



**UNIVERSIDAD DE CHILE
FACULTAD DE CIENCIAS FÍSICAS Y MATEMÁTICAS
DEPARTAMENTOS DE CIENCIAS DE LA COMPUTACIÓN**

**DISEÑO E IMPLEMENTACIÓN DE UN SISTEMA DE APOYO A LA
GENERACIÓN DE ESTADÍSTICAS E INDICADORES.**

**MEMORIA PARA OPTAR AL TÍTULO DE INGENIERO CIVIL
EN COMPUTACION**

FRANCISCO JAVIER CESPEDES VIÑUELA

**PROFESOR GUIA:
JULIO ANTONIO SALAS MANZUR**

**MIEMBROS DE LA COMISION:
JUAN FERNANDO ALVAREZ RUBIO
CLAUDIO GUTIERREZ GALLARDO**

**SANTIAGO DE CHILE
ENERO 2009**

RESUMEN DE LA MEMORIA
PARA OPTAR AL TITULO DE
INGENIERO CIVIL EN COMPUTACION
POR: FRANCISCO CESPEDES V.
FECHA: 22/01/2009
PROF. GUIA: Sr. JULIO SALAS MANZUR

“DISEÑO E IMPLEMENTACIÓN DE UN SISTEMA DE APOYO A LA GENERACIÓN DE ESTADÍSTICAS E INDICADORES”

Este trabajo aborda el problema de desarrollar una aplicación que permita implementar índices e informes estadísticos de apoyo a la gestión docente de la Escuela de Ingeniería y Ciencias de la Universidad de Chile, de una manera sencilla y escalable utilizando tecnología Web. Como resultado de este trabajo se obtuvo una aplicación denominada “Sistema de apoyo a la generación de reportes SIAGER”.

La utilización de un modelo de datos ROLAP estrella permitió responder de manera satisfactoria tanto a las consultas sencillas como a las que implican una mayor carga. Al comparar SIAGER con los resultados obtenidos por el antiguo sistema SIESTA, se observaron reducciones de al menos un cincuenta por ciento tanto en el espacio utilizado en disco como en los tiempos de respuesta a las consultas.

La solución se desarrolló en varias etapas. Primero se realizó el análisis de la situación actual, tanto de las herramientas utilizadas internamente por la Escuela de Ingeniería y Ciencias, como las disponibles en el mercado. Paralelamente, se realizaron entrevistas con los usuarios que solicitan y generan indicadores e informes estadísticos, a fin de determinar los procesos y la información frecuentemente requerida. La segunda etapa consistió en el diseño e implementación de una herramienta de procesamiento de consultas en línea, para mejorar el control y la eficiencia en los estudios y proyecciones realizadas.

SIAGER permite realizar consultas de acuerdo a condiciones de filtrado definidas por el usuario, además de generar gráficos e informes finales. Las consultas efectuadas pueden ser almacenadas, tanto en el sistema, como en un archivo que los usuarios pueden descargar en su disco. Los resultados de las consultas son desplegados en una interfaz por pantalla junto a los gráficos asociados, y pueden ser exportados a los formatos Word, RTF, Excel y PDF. El sistema permite crear y gestionar los modelos de datos ROLAP estrella a través de una interfaz que le facilita al usuario su definición.

Finalizado el trabajo, se pudo comprobar la simplicidad para crear nuevos informes e indicadores estadísticos a partir de diferentes fuentes de información y el potencial que ofrece a los altos niveles de dirección estratégica. Actualmente la aplicación se encuentra en un ambiente de preproducción y se espera que sea llevada próximamente a producción.

**Quiero agradecer a mi familia
por su apoyo incondicional en
esta etapa que termina.**

Índice de Contenidos

1	Introducción	6
1.1	Justificación del problema	7
1.2	Objetivos	8
1.3	Metodología	9
1.3.1	Definición de requerimientos	9
1.3.2	Diseño de una solución factible	9
1.3.3	Prototipo de la solución	9
1.3.4	Estudio de la plataforma U-Campus	9
1.3.5	Implementar la solución	9
1.3.6	Pruebas e implantación en producción	9
2	Marco conceptual	10
2.1	Business Intelligence	10
2.2	Bases de datos relacionales	10
2.3	Bases de datos multidimensionales	11
2.4	Tecnologías asociadas	15
3	Análisis de la situación actual	17
3.1	Catastro de bases de datos	17
3.2	Entidades y usuarios que requieren información	18
3.3	Herramientas	19
3.4	Reportes estadísticos e indicadores frecuentes	32
4	Requerimientos	34
4.1	Requerimientos de información	34
4.2	Requerimientos funcionales	34
4.3	Requerimientos no-funcionales	34
4.4	Requerimientos de interfaces	35
4.5	Requerimientos operacionales	35
4.6	Requerimientos tecnológicos	35
5	Diseño	36
5.1	Capa de datos	36
5.2	Capa lógica	42
5.3	Capa de presentación	59
5.4	Modelos ROLAP estrella propuestos para reportes frecuentes	59
6	Implementación	73
6.1	Archivos template	73
6.2	Scripts del sistema	74
6.3	Funciones	75
7	Interfaces y funcionalidades	95
7.1	Módulo de inicio	95
7.2	Módulo de Consulta	95
7.3	Módulo de creación de un reporte	111
7.4	Módulo de campos visibles	115
8	Pruebas del sistema e implantación	116
8.1	Pruebas del sistema	116
8.2	Implantación en producción	129
9	Conclusiones	131
9.1	Contribuciones	131
9.2	Limitaciones	132

9.3 Trabajos futuros	133
10 Referencias	134
10.1 Definiciones:	134
10.2 Documentos:	135
10.3 Librerías:	135
10.4 Libros:	136
10.5 Memorias:	136
10.6 MySQL:	136
10.7 Paper's:	138
10.8 Sitios Web:	139
11 Anexos	142
11.1 Entrevistas	142
11.2 Librerías	152
11.3 Análisis de opciones PACK_KEYS	159

1 Introducción

La Escuela de Ingeniería y Ciencias durante la última década se ha dedicado a desarrollar y mejorar un conjunto de aplicaciones cuyo principal propósito es apoyar sus procesos operativos, optimizándolos, disminuyendo los tiempos de procesos y mejorando la calidad de la información entregada a sus estudiantes y profesores.

Entre los procesos que ha logrado llevar a sistemas computacionales se encuentran el sistema de Encuesta Docente, que está operando desde el año 2001, el sistema de Administración Docente, que opera desde el año 1970 y que comprende los procesos de Admisión, Inscripción Académica, Catálogo de Cursos, Solicitudes, Actas de Exámenes, entre otros. Además se cuenta con información histórica de los resultados obtenidos por la prueba de selección universitaria (PSU).

Luego de un importante desarrollo de sistemas de apoyo a los procesos administrativos, la Escuela centró sus esfuerzos en el desarrollo de info-tecnologías de apoyo a los procesos de enseñanza-aprendizaje, creándose así la plataforma U-Cursos que concentra la mayor parte de estos servicios. Otros servicios de esta naturaleza los constituyen la escuela móvil, el foro institucional, entre otros.

Por otro lado, la Facultad de Ciencias Físicas y Matemáticas ha desarrollado un conjunto de sistemas que son de interés de la Escuela, como son: Sistema de registro de nombramientos, que contienen la jerarquía, antigüedad y grado de cada profesor, el sistema U-Curriculum que contiene diversa información de interés como los títulos profesionales, grados, etc.

Por su parte, los servicios centrales de la Universidad de Chile administran el sistema de matrícula para todos los estudiantes y cuya información no se encuentra disponible con la facilidad y oportunidad requerida por la Escuela para sus diversas necesidades de manejo de información.

Todos estos sistemas y servicios operan en servidores ubicados en distintas dependencias, con administradores independientes, con sistemas operativos y de bases de datos distintos y han acopiado una cantidad de información significativa. Esta diversidad en la forma de administrar información, sumado a los altos volúmenes de datos contenidos, ha producido importantes dificultades en el manejo de información y, en especial, en la generación de informes estadísticos e indicadores de gestión que requiere la Escuela para tomar decisiones de carácter estratégico en el ámbito de la docencia y de su gestión en general.

Entre las estadísticas e indicadores frecuentes y que son de costosa elaboración según los encargados de generarlas, se encuentran:

- Estadísticas e indicadores sobre ingreso de alumnos nuevos, tales como número de alumnos matriculados de Primer Año, distribuidos por región de procedencia, tipo de establecimiento educacional, género, tipo de ingreso (PSU, deportista, extranjeros, etc.), puntaje promedio ponderado de ingreso, puntajes promedio por prueba PSU, promedios de notas de enseñanza media, distribución etárea de los alumnos, aranceles, índices de retención, indicadores de rendimiento, entre otros aspectos por determinar.

- Estadísticas e indicadores sobre Docentes, tales como el número de docentes por tipo de jornada, número de horas contratadas promedio por tipo de jornada, número de docentes por grado académico, número de horas contratadas por grado académico, docencia directa por grado académico, número de docentes por departamento, distribución etárea por departamento y grado académico, género, entre otros aspectos por determinar.
- Estadísticas e indicadores sobre Estudios de evolución de cohortes de todas las carreras y por departamento, tales como el número de alumnos matriculados, tasas de retención, número de egresados, número de titulados, tasas de avance en la carrera, tasas de egreso, tasas de titulación, género, entre otros aspectos por determinar.

En este contexto, la Escuela de Ingeniería y Ciencias ha tomado la decisión de emprender el desarrollo de un sistema que le permita generar reportes estadísticos e indicadores de una manera sencilla, oportuna y de calidad.

1.1 Justificación del problema

El presente tema nace de la preocupación de la Escuela de Ingeniería y Ciencias por actualizar y mejorar su proceso de generación de informes estadísticos e indicadores. El proceso actual es complejo y propenso a errores, esto se debe principalmente a que gran parte de la obtención de los datos se realiza a través de la utilización de sistemas que requieren conocimientos de SQL o bases de datos, digitación humana y no existen controles adecuados del proceso.

Además, para poder generar ciertos informes, primero es necesario conseguir los datos con los respectivos Administradores de los sistemas involucrados, subirlos a algún repositorio para después diseñar las consultas que permitan cruzar dicha información con el resto de los datos disponibles localmente. Normalmente, estos resultados son entregados a un usuario, en un formato de planilla electrónica, para que este realice otra serie de cálculos y tareas de formatos antes de obtener el informe final requerido.

Estas dificultades para la generación de informes estadísticos e indicadores hacen imprescindible e impostergable contar con una herramienta tecnológica que permita sistematizar y resolver este tipo de tareas de una manera más sencilla, oportuna y con los estándares de calidad requeridos por la Escuela.

El desarrollo de una aplicación que permita realizar las acciones anteriormente descritas mejorará significativamente la gestión y el desarrollo de la Escuela, a través de un mejor control y eficiencia para realizar estudios, proyecciones y planes estratégicos. Por otra parte, los funcionarios encargados de generar dicha información, podrán dedicarse a otras tareas.

Pese a existir una gran cantidad de herramientas de procesamiento analítico en línea y de análisis de grandes bases de datos o Data Warehouse, todas ellas deben enfrentar, como algunos de sus principales problemas, la implementación eficiente, tiempos de respuestas y rendimiento aceptables frente a las consultas. A pesar del gran número de estas herramientas y de contar con una Web desarrollada, son pocas las aplicaciones que utilizan el Browser como medio de comunicación con el usuario. La entrada del procesamiento analítico en línea en la Web ha sido lenta y el desarrollo de estas aplicaciones está orientada en su mayoría a computadores con sistema operativo Windows en modalidad cliente-servidor.

Las herramientas de procesamiento analítico en línea presentan más complicaciones que la mayoría del software común, ya que incluyen tanto a usuarios finales como a Administradores. No es de extrañar que muchas aplicaciones estén orientadas a estos últimos por sobre los usuarios finales, debido al conocimiento técnico que involucra. Los usuarios finales en muchos casos son incapaces o reacios a especificar sus necesidades detalladamente, disponen de poco tiempo para aprender herramientas de software y desconocen sobre la migración, transformación y transferencia de datos entre múltiples fuentes de información. Este problema se agrava si cada una de estas fuentes tiene sus propios problemas e incompatibilidades.

Dentro de los desafíos en el diseño de una herramienta de esta naturaleza, se encuentra el resolver de una manera estándar la captura de información desde diversos administradores de bases de datos, su acopio de una manera eficiente y el desarrollo de funcionalidades que faciliten la generación de informes estadísticos e indicadores de una manera rápida y de bajo costo.

Este trabajo se enmarca en el área de *Ingeniería de Software*, debiéndose cumplir con todas las etapas de un desarrollo normal de software. Algunas de estas etapas son el análisis de requerimientos, el desarrollo de prototipos, validación de funcionalidades, pruebas, entre otras.

1.2 Objetivos

1.2.1 Objetivo general

Diseño y construcción de un sistema computacional que facilite la generación y consulta de indicadores estadísticos, de eficiencia y de control de gestión para la Escuela de Ingeniería y Ciencias.

1.2.2 Objetivos específicos

- Analizar el proceso actual de recopilación de información estadística para su estudio y comprensión de manera de determinar las principales fallas y sus soluciones.
- Desarrollar una aplicación que permita implementar índices e informes estadísticos de una manera sencilla y escalable, utilizando tecnología Web, considerando diversas fuentes de información.
- Desarrollar una aplicación que permita implementar índices e informes estadísticos sobre ingreso de alumnos nuevos, docentes y estudios de evolución de cohortes.
- Diseñar el sistema de tal manera de minimizar los costos de implementación de nuevos informes e índices.
- Proporcionar un sistema que apoye con información relevante a los altos niveles de dirección estratégica.
- Disminuir los tiempos de respuesta y error durante el desarrollo del proceso de generación de indicadores y reportes estadísticos.
- Mejorar la calidad del servicio y facilitar el control de gestión de la Escuela.

1.3 Metodología

En este punto se explican los pasos seguidos en el desarrollo de este trabajo, los cuales van desde el análisis de requerimientos hasta la implantación del sistema. Entre los pasos seguidos se incluye el análisis de la aplicación utilizada para generar reportes estadísticos e indicadores, cuyo nombre es SIESTA.

1.3.1 Definición de requerimientos

Basándose en las entrevistas y el estudio de la situación actual es posible determinar los requerimientos que la aplicación a desarrollar debe cumplir. Estos requerimientos consideran aspectos operacionales, funcionales, no-funcionales, de interfaz, tecnológicos y de información requerida.

1.3.2 Diseño de una solución factible

Estudiar el modelo de datos y el código fuente de la aplicación SIESTA, de manera de detectar errores en la lógica o en su implementación, evitándolos en la nueva implementación. También permite entender y re-utilizar de mejor forma los archivos PHP que contienen la lógica y las reglas del sistema SIESTA.

1.3.3 Prototipo de la solución

Generar un prototipo de la solución diseñada en el paso anterior, para validar las funcionalidades a desarrollar y los requerimientos. Se debe seguir el estilo del sitio U-Campus, plataforma utilizada ampliamente en la Facultad.

1.3.4 Estudio de la plataforma U-Campus

Entender el estilo de codificación del Área de Info-tecnologías (ADI) para programar según sus reglas y normas. También es necesario aprender sobre el desarrollo de módulos para el sitio U-Campus.

1.3.5 Implementar la solución

En esta etapa se deberá implementar la solución utilizando el conocimiento adquirido en las etapas anteriores. Además, se debe profundizar en el conocimiento de las herramientas y lenguajes de programación sobre los cuales se desarrolla la aplicación.

1.3.6 Pruebas e implantación en producción

Probar con pruebas de aceptación y unitarias que la solución responde a los requerimientos y necesidades. Posterior a las pruebas, y luego que los errores y bugs han sido corregidos, el sistema puede ser llevado a un ambiente de producción.

2 Marco conceptual

En este capítulo se profundiza en los conceptos teóricos necesarios para comprender el tema abordado en este trabajo, a través de la definición de los términos clave y la descripción del modelo de datos utilizado en la aplicación desarrollada.

2.1 Business Intelligence

El concepto de Business Intelligence surge de la necesidad de las empresas de realizar análisis de información, que les permitan tomar decisiones informadas y así definir direcciones de avance o cambios. Los sistemas de inteligencia de negocios tienen una importancia de peso al interior de las empresas, por cuanto el éxito o fracaso de estas depende de su velocidad para analizar y sintetizar la información con que cuentan.

Algunos conceptos relevantes a considerar son:

- **Cubos OLAP o híper cubos:** Término acuñado por EF Codd & Associates en el paper titulado “Providing OLAP to User-Analysts: An TI Mandate” (1993). Consiste en disponer los datos en vectores o cubos que permitan un análisis rápido de grandes volúmenes de datos, que es una de las limitaciones de las bases de datos relacionales tradicionales.
- **OLAP:** Es el acrónimo en inglés de Procesamiento Analítico En Línea (On Line Analytical Processing). Su objetivo es agilizar las consultas en grandes volúmenes de información, constituyéndose en un complemento a las herramientas actuales de administración.

Permite satisfacer de manera rigurosa el tipo de análisis de datos que la empresa necesita en ámbitos tan diversos como ventas, marketing, informes, etc. La información histórica de la empresa es sintetizada y consolidada, para posteriormente ser almacenada separadamente de los datos históricos.

- **OLTP:** Es el acrónimo en inglés de Procesamiento de Transacciones En Línea (On Line Transaction Processing). Es un tipo de sistema que facilita y administra aplicaciones transaccionales, para entrada de datos, recuperación y/o procesamiento de transacciones.
Este tipo de software se basa en una arquitectura cliente-servidor que permite a las transacciones correr en diferentes redes. Sus ventajas son la eficiencia y ejecución más rápida de los procesos individuales, la ampliación de los usuarios y una mejor estabilidad producto de una oportuna actualización. Sus desventajas son los altos costos económicos asociados a mantención y fallas, y los riesgos asociados a la seguridad.

2.2 Bases de datos relacionales

El modelo de bases de datos relacional es el más utilizado en la actualidad. Fue postulado en 1970 por Edgar Frank Codd, su idea fundamental es el uso de relaciones entre entidades o tablas discretas normalizadas.

Previo al trabajo realizado por EF Codd, las bases de datos eran colecciones de ideas ad hoc de sistemas caseros. El objetivo inicial en el desarrollo del modelo relacional fue hacer frente a las deficiencias de los sistemas de finales de la década de los sesenta y abrir un camino para futuros desarrolladores, cuyas aplicaciones y herramientas aumentarían con el tiempo.

En principio, el modelo relacional fue utilizado por aplicaciones que necesitaban recuperar, actualizar, eliminar y guardar uno o múltiples elementos de datos para procesos operacionales y transaccionales. Posteriormente se extendieron para llegar a ser una solución de propósito general. Sin embargo, tiene un desempeño lento para consultas multi-tablas o con un alto grado de agregación.

2.3 Bases de datos multidimensionales

Las bases de datos multidimensionales consisten en disponer la información en vectores o cubos que permitan realizar un análisis rápido de la información. Surge como una mejora al modelo relacional, cuyo desempeño y diseño físico se ve mermado por consultas que requieren un alto grado de agregación o consolidación de la información. El gran volumen de información también constituye una debilidad de los modelos relacionales, así es como en la década de los ochenta las empresas ya manejaban volúmenes del orden de megas y gigabytes. Actualmente estos volúmenes han crecido hasta los teras y petabytes.

Algunas de las ventajas de este modelo son las siguientes:

- Es intuitivo y fácil de entender. Señala explícitamente qué dimensiones se relacionan con cuál tabla de hechos; esta característica puede mejorar si se restringe la unión entre dimensiones o entre tablas de hechos. Para un usuario resultará sencillo comprender la relación entre una única tabla de hechos que referencia varias dimensiones, las cuales son “campos” o “filtros” por los cuales desearía resumir, agregar o filtrar la información, como por ejemplo Tiempo, Promoción, Producto, Antecedentes demográficos, etc.
- Representa de manera explícita cuáles son las tablas relevantes. En un modelo multidimensional, las tablas de hechos pueden ser consideradas las de mayor importancia, ya que relacionan todas las dimensiones y sobre ellas se realizan cálculos como suma, desviación estándar, varianza, promedio, etc.
- Capacidad de manejar grandes volúmenes de información. Las dimensiones son el punto de entrada a la tabla de hechos, al realizar un filtro en alguna dimensión se seleccionan las claves que satisfacen estas condiciones, para posteriormente realizar las uniones en la tabla de hechos y la recuperación de los campos desde otras dimensiones. Si se considera que el número de tuplas de una dimensión es geoméricamente más pequeño que la tabla de hechos y la utilización de índices tanto en las dimensiones como en los hechos, el modelo cumple efectivamente con su cometido.

2.3.1 Conceptos de modelos de bases de datos multidimensionales

- **Browsing:** Se refiere a examinar y obtener los diferentes valores que puede tomar un campo o columna en una tabla determinada.
- **Cardinalidad:** Número de valores únicos que puede tomar una columna en una tabla.
- **Claves subrogadas** (Sinónimos: Clave entera, clave no-natural, clave artificial, clave sintética): Se refiere a claves enteras que son enumeradas secuencialmente en demanda durante el poblado de las dimensiones, así es como la primera tupla insertada en una dimensión tendrá asignada generalmente el valor “1” como clave.
- **Dimensiones conformadas:** Las dimensiones están conformadas cuando son las mismas o una es subconjunto de otra. El resultado de dos dimensiones conformadas deben coincidir. Dimensiones conformadas pueden ser dimensiones compartidas o vistas, o bien copias de tablas con una administración que garantice su conformidad.
- **Dimensión degenerada:** Una dimensión en la forma de clave, sin una tabla dimensión propiamente tal. Generalmente se utiliza para números de facturas o tickets, que no tienen más atributos que el número en sí.
- **Granularidad:** En una tabla de hechos corresponde al nivel en el cual toma un valor único.
- **Hechos aditivos:** Se refiere a atributos de las tablas de hechos que pueden ser sumados a través de todas las dimensiones. Un ejemplo común es la renta o sueldo de un trabajador o cualquier valor en dólares o pesos.
- **Hechos semi-aditivos:** Similar al anterior, pero solo pueden ser sumados a través de algunas dimensiones.
- **Hechos no-aditivos:** Se refiere a atributos de las tablas de hechos que no pueden ser sumados a través de ninguna dimensión. Ejemplos de este tipo de campos son los radios o porcentajes.
- **Mini-dimensiones:** Subconjunto de otra dimensión de gran volumen. Esta separación se realiza para controlar tanto el crecimiento exponencial de las dimensiones como sus cambios.
- **Table engine:** Se refiere al tipo de la tabla creada.

2.3.2 Modelo ROLAP

Acrónimo de procesamiento analítico en línea relacional. Esta variante está diseñada para analizar los datos a través del uso de modelos de datos multidimensionales, pero implementado sobre un modelo de bases de datos relacional. Los esquemas más comunes son **estrella** y **copo de nieve**.

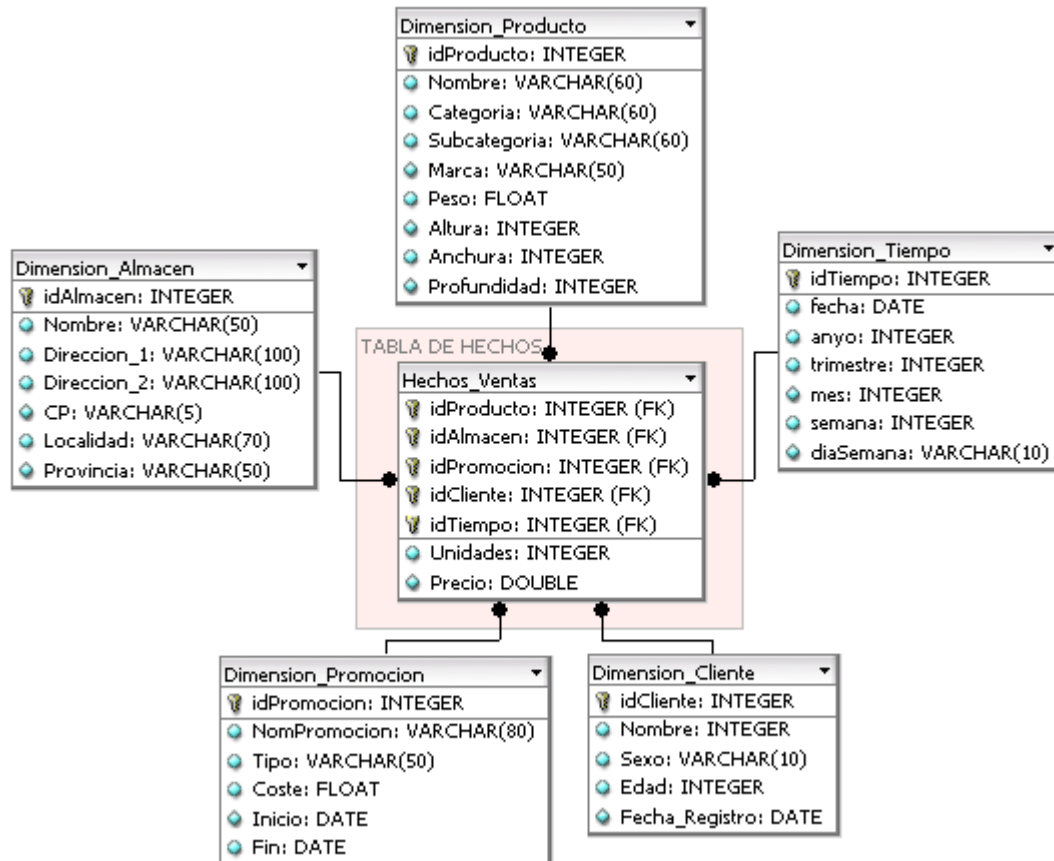


Fig 2.1 Esquema de estrella (Fuente: Wikipedia)

El diseño estrella incluye la creación de una tabla intermedia adicional llamada **tabla de hechos o resumen**, la cual resume la historia de una **magnitud**, por ejemplo las ventas en un determinado periodo. La tabla de hechos está conformada por una serie de relaciones a diferentes tablas, donde cada una almacena la información de los parámetros o **dimensiones del cubo**. En otras palabras, los hechos corresponden a relaciones uno-a-muchos o muchos-a-muchos entre diferentes dimensiones.

La **cardinalidad** o el número de tuplas de la tabla de hechos no será mayor a la suma de las cardinalidades de las tablas de **dimensiones**. Es importante señalar que las tablas de dimensiones cuentan con un número de tuplas geoméricamente menor que la tabla de hechos.

El diseño de **copo de nieve** corresponde a jerarquizar una tabla, añadiendo una nueva tabla relacionada con alguna dimensión de manera de normalizar la información de la dimensión. Sin embargo, este esquema de tabla cuenta con ventajas para bases de datos OLTP pero disminuye el rendimiento de consultas multi-tablas.

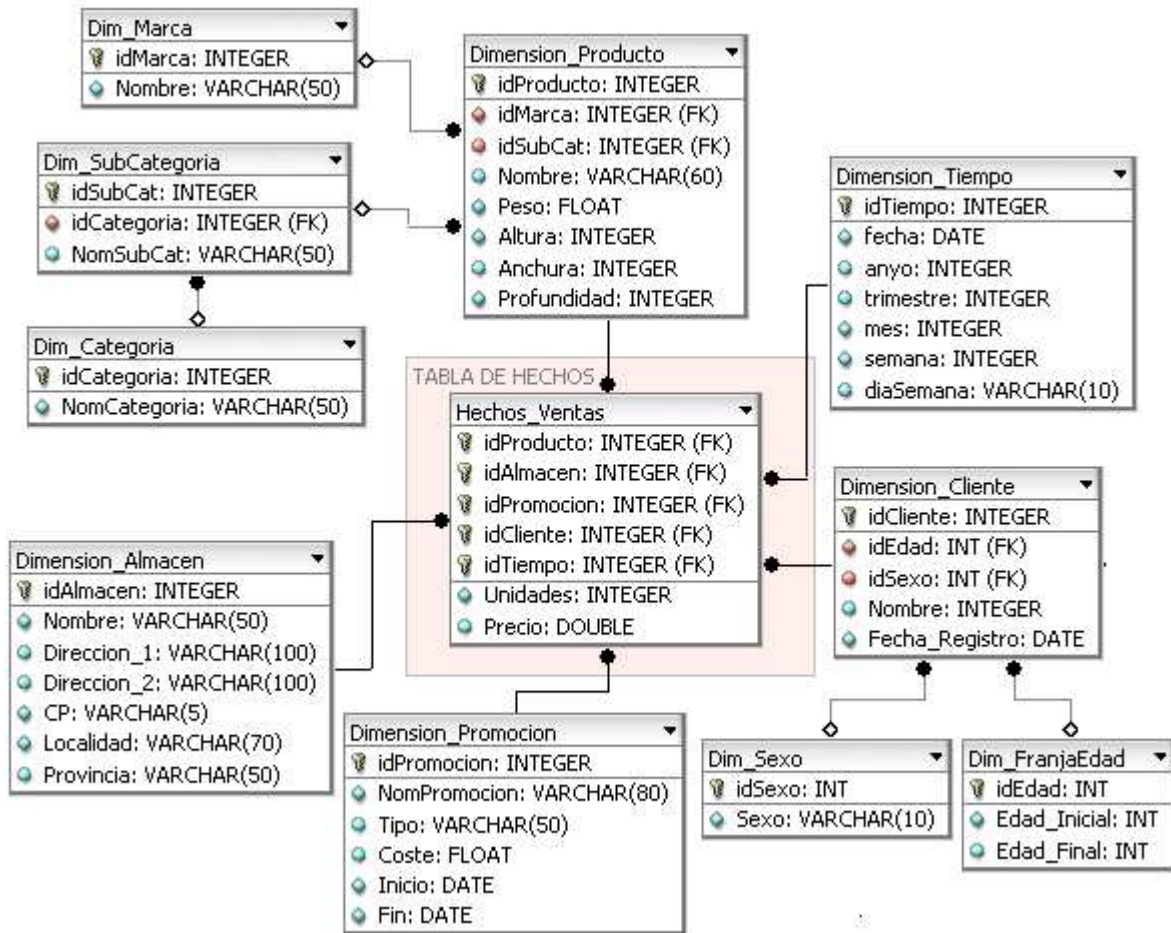


Fig 2.2 Esquema de copo de nieve (Fuente: Wikipedia)

Otro esquema menos popular son **las constelaciones de tablas de hechos**, o tablas de hechos que comparten sus dimensiones. Presenta algunas complicaciones dado que cada tabla de hechos puede presentar su propio grado de agregación, sin embargo el control de la información es más sencillo.

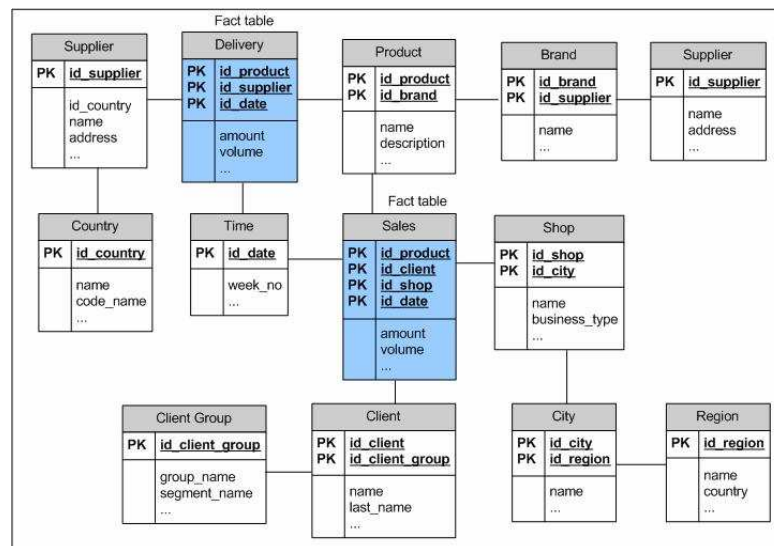


Fig 2.3 Esquema de constelación de tablas de hechos [12.8 Sitios Web 1]

Algunas ventajas de ROLAP son las siguientes:

- Maneja de mejor forma grandes volúmenes de datos y dimensiones de gran cardinalidad.
- Al utilizar una base de datos relacional es posible hacer uso de las herramientas y aplicaciones estándar para ella, como SQL o herramientas de extracción, transformación y carga de datos.
- Permite un mejor control de cambios, puesto que pueden reflejarse cambiando las referencias a las dimensiones o bien modificando un campo en alguna de ellas.
- Los Administradores de bases de datos procesan estos modelos de forma muy rápida, pues pueden evaluar las uniones con las dimensiones realizando una única pasada por la tabla de hechos; primero evaluando las dimensiones, para luego consultar la tabla de hechos con las claves primarias que satisfacen las condiciones.

Algunas de sus desventajas son:

- Las herramientas ROLAP tienen menor rendimiento que las herramientas MOLAP (el cual se definirá más adelante).
- Se requiere de herramientas personalizadas para gestionar la carga de tablas.

2.3.3 Modelo MOLAP

Es el acrónimo de procesamiento analítico en línea multidimensional. Es una alternativa a ROLAP, y a diferencia de él los datos deben ser pre-procesados y almacenados en un cubo. Pese a ocupar menos espacio en disco y a presentar optimizaciones, introduce redundancia de datos y tiene problemas frente a un gran número de dimensiones.

2.3.4 Modelo HOLAP

Acrónimo de procesamiento analítico en línea híbrido, combina ROLAP con MOLAP, almacenando una parte de los datos en cada uno de ellos para obtener las ventajas de cada uno de ellos según corresponda.

2.4 Tecnologías asociadas

Tanto para la implementación del sistema como para la utilización de los distintos módulos del sistema, fue necesario conocer ciertas tecnologías, lenguajes y conceptos de programación. Además fue necesario conocer las costumbres y prácticas del Área de Infotecnologías (ADI), lugar donde se llevó a cabo el trabajo. Estas tecnologías se describen a continuación.

2.4.1 Apache HTTP Server Project

Es un servidor HTTP de código abierto, seguro, extensible y escalable tanto para sistemas UNIX como para Windows. Goza de gran popularidad y es la solución más extendida como

servidor de páginas Web en el mundo. Soporta una gran variedad de lenguajes de programación, como PERL y PHP.

2.4.2 Hypertext Preprocessor (PHP)

Lenguaje de programación de script pensado para el desarrollo de páginas Web. Permite la programación con objetos y clases, además de todas las características propias de un lenguaje imperativo. Cuenta con una gran colección de funciones, clases, documentación y ayuda en línea. Dado lo anterior, goza de gran popularidad y está ampliamente extendido entre los desarrolladores de sitios Web.

2.4.3 Mysql

Es un motor de bases de datos de código abierto. Utiliza la sintaxis SQL para realizar las consultas sobre las bases de datos. Cuenta con una gran variedad de tecnologías de almacenamiento (table engine), como MyISAM o InnoDB. La primera de ellas brinda una mayor velocidad de recuperación de datos, gracias a que no realiza chequeos de integridad. Por otra parte, InnoDB soporta transacciones de tipo ACID (Atomicity, Consistency, Isolation, Durability), además permite el bloqueo de registros y chequeo de integridad de referencias, razones que la hacen más confiable que la primera.

2.4.4 Subversion (SVN)

Programa de código abierto cuya función es administrar las versiones de una aplicación en desarrollo. Está pensado para el trabajo en equipo, permitiendo coordinar las actualizaciones, ver diferencias entre dos archivos y recuperar cualquier versión de un archivo del proyecto. Existen aplicaciones tanto para consolas como del tipo GUI, las cuales facilitan su utilización.

2.4.5 Linux

Es un sistema operativo Unix de código abierto, desarrollado a partir del trabajo de Linux Torvals en 1991. Actualmente se utiliza en servidores, computadores de escritorio y laptops. Existe una gran cantidad de información en línea y está muy extendido a nivel de desarrolladores, contando con un gran número de comunidades que realizan desarrollos.

Existen diferentes versiones o distribuciones, cada una de ellas orientada a diferentes usuarios y necesidades. Todas ellas cuentan con una gran variedad de aplicaciones gráficas, servidores Web, servidores de correo, servidores FTP, servidores SSH, etc.

2.4.6 Entorno de Desarrollo

El ambiente de desarrollo y producción de la aplicación fue provisto por el Área de Infotecnologías (ADI) de la Facultad de Ciencias Físicas y Matemáticas de la Universidad de Chile. En el desarrollo, se cumplieron las normas y estructuras de programación de dicha área.

El sistema funciona sobre un sistema operativo Linux con un servidor Web Apache y programado en el lenguaje PHP. El motor de base de datos utilizado es MySQL. Para el control de versión se utilizó Subversión (SVN) y para su desarrollo la versión de evaluación del editor de PHP Zend Studio.

3 Análisis de la situación actual

En el análisis de la situación actual se debe realizar un catastro de las bases de datos utilizadas por la Escuela para realizar consultas sobre reportes estadísticos e indicadores. En este análisis se incluyen los volúmenes de información, las entidades y Administradores involucrados, el tipo de información almacenada en ellas, entre otros.

Luego de describir las bases de datos incluidas en el catastro se describen las herramientas utilizadas por la Escuela para la generación de indicadores y reportes estadísticos, analizándose de manera más profunda la plataforma SIESTA.

3.1 Catastro de bases de datos

En este punto se describen las bases de datos sobre las cuales se rescata de manera frecuente información para generar reportes estadísticos e indicadores. Estas bases de datos se conocen con los nombres de GUIA, SISPER y SIMBAD.

3.1.1 Base de datos GUIA

Base de datos de la Casa Central de la Universidad de Chile, en la cual se cuenta con la información de matrícula y aranceles de aproximadamente 20.000 alumnos. Las vistas necesarias para extraer información son generadas ad hoc, para lo cual debe especificarse claramente qué datos se necesitan. La STI (Dirección de Servicios de Tecnológica de Información de la Universidad de Chile) es la entidad encargada de generarlas.

3.1.2 Base de datos SISPER

Base de datos del personal de la Universidad de Chile. En esta base de datos se puede encontrar información sobre grados académicos e información de datos laborales y personales.

Algunas de las vistas más utilizadas son las siguientes:

- **NOMBRAMIENTOS:** Esta vista contiene todos los nombramientos vigentes y no vigentes realizados por los departamentos de personal de toda la Universidad, administrados por la Unidad de Remuneraciones.
- **SUELDOS:** Vista que proporciona información referente a procesos de sueldos y el detalle de sus beneficios como columnas asociados a un nombramiento.
- **SITUACIONES:** Vista que proporciona información referente a todas las situaciones de un funcionario en particular, en los cuales su decreto haya sido aprobado por Contraloría Interna.
- **HABERES:** Vista que proporciona información referente a todos los haberes asignados a un funcionario en particular.

- **HABERES_EJECUTADOS:** Vista que proporciona información referente a todos los haberes que se pagaron en un periodo de sueldos

3.1.3 Base de datos SIMBAD

Base de datos relacional que contiene la información sobre alumnos y cursos de pregrado de la Facultad de Ciencias Físicas y Matemáticas de los últimos 40 años. También cuenta con la información de los alumnos titulados desde la década de los treinta del siglo XIX a la fecha.

Desde ella es posible conocer los ramos reprobados, eliminados y aprobados por un alumno. El volumen de información es considerable; maneja aproximadamente 40.000 personas, 1.5 millones de cursos inscritos y 13 carreras, entre otros.

La información contenida en esta base de datos presenta algunos problemas de diseño, los cuales se deben a errores de unificación y conversión, por ejemplo los códigos de colegios no se encuentran unificados y la conversión de los datos desde sus formatos originales (MARKIV) a los actuales (ORACLE) en la década de los ochenta, provocó algunos errores.

Actualmente es administrada por el Jefe del Área de Informática de la Facultad de Ciencias Físicas y Matemáticas. Además, el Área de Info-tecnologías (ADI) cuenta con una copia de ella para realizar pruebas y generar reportes de manera segura e independiente.

El Administrador conoce el modelo de datos y cuenta con la experiencia para estimar si los datos de una consulta son coherentes. Esto resulta de vital importancia si se considera que un reporte se basa en reglas que definen actores o estados, como la condición de mechón o alumno nuevo, que requieren de conocimientos profundo sobre las reglas y mallas académicas, los casos aislados y las condiciones de borde.

3.2 Entidades y usuarios que requieren información

Los actores que generan reportes estadísticos e indicadores son:

- Director Asuntos Académicos Facultad de Ciencias Físicas y Matemáticas de la Universidad de Chile.
- Jefe de Área de Informática de la Facultad de Ciencias Físicas y Matemáticas de la Universidad de Chile.
- Gerente de proyecto del Área de Info-tecnologías (ADI) de la Facultad de Ciencias Físicas y Matemáticas de la Universidad de Chile.

Los actores que solicitan información son:

- Decano Facultad Ciencias Físicas y Matemáticas de la Universidad de Chile.
- Vice-decano Facultad Ciencias Físicas y Matemáticas de la Universidad de Chile.
- Director Escuela Ingeniería y Ciencias de la Universidad de Chile.

- Director Asuntos Estudiantiles Facultad Ciencias Físicas y Matemáticas de la Universidad de Chile.
- Jefa de Bienestar Estudiantil Facultad Ciencias Físicas y Matemáticas de la Universidad de Chile.
- Coordinador Área de Deportes Facultad Ciencias Físicas y Matemáticas de la Universidad de Chile.
- Jefa Área de Inglés Facultad Ciencias Físicas y Matemáticas de la Universidad de Chile.
- Coordinador Área de Humanidades Facultad Ciencias Físicas y Matemáticas de la Universidad de Chile.

Las entidades y actores que solicitan información son:

- Jefes docentes de cada departamento.
- Coordinador de primer año.
- Comisión de evaluación de primer año.
- Directores de cada Departamento.
- Áreas de apoyo estudiantil.
- Centros de alumnos.
- Casa central de la Universidad de Chile.
- Mecesup.

Los actores que manejan información técnica de las bases de datos son:

- Jefe del Área de Info-tecnologías (ADI) Facultad Ciencias Físicas y Matemáticas de la Universidad de Chile.
- Jefe de Área de Informática de la Facultad de Ciencias Físicas y Matemáticas de la Universidad de Chile.
- Jefe Desarrollo del Área de Info-tecnologías (ADI) Facultad Ciencias Físicas y Matemáticas de la Universidad de Chile.

3.3 Herramientas

La Escuela de Ingeniería y Ciencias, para la generación de información estadística e indicadores, recurre al Administrador de sus bases de datos, quien construye consultas en SQL en el administrador ORACLE de la base de datos SIMBAD. Cuando la información contenida en este administrador no es suficiente, se solicita a otras áreas administrativas de la Universidad dicha información, la que es subida a tablas temporales o vistas para poder hacer las consultas con toda la información, esto genera tiempos muertos producto de la burocracia de coordinación. Dicha información normalmente es entregada en un formato Excel o en archivos planos, los cuales muchas veces deben ser tratados por los usuarios con otro tipo de herramientas.

Actualmente la Escuela utiliza dos sistemas separados de estadísticas e indicadores. El primero de ellos corresponde a los reportes y formularios ORACLE que son administrados por el Jefe del Área de Informática de la Facultad de Ciencias Físicas y Matemáticas, el segundo es el sistema conocido como SIESTA.

Al realizar el estudio y análisis de las herramientas actualmente utilizadas por la Escuela, se ha podido comprobar la importancia que representa para el usuario contar con una herramienta que genere reportes estadísticos e indicadores de forma dinámica. Los sistemas dinámicos de estadísticas presentan las siguientes ventajas:

- Mejoran la productividad de los usuarios.
- Mejoran la eficiencia organizacional.
- Rápida generación de reportes.
- Reducción de tiempos, costos y cuellos de botella.

3.3.1 Reportes y formularios ORACLE

Los actuales sistemas de formularios y reportes ORACLE fueron desarrollados por los estudiantes del Departamento de Ciencias de la Computación Rodrigo Trujillo, Cristián Palma y Cristián Acuña en el año 1995, como parte de sus memorias de título.

Los reportes son archivos almacenados en disco que pueden ser editados y que definen el formato y la distribución de la información del reporte en pantalla. Por otra parte, los formularios son interfaces que permiten realizar filtros sobre la información contenida en la base de datos SIMBAD.

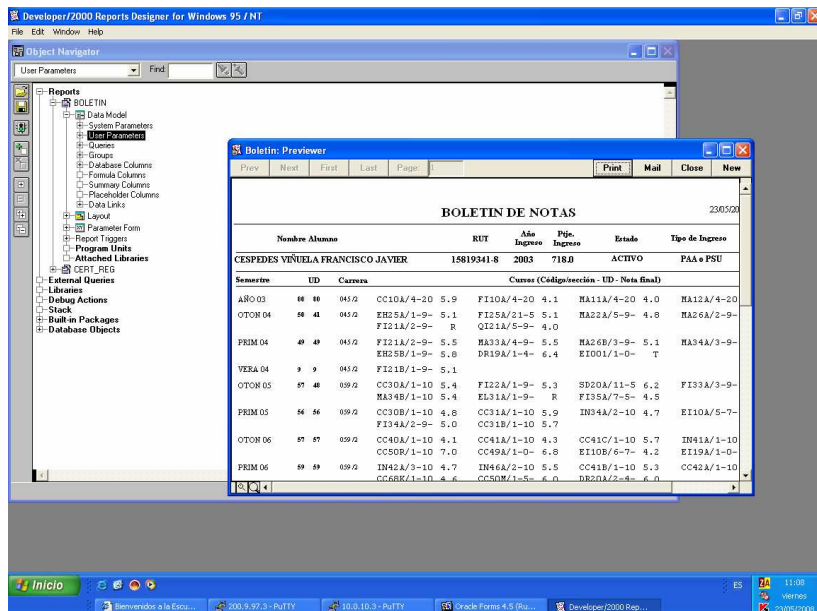


Fig 3.1 Reporte "Boletín de notas" ORACLE utilizado para crear reportes

Estos reportes y formularios incluyen tanto los utilizados por el Administrador de SIMBAD, como los utilizados por funcionarios en el edificio central de la Escuela de Ingeniería y Ciencias de la Universidad de Chile.

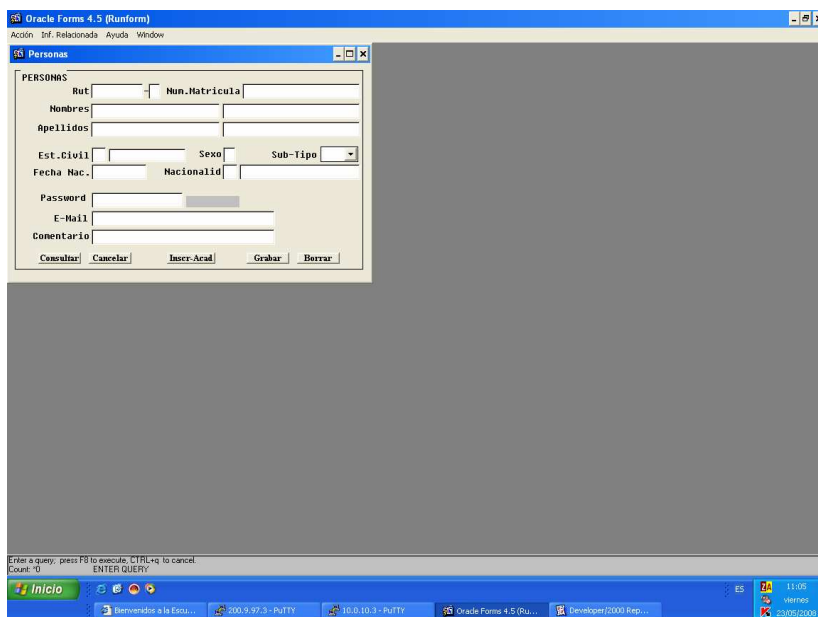


Fig 3.2 Formulario ORACLE utilizado para consultar la base de datos SIMBAD

Las ventajas del sistema de formularios y reportes de ORACLE son las siguientes:

- Requiere instalación en el computador del Administrador, lo cual agrega seguridad al sistema, al restar ubicuidad.
- Cuenta con control de usuario y password para su acceso.
- Usabilidad bien calificada por el Administrador.

Sus desventajas:

- Es un sistema antiguo que puede ser modernizado.
- Provoca una gran dependencia del Administrador, generando cuellos de botella y burocracia al usuario final.

3.3.2 Plataforma U-Campus

Plataforma de administración de servicios, orientada completamente a los usuarios, que surge a finales del año 2006, como una forma de responder a la necesidad de contar con una plataforma que agrupe en forma ordenada y con un perfil de acceso las dispersas aplicaciones con que cuenta la Facultad en diferentes sitios. La aplicación a desarrollar en este trabajo será implantada en esta plataforma, lo cual hace necesario su descripción.

El sistema está dividido en grupos de acuerdo al rol que desempeñe cada uno de los usuarios, lo cual brinda una mayor flexibilidad en el manejo de contenidos. Un alumno podría encontrar los siguientes servicios:

- Acta Consejo
- Boletines Inscripción Académica
- Boletines de Notas
- Workflow

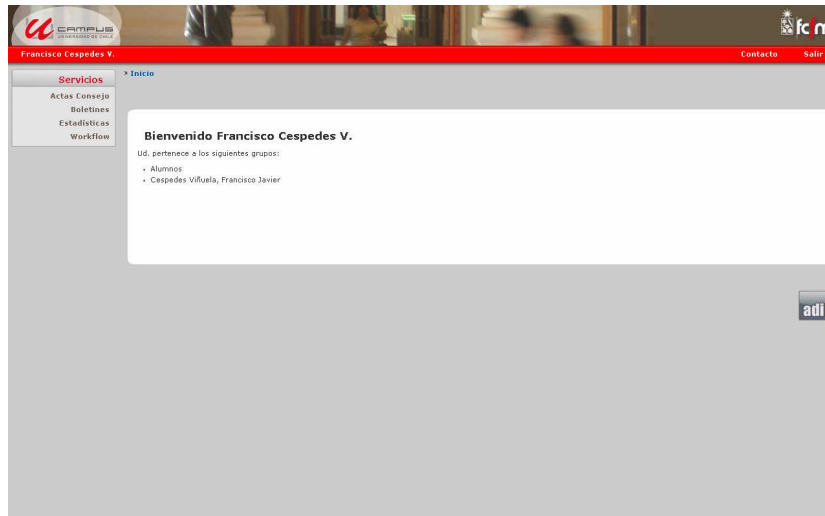


Fig 3.3 Vista para un alumno de la Plataforma U-Campus

Los servicios para un profesor perteneciente al grupo Decanato son:

- Cursos Dictados
- Cursos Históricos
- Estadísticas
- Envío de Correos
- Planilla de Sueldos
- U-Curriculum

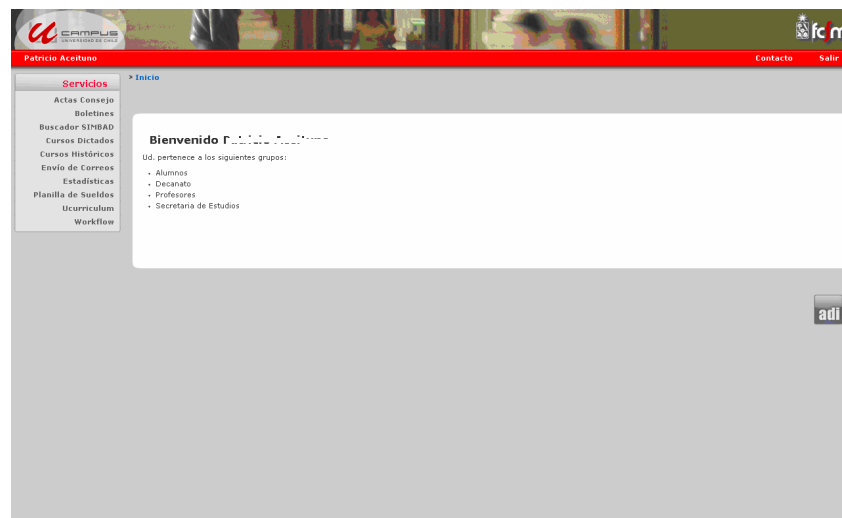


Fig 3.4 Vista para un docente Plataforma U-Campus

La Plataforma U-Campus cuenta con un API que puede ser utilizada por los desarrolladores para implementar y montar aplicaciones en ella. El estándar de este API y

la administración de la información de los usuarios cumplen las normas y reglas de buenas costumbres y de programación del Área de Info-tecnologías (ADI).

3.3.3 Plataforma SIESTA

En este punto se describe la arquitectura física y lógica de la plataforma SIESTA, de manera de contrastar estas arquitecturas con los de la nueva aplicación implementada, denominada SIAGER.

3.3.3.1 Arquitectura física

En forma gráfica la plataforma SIESTA puede ser modelada de la siguiente manera:

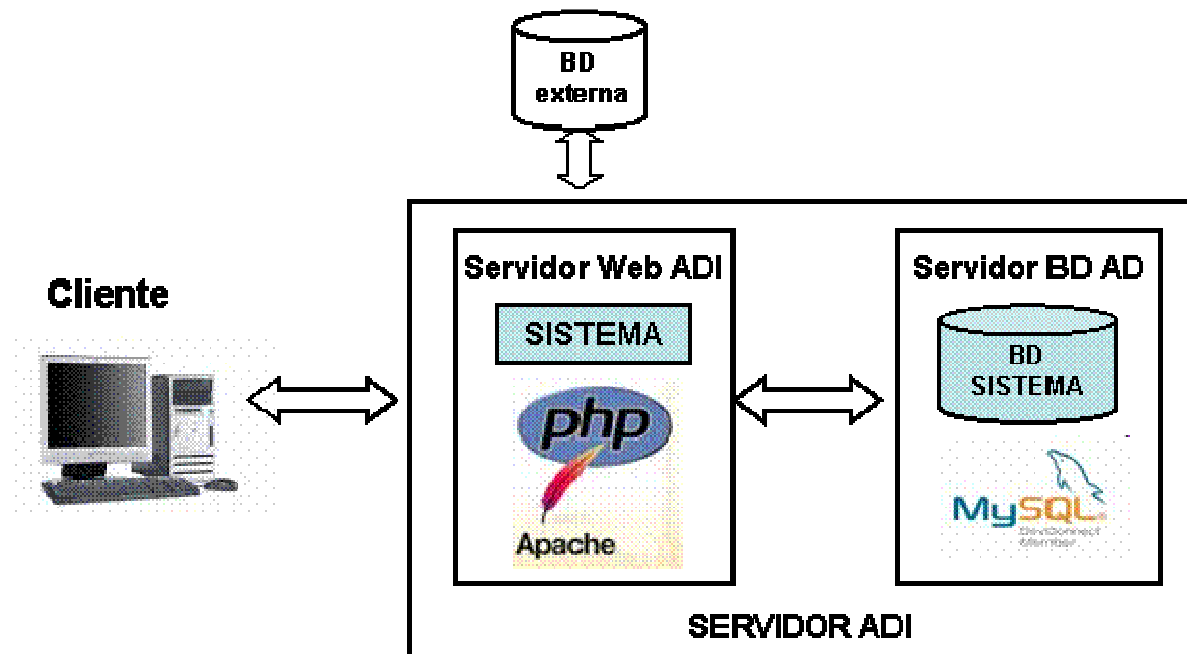


Fig 3.5 Arquitectura física de SIESTA

Como puede verse en la figura 3.5, la implementación de SIESTA cuenta con un motor de base de datos MySQL relacional en el cual se almacena la información recuperada desde fuentes externas, a través de los archivos y funciones con que cuenta el sistema.

Por otra parte, el usuario utilizando un navegador Web accede al sistema instalado en los servidores Web Apache del ADI. El usuario ve un único sistema con toda la información disponible para realizar consultas.

3.3.3.2 Arquitectura lógica

El diseño arquitectónico de la plataforma SIESTA es MVC (Modelo-Vista-Controlador), utilizado en todas las aplicaciones desarrolladas por el Área de Info-tecnologías (ADI). Este modelo consiste en separar la capa de permanencia de datos, los controladores o lógica y la interfaz de usuario. Al separar las capas de esta manera, las modificaciones efectuadas en algunas de ellas no afectarán al resto. Este diseño es ampliamente utilizado en aplicaciones Web.

La capa de presentación interactúa con el usuario y envía a la capa lógica sus acciones. A su vez la capa lógica responde al usuario a través de la capa de presentación. La capa lógica está programada en PHP y también es responsable de controlar la capa de datos. La capa de presentación utiliza plantillas HTML o templates para generar las interfaces de usuario.

La estructura de directorios del sistema cumple con las normas establecidas por el ADI, al separar las capas en directorios distintos. En la siguiente imagen se puede ver la estructura exigida por el ADI para sus proyectos:



Fig 3.6 Estructura de directorio de un proyecto desarrollado por el ADI

En la carpeta *include* se almacenan los archivos de librerías y funciones utilizadas por la capa lógica para controlar la capa de datos o la de presentación. La carpeta *template* contiene los templates o plantillas HTML que constituyen las pantallas presentadas al usuario por el sistema. Finalmente, el directorio *web* contiene los archivos de la capa lógica del sistema. En la imagen anterior se puede ver un archivo presente en todos los sistemas desarrollados en el ADI, el archivo de configuración centralizado *config.php*.

En cuanto a los parámetros de conexión de los sistemas desarrollados en el ADI, cada uno de ellos corresponde a una llave de InfoNúcleo (un arreglo asociativo que define los parámetros de conexión con las bases de datos, los directorios, etc). La clave de InfoNúcleo de los sistemas tiene la siguiente forma:

```
$info['estadisticas'] = array(
    'user' => 'USUARIO',
    'pass' => 'PASSWORD',
    'server' => 'localhost',
    'name' => 'BASE_DE_DATOS',
    'phptype' => 'mysql',
    'ruta_pdf' => '/libPHP/PDFv009/',
    'ruta_temporal' => '/tmp',
    'ruta_base' => '/home/fcespede/public_html/fcfm/estadisticas',
    'ruta_web' => '/~fcespede/fcfm/fcfm/modulos/estadisticas',
    'funciones' => array( 'aad', 'bienestar', 'escpostgrado', 'ucurriculum', 'demre',
);    'simbad' ),
```

Estos parámetros son recuperados en el archivo *config.php* del sistema, para recuperar el valor de la variable *ruta_base* debe realizarse una llamada a la función:

```
$ruta_base=InfoNucleo::getParametro('estadisticas','ruta_base');
```


3.3.3.2.1 Capa de datos

En este punto se describe la capa de permanencia de datos, detallando las bases de datos utilizadas por el sistema, explicando sus campos y analizando la estructura utilizada en cada una de ellas.

3.3.3.2.1.1 Base de datos de los reportes

El modelo de datos de cada reporte en el sistema SIESTA está constituido por una única gran tabla. Considerando el ejemplo de un reporte de tesis y memorias de la Facultad, del año 2007, la tabla que contiene la información utilizada en las consultas estadísticas es la siguiente:










































































	Campo	Tipo	Collation	Atributos	Nulo	Predeterminado	Extra	Acción
<input type="checkbox"/>	id	int(11)			No		auto_increment	     
<input type="checkbox"/>	c_fecha	varchar(255)	latin1_swedish_ci		Sí	NULL		     
<input type="checkbox"/>	c_rut_alumno	varchar(255)	latin1_swedish_ci		Sí	NULL		     
<input type="checkbox"/>	c_alumno	varchar(255)	latin1_swedish_ci		Sí	NULL		     
<input type="checkbox"/>	c_titulo	varchar(255)	latin1_swedish_ci		Sí	NULL		     
<input type="checkbox"/>	c_rut_profesor	varchar(255)	latin1_swedish_ci		Sí	NULL		     
<input type="checkbox"/>	c_nombres_profesor	varchar(255)	latin1_swedish_ci		Sí	NULL		     
<input type="checkbox"/>	c_apellidos_profesor	varchar(255)	latin1_swedish_ci		Sí	NULL		     
<input type="checkbox"/>	c_tipo_profesor	varchar(255)	latin1_swedish_ci		Sí	NULL		     
<input type="checkbox"/>	c_memoria	varchar(255)	latin1_swedish_ci		Sí	NULL		     
<input type="checkbox"/>	c_edad	varchar(255)	latin1_swedish_ci		Sí	NULL		     
<input type="checkbox"/>	c_departamento	varchar(255)	latin1_swedish_ci		Sí	NULL		     
<input type="checkbox"/>	c_categoria	varchar(255)	latin1_swedish_ci		Sí	NULL		     

Fig 3.7 Modelo de datos del reporte tesis y memorias de la Facultad año 2007 en SIESTA

Los problemas asociados a estas tablas son los siguientes:

- Campos declarados como NULL innecesariamente, lo que hace más lento su recorrido y aumenta el espacio utilizado.
- Campos de largo variable, lo que convierte cada tabla a un tipo dinámico. El largo de cada campo solicita más espacio del realmente utilizado, solo algunos de los campos de la tabla llegan a utilizar más de 200 caracteres.
- Algunos campos numéricos no se encuentran separados, por ejemplo el campo c_rut_profesor de la forma XXXXXXX-D podría ser separado en dos campos, uno para el rut y otro para el dígito verificador. Este tipo de campos reduce la velocidad de búsqueda, de agrupación, etc.
- Existencia de tuplas repetidas debido a que la clave primaria definida no asegura la unicidad de la combinación de los campos.
- La utilización de índices en cada campo aumenta significativamente la utilización de espacio en disco y disminuye la velocidad de actualización y carga de información.

- Altos tiempos de respuestas a consultas de agrupación debido a la alta cardinalidad y al largo de algunos de los campos.

3.3.3.2.1.2 Base de datos local (BD Local)

Estas tablas son utilizadas para controlar el proceso de actualización y para almacenar las consultas estadísticas guardadas por el usuario en el sistema. La estructura de estas tablas es la siguiente:

FUNCION	CONSULTAS
FUNCION: VARCHAR(255)	CONSULTAS_ID: INTEGER
FECHA: DATETIME	NOMBRE_DEFINIDO_USUARIO: VARCHAR(255)
COLUMNAS: LONGTEXT	NOMBRE_FUNCION: VARCHAR(255)
TABLA: VARCHAR(255)	PROPIETARIO: INTEGER
TIEMPO: DOUBLE	FECHA: DATETIME
OFFSET: VARCHAR(255)	DATOS: LONGTEXT

Fig 3.8 BD local, cuenta con las tablas FUNCION y CONSULTAS

1. **Tabla FUNCION:** Esta tabla contiene información sobre los atributos de los campos de cada reporte y datos sobre la última actualización realizada a la tabla del reporte. En ella se encuentran los siguientes campos:
 - **FUNCION:** El nombre de la tabla de datos del reporte.
 - **FECHA:** La fecha de actualización del reporte.
 - **COLUMNAS:** Los parámetros para cada columna, por ejemplo: Visibilidad, permisos de agrupación, permisos de selección, tipo, título, descripción, entre
 - **TABLA:** El nombre de la tabla que contiene la información del reporte.
 - **TIEMPO:** El tiempo que tomó la última actualización realizada.
 - **OFFSET:** El OFFSET de la última actualización, utilizado para recuperar incrementalmente información desde fuentes externas.
2. **Tabla CONSULTAS:** Esta tabla contiene información sobre las consultas estadísticas almacenadas en el sistema por los usuarios. Sus campos son los siguientes:
 - **NOMBRE_FUNCION:** Es el nombre de la tabla de datos del reporte.
 - **FECHA:** Es la fecha en la cual fue guardada la consulta.
 - **DATOS:** Corresponde a los parámetros de la consulta, como por ejemplo los filtros, los campos seleccionados, etc.
 - **PROPIETARIO:** Corresponde al rut (sin dígito verificador) del usuario propietario de la consulta.
 - **NOMBRE_DEFINIDO_USUARIO:** Corresponde al nombre asignado por el usuario a la consulta, es el que el usuario recuerda.

3.3.3.2.1.3 Tablas para la actualización de los reportes

Durante la actualización de las tablas de datos de los reportes, se realiza un bloqueo artificial de tablas mediante la creación de una nueva tabla, esta tabla es creada puesto que

la real sufre de ciertos problemas al intentar bloquearla, en algunas ocasiones incluso es eliminada para volver a ser creada. Esta tabla artificial tiene la siguiente estructura:

	Campo	Tipo	Collation	Atributos	Nulo	Predeterminado	Extra	Acción
<input type="checkbox"/>	campo_falso	char(1)	latin1_swedish_ci		Sí	NULL		     

Fig 3.9 Estructura de la tabla artificial creada para el bloqueo de la tabla del reporte

Como se ve en la imagen anterior, el campo campo_falso corresponde a un valor creado únicamente para cumplir con la sintaxis de creación de tablas de SQL, pues no es posible crear tablas sin columnas. En la siguiente imagen se muestran algunas tablas de bloqueo de algunos reportes:

```

cache_estadisticas__alumnos_difusion_2008
cache_estadisticas__alumnos_por_colegio_demre_2007_2008
cache_estadisticas__cruce_psu_2007_2008
cache_lock_estadisticas__alumnos_difusion_2008
cache_lock_estadisticas__alumnos_por_colegio_demre_2007_2008
cache_lock_estadisticas__cruce_psu_2007_2008
cache_lock_estadisticas__cruce_psu_2008_2008
    
```

Fig 3.10 Tablas de bloqueo y de información de los reportes

3.3.3.2.2 Capa de lógica

A continuación se ahondará en la capa de lógica, la cual se encarga de realizar los cálculos, tomar las decisiones del sistema y contener la lógica utilizada para ejecutar las diferentes funcionalidades con que cuenta SIESTA.

3.3.3.2.2.1 Recuperación de información desde fuentes externas

La recuperación de la información desde las fuentes externas se realiza mediante una serie de archivos programados en PHP, los cuales aplican una lógica de rescate incremental almacenando la información en las tablas de datos de los reportes. Estos archivos cuentan con reglas definidas para cada uno de los reportes que la plataforma SIESTA es capaz de entregar. Cada uno de ellos tiene un nombre estandarizado, similar a los de un canal de noticias del protocolo de lectura de noticias (NNTP), es decir, un árbol o estructura de dependencias separadas por puntos. El valor de estos archivos es alto, puesto que definen reglas de negocios y una lógica que ha sido perfeccionada y corregida a lo largo del tiempo.

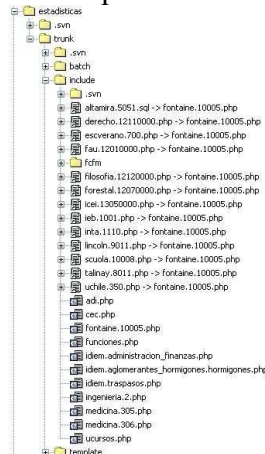


Fig 3.11 Archivos que rescatan información desde las fuentes externas

3.3.3.2.2 Actualización de las tablas de los reportes

La lógica de actualización de las tablas de los reportes del sistema SIESTA puede ser modelada con el siguiente diagrama:

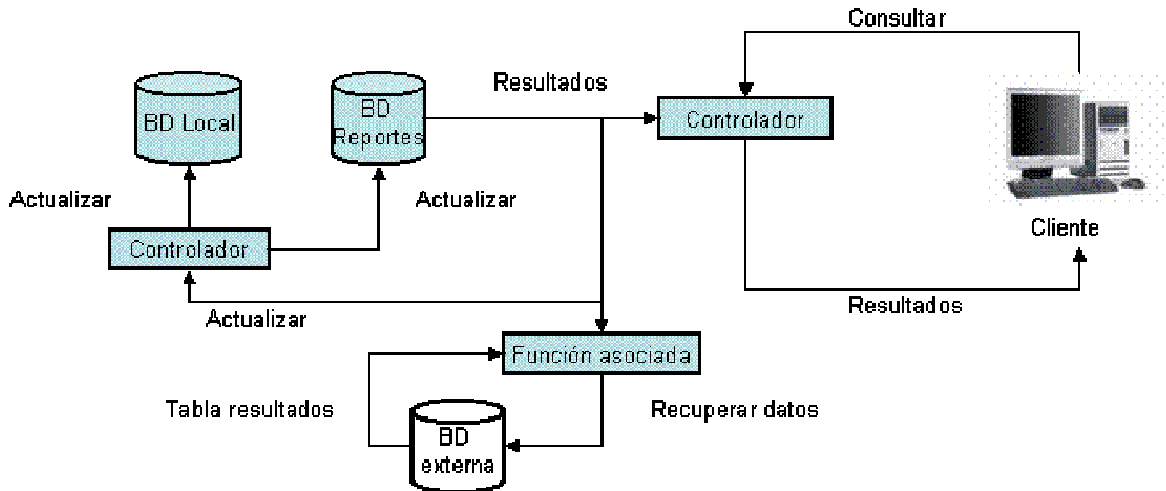


Fig 3.12 Diagrama del proceso de actualización de las tablas de los reportes

Como se puede ver en la figura anterior, cada vez que el usuario ejecuta una consulta estadística un controlador se encarga de rescatar los nuevos datos desde las fuentes externas a través de una función totalmente independiente del sistema. Al aislar estas funciones en archivos especiales es posible que un programador que conoce el modelo de datos de la fuente externa, escriba directamente en ellas una consulta SQL sin necesidad de tener conocimientos del sistema SIESTA. Luego los datos recuperados son almacenados en la tabla local del reporte y los datos de la actualización del reporte son actualizados en la base de datos local, la cual contiene información sobre la actualización de los reportes, consultas estadísticas, etc. Finalmente, se ejecuta la consulta del usuario sobre la tabla del reporte actualizada.

3.3.3.2.3 Capa de presentación

En este punto se describen las diferentes interfaces con que cuenta el sistema SIESTA, de manera de conocer en detalle sus funcionalidades más importantes. Las interfaces están constituidas por una central, donde se realizan las consultas y una de selección de campos visibles.

3.3.3.2.3.1 Interfaz central

En esta interfaz es posible realizar consultas estadísticas que incluyan filtros sobre campos, agrupaciones, seleccionar los campos a ser desplegados y la formas de ordenar los resultados.

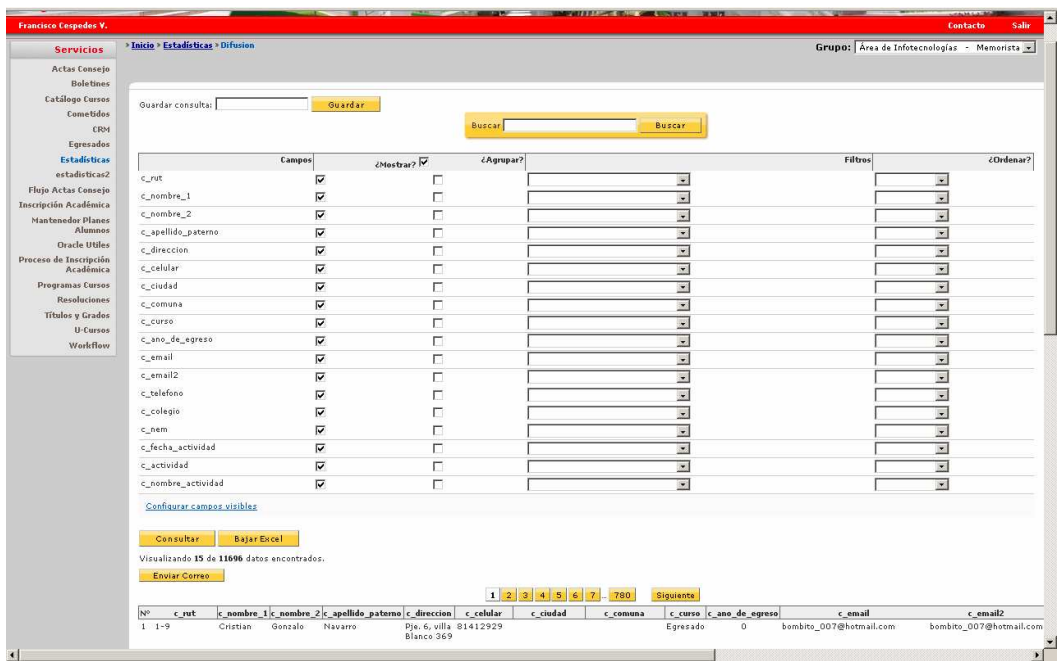


Fig 3.13 Interfaz central

La tabla de resultado de la consulta estadística es dinámica, es decir, varía en su estructura de acuerdo a los resultados. Además, cuenta con la opción de desplegar los resultados en diferentes hojas numeradas, lo cual resulta especialmente útil cuando el número de resultados es considerable.

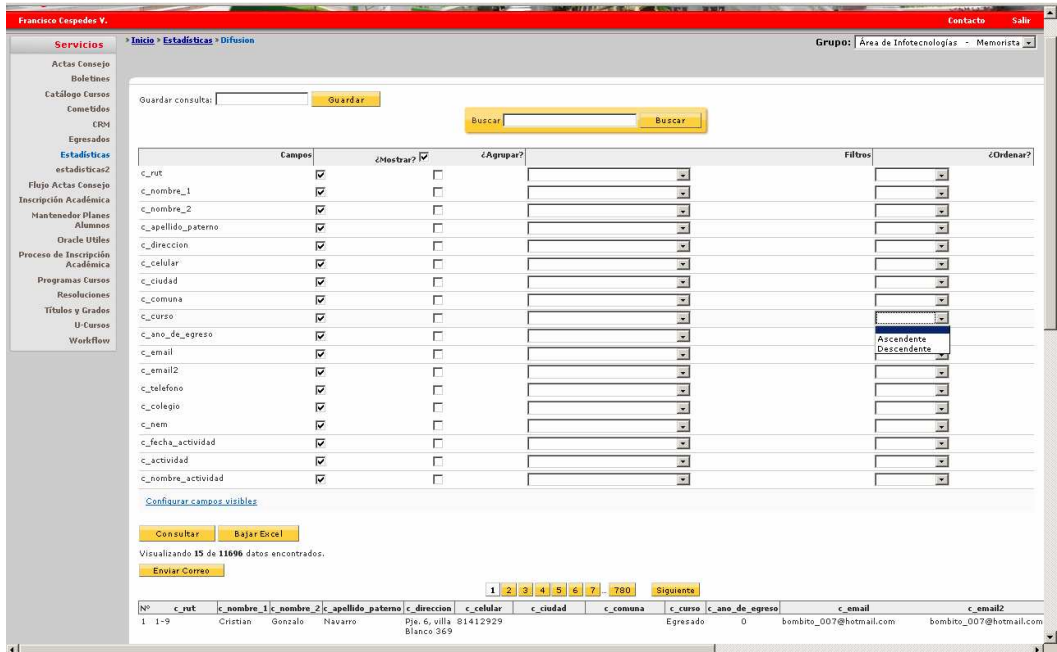


Fig 3.14 Selección de condición de ordenación de los datos a desplegar

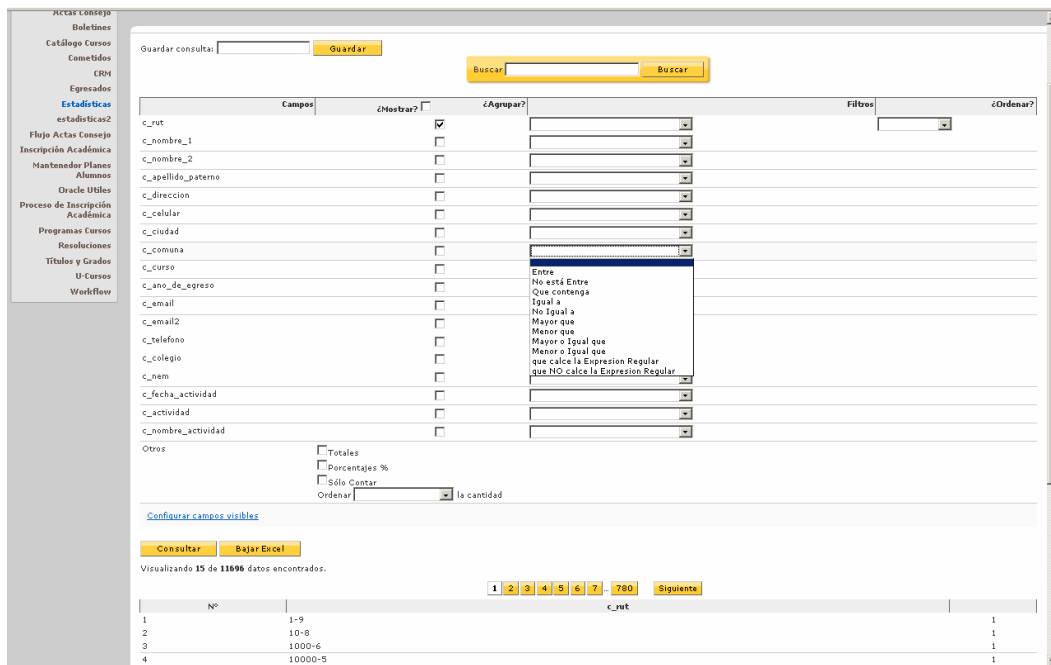


Fig 3.15 Agrupar una consulta con criterios de agrupación como rango de números

La posibilidad de exportar los resultados a un archivo Excel es altamente valorada por el usuario, ya que le permite manejar y analizar los datos en esta aplicación, aprovechando su potencial. Junto con lo anterior, la opción para guardar reportes resulta muy útil para el usuario.

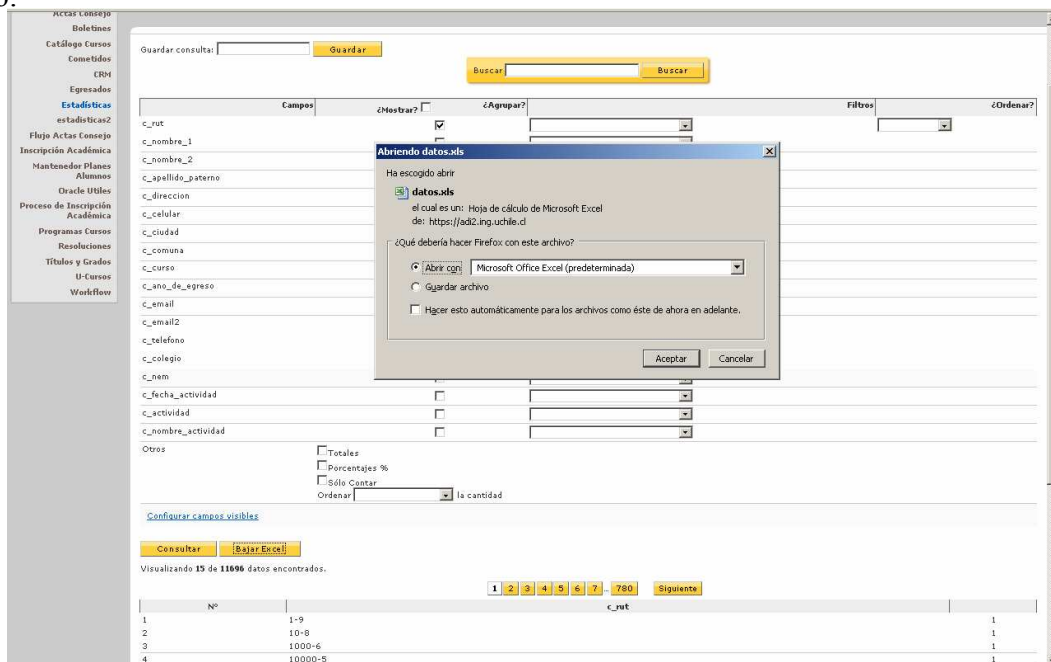


Fig 3.16 Exportar los resultados a un archivo Excel

3.3.3.2.3.2 Interfaz selección campos visibles

En esta interfaz se puede realizar una selección de los campos que quieren hacerse visibles en la interfaz anterior.

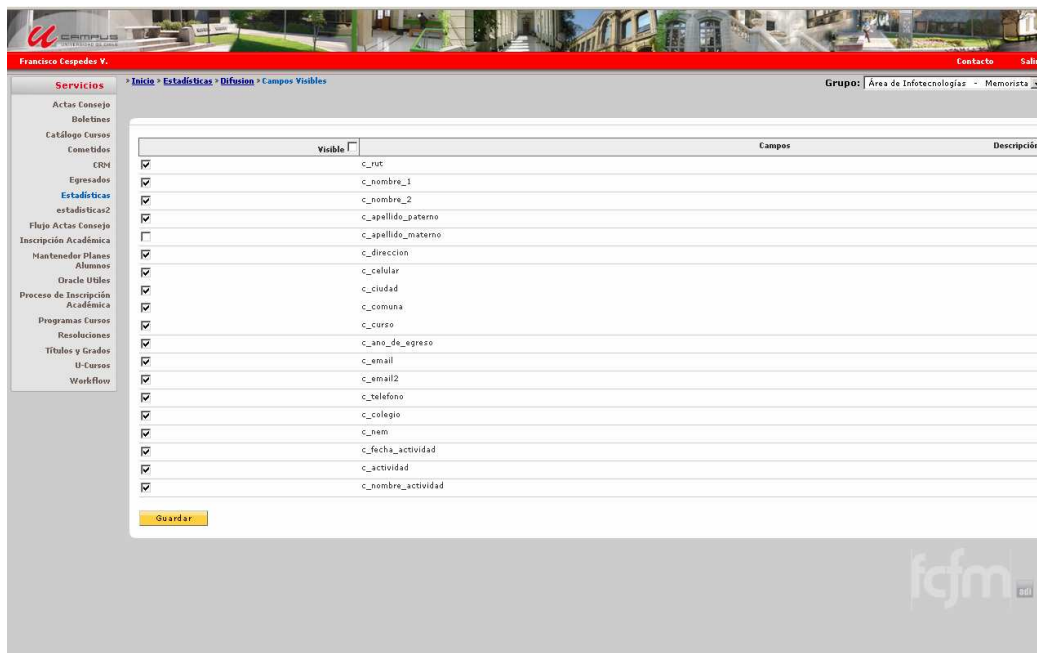


Fig 3.17 Interfaz selección de campos visibles

3.3.3.3 Fortalezas y debilidades del sistema SIESTA

Algunas de sus fortalezas son las siguientes:

- Es un sistema Web que puede ser accedido sin la necesidad de la instalación de archivos de ningún tipo, exceptuando el propio navegador Web.
- Cuenta con control de usuario y password para su acceso.
- Posee una gran versatilidad en cuanto a la cantidad de reportes estadísticos e indicadores que es capaz de generar.

Sus debilidades o puntos mejorables son:

- No posee la capacidad de guardar las consultas estadísticas previamente realizadas en un reporte final.
- No grafica los resultados de las consultas estadísticas.
- No almacena la configuración de un reporte en un archivo en disco. Tampoco carga la configuración de un reporte desde un archivo en disco.
- Debido a que el sistema obtiene la información desde diferentes fuentes, ciertas consultas pueden demorar un tiempo considerable debido a que es necesario

rescatar y cargar la información en el sistema local, produciéndose bloqueos cuando el usuario realiza una consulta.

- El modelo de datos que SIESTA utiliza para almacenar los datos de un reporte consiste en una única gran tabla que contiene los campos necesarios para almacenar la información, y donde cada uno de ellos tiene un índice asociado.
- La extracción de información desde fuentes externas puede ser mejorada cambiando el proceso de actualización en línea por consultas diferidas en el tiempo. Al actualizar la información en línea, algunos usuarios pueden experimentar bloqueos.
- Algunas consultas pueden llegar a ser muy pesadas, provocando problemas de rendimientos en los servidores que las procesan. Los tiempos de respuestas de algunas de estas consultas son demasiado altos.
- Las consultas efectuadas en el sistema por los usuarios rescatan toda la información disponible desde la tabla del reporte, independientemente del tamaño que la tabla pueda tener. Esto aumenta el tiempo de ejecución de la consulta y corre el riesgo de sobrecargar los servidores.
- No cuenta con la posibilidad de realizar cálculos de funciones matemáticas como el promedio, el máximo, mínimo o la desviación estándar. Estas operaciones son de gran utilidad para el usuario que realiza análisis sobre la información disponible.

3.3.4 Percepción de las fortalezas y debilidades de los sistemas actuales

Ninguna de las herramientas utilizadas es capaz de responder a todas las consultas estadísticas formuladas por un usuario. En esos casos son los Administradores de sistemas los encargados de resolverlas, ejecutando consultas SQL directamente en las bases de datos o realizando las peticiones de información que correspondan. En el trabajo a desarrollar se cubrirán los casos más comunes.

Algunos ejemplos de reportes complejos y costosos son los relacionados con las mallas de las carreras, las cuales han cambiado a lo largo del tiempo, lo que obliga a determinar cual es la malla que le corresponde a un alumno.

El medio por el cual las diferentes entidades y personas solicitan estos reportes o consultas son informales y burocráticas, ya que utilizan medios a los cuales resulta difícil realizarles seguimientos. El formato en el cual se entrega la información tampoco está definido y varía en cada solicitud, lo que dificulta las automatizaciones que se pretende lograr en algunos procesos.

3.4 Reportes estadísticos e indicadores frecuentes

A continuación se detallarán los reportes frecuentemente requeridos sobre Alumnos Nuevos, Docentes y estudios de Cohorte solicitados por las autoridades de la Escuela de Ingeniería y Ciencias y de la Facultad de Ciencias Físicas y Matemáticas.

3.4.1 Reportes estadísticos e indicadores frecuentes sobre Alumnos Nuevos

Algunos de los requerimientos frecuentes sobre los Alumnos Nuevos o mechones son:

- Reporte del ingreso de los alumnos según el colegio, región y ciudad de origen.
- Reporte de los alumnos que han aprobado todos los ramos del primer año.
- Reporte de seguimiento de rendimiento o promedio de notas por alumno, curso y sección.
- Reporte de mechones repitentes y eliminados.
- Seguimiento del desempeño de los alumnos en los cursos del primer año.
- Reportes sobre el tipo de ingreso de los alumnos, los que pueden ser por Bachillerato, BEA, Deportistas o por PAA/PSU.
- Reportes de rendimiento de los alumnos según sexo.

Por otra parte, Bienestar estudiantil requiere los siguientes reportes estadísticos e indicadores:

- Entrevista sistemática (Entrevista que se efectúa a todos los nuevos alumnos cuando ingresan).
- Informe social (Utilizada para becas y asuntos judiciales)

3.4.2 Reportes estadísticos e indicadores frecuentes sobre Docentes

Los requerimientos frecuentes en este caso son:

- Reportes sobre el número de docentes por tipo de jornada.
- Reportes sobre los alumnos memoristas de un docente.
- Reportes sobre los cursos dictados por un docente.
- Grados académicos de un docente.

3.4.3 Reportes estadísticos e indicadores frecuentes sobre Cohorte

Los requerimientos frecuentes en este caso son:

- Reporte de seguimiento de los alumnos retirados o re-incorporados.
- Reporte de seguimiento de los alumnos ingresados en un determinado año a lo largo de su carrera.
- Reporte sobre los promedios de un alumno en su carrera antes de la defensa de su título.
- Reporte sobre el promedio, tema, profesor guía y profesores de la comisión de la defensa de su título.

4 Requerimientos

En este capítulo se definen los requerimientos que debe satisfacer el sistema. Estos requerimientos cubren la información requerida frecuentemente, los requerimientos funcionales que definen el comportamiento del software, los requerimientos no-funcionales, los requerimientos que definen las características de la interfaz y por último, los aspectos operacionales y tecnológicos.

4.1 Requerimientos de información

El sistema deberá ser capaz de responder a las consultas estadísticas más frecuentes y a las que se generan con la plataforma SIESTA. Las consultas infrecuentes o complejas serán generadas por el propio Administrador de la base de datos correspondiente.

4.2 Requerimientos funcionales

- El sistema debe contar con un control de acceso y validación de todos los usuarios que acceden a él.
- Debe permitir generar reportes finales, para así evitar realizar consultas estadísticas repetitivamente.
- Cada reporte será paramétrico, esto significa que cada uno de ellos podrá responder a cambios efectuados por el usuario en sus valores de forma dinámica. Esto resultará especialmente útil para reportes que mantienen su estructura a lo largo del tiempo (como reportes sobre mechones), pero que experimentan cambios en la información de acuerdo al año que se considera. Cada parámetro contará con un menú o campo de texto para ingresar su valor, el que podrá modificarse en línea.
- Los gráficos podrán ser de barras, tortas o líneas. Las tablas de un reporte podrán ordenarse ascendentemente o descendentemente según alguna columna.
- El sistema deberá contar con la opción de exportar los reportes en los formatos Excel, PDF y MS Word.

4.3 Requerimientos no-funcionales

- El código, reglas y normas de programación deben seguir lo establecido por el Área de Info-tecnologías (ADI).
- Se utilizará la clase DB provista por la librería PEAR para abstraer las consultas del motor de bases de datos.
- El sistema deberá estar pensado de manera modular y con una alta cohesión, lo que permita realizar modificaciones sencillas, aisladas y rápidas, garantizando su evolución a lo largo del tiempo.

- El sistema debe estar bien diseñado, sin parches y tener una visión integral de los reportes estadísticos e indicadores que podrá generar. En ese sentido, se cuenta con la ventaja de conocer las debilidades y fortalezas de los sistemas actuales