orden Columna	Query
1	0	SELECT tb_Comunidades.comunidad FROM tb_C1_Personal INNER JOIN tb_Comunidades ON tb_C1_Personal.CodDel = tb_Comunidades.DH WHERE tb_C1_Personal.Anio = @Anio GROUP BY tb_Comunidades.CA, tb_Comunidades.comunidad
1	1	SELECT SUM(tb_C1_Personal.TOT_FUESTOAD_2 + tb_C1_Personal.TOT_FUCCAA_2) FROM tb_C1_Personal INNER JOIN tb_Comunidades ON tb_C1_Personal.CodDel = tb_Comunidades.DH WHERE tb_C1_Personal.Anio = @Anio GROUP BY tb_Comunidades.CA
1	2	SELECT SUM(TOT_OPFU_2) FROM tb_C1_Personal INNER JOIN tb_Comunidades ON tb_C1_Personal.CodDel = tb_Comunidades.DH WHERE tb_C1_Personal.Anio = @Anio GROUP BY tb_Comunidades.CA
1	3	SELECT SUM(TOT_PCLA_2) FROM tb_C1_Personal INNER JOIN tb_Comunidades ON tb_C1_Personal.CodDel = tb_Comunidades.DH WHERE tb_C1_Personal.Anio = @Anio GROUP BY tb_Comunidades.CA
1	5	SELECT SUM(TOTEMPSE_2) FROM tb_C1_Personal INNER JOIN tb_Comunidades ON tb_C1_Personal.CodDel = tb_Comunidades.DH WHERE tb_C1_Personal.Anio = @Anio GROUP BY tb_Comunidades.CA
1	7	SELECT SUM(TOT_OFLIQ_2) FROM tb_C1_Personal INNER JOIN tb_Comunidades ON tb_C1_Personal.CodDel = tb_Comunidades.DH WHERE tb_C1_Personal.Anio = @Anio GROUP BY tb_Comunidades.CA
2	0	SELECT tb_Comunidades.comunidad FROM tb_C1_Personal INNER JOIN tb_Comunidades ON tb_C1_Personal.CodDel = tb_Comunidades.DH WHERE tb_C1_Personal.Anio = @Anio GROUP BY tb_Comunidades.CA, tb_Comunidades.comunidad
2	1	SELECT SUM(tb_C1_Personal.TOT_FUESTOAD_2) FROM tb_C1_Personal INNER JOIN tb_Comunidades ON tb_C1_Personal.CodDel = tb_Comunidades.DH WHERE tb_C1_Personal.Anio = @Anio - 1 AND tb_Comunidades.CA IN (SELECT tb_Comunidades.CA FROM tb_C1_Personal INNER JOIN tb_Comunidades ON tb_C1_Personal.CodDel = tb_Comunidades.DH WHERE tb_C1_Personal.Anio = @Anio-1 GROUP BY tb_Comunidades.CA) GROUP BY tb_Comunidades.CA
2	2	SELECT SUM(tb_C1_Personal.TOT_FUESTOAD_2) FROM tb_C1_Personal INNER JOIN tb_Comunidades ON tb_C1_Personal.CodDel = tb_Comunidades.DH WHERE tb_C1_Personal.Anio = @Anio GROUP BY tb_Comunidades.CA
2	3	SELECT SUM(tb_C1_Personal.TOT_FUCCAA_2) FROM tb_C1_Personal INNER JOIN tb_Comunidades ON tb_C1_Personal.CodDel = tb_Comunidades.DH WHERE tb_C1_Personal.Anio = @Anio - 1 AND tb_Comunidades.CA IN (SELECT tb_Comunidades.CA FROM tb_C1_Personal INNER JOIN tb_Comunidades ON tb_C1_Personal.CodDel = tb_Comunidades.DH WHERE tb_C1_Personal.Anio = @Anio GROUP BY tb_Comunidades.CA) GROUP BY tb_Comunidades.CA

2	4	SELECT SUM(tb_C1_Personal.TOT_FUCCAA_2) FROM tb_C1_Personal INNER JOIN tb_Comunidades ON tb_C1_Personal.CodDel = tb_Comunidades.DH WHERE tb_C1_Personal.Anio = @Anio GROUP BY tb_Comunidades.CA
2	5	SELECT SUM(tb_C1_Personal.TOT_OPFU_2) FROM tb_C1_Personal INNER JOIN tb_Comunidades ON tb_C1_Personal.CodDel = tb_Comunidades.DH WHERE tb_C1_Personal.Anio = @Anio-1 GROUP BY tb_Comunidades.CA
2	6	SELECT SUM(tb_C1_Personal.TOT_OPFU_2) FROM tb_C1_Personal INNER JOIN tb_Comunidades ON tb_C1_Personal.CodDel = tb_Comunidades.DH WHERE tb_C1_Personal.Anio = @Anio GROUP BY tb_Comunidades.CA
2	9	SELECT SUM(tb_C1_Personal.TOT_PCLA_2) FROM tb_C1_Personal INNER JOIN tb_Comunidades ON tb_C1_Personal.CodDel = tb_Comunidades.DH WHERE tb_C1_Personal.Anio = @Anio - 1 AND tb_Comunidades.CA IN (SELECT tb_Comunidades.CA FROM tb_C1_Personal INNER JOIN tb_Comunidades ON tb_C1_Personal.CodDel = tb_Comunidades.DH WHERE tb_C1_Personal.Anio = @Anio GROUP BY tb_Comunidades.CA) GROUP BY tb_Comunidades.CA
2	10	SELECT SUM(tb_C1_Personal.TOT_PCLA_2) FROM tb_C1_Personal INNER JOIN tb_Comunidades ON tb_C1_Personal.CodDel = tb_Comunidades.DH WHERE tb_C1_Personal.Anio = @Anio GROUP BY tb_Comunidades.CA

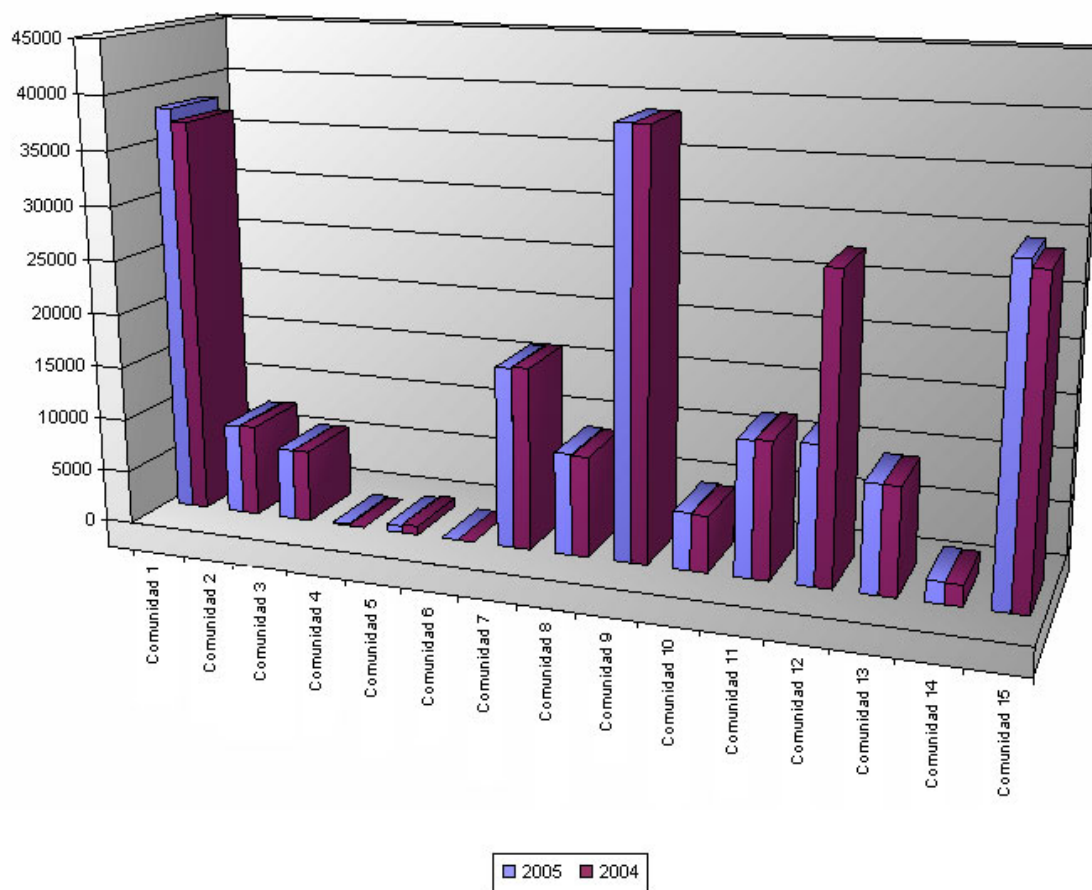
Tabla Operandos:

ID Cuadro	Orden Columna	Operando	Orden operación
1	4	1	1
1	4	2	2
1	4	3	3
1	6	4	1
1	6	5	2
1	8	6	2
1	8	7	3
2	7	1	1
2	7	3	2
2	7	5	3
2	8	2	1
2	8	4	2
2	8	6	2
2	11	7	1
2	11	9	2
2	12	8	1
2	12	10	2

El resultado son cuadros como se puede ver en la siguiente imagen (los datos son ficticios):

Cuadro n° 1
Personal total afecto a la gestión de los Tributos Cedidos en las comunidades que la han asumido *

Comunidades Autónomas	Oficinas gestoras						Oficinas liquidadoras	Total
	Personal del Servicio Territorial				Personal de empresas de servicios	Total		
	Personal funcionario	Personal interino	Personal contratado	Suma				
Comunidad 1	218	58	24	300	2	302	104	405
Comunidad 2	40	6	2	48	5	52	20	72
Comunidad 3	16	4	0	20	3	23	14	38
Comunidad 4	27	3	0	31	2	32	17	49
Comunidad 5	33	0	10	43	15	59	37	95
Comunidad 6	13	0	1	14	2	16	10	26
Comunidad 7	166	22	20	208	0	208	68	276
Comunidad 8	88	29	4	121	0	121	62	182
Comunidad 9	174	33	8	215	10	224	207	431
Comunidad 10	50	14	2	67	0	67	28	95
Comunidad 11	167	22	19	208	2	209	71	281
Comunidad 12	29	6	5	40	7	47	10	57
Comunidad 13	11	2	2	15	6	21	5	26
Comunidad 14	4	1	0	4	0	5	1	5
Comunidad 15	16	17	3	35	0	35	32	68
Total	847	208	91	1.146	75	1.221	560	1.781



5.- Resumen

En esta solución tecnológica responde a la conveniencia y necesidad de seguir una de las filosofías de desarrollo más actuales en el complejo mundo de las grandes aplicaciones informáticas como son los Patrones de Desarrollo de Software Orientado a Objetos.

Como resultado final, se consigue un potente motor conformado por clases cuya versatilidad hace posible un mantenimiento de las aplicaciones rápido y fácil. Además, el motor facilita la ampliación y adaptabilidad al moderno, variado y cambiante Sistema Tributario Español haciendo que la información surja con un menor esfuerzo y manteniéndose unos estándares altos de calidad en el desarrollo de aplicaciones informáticas.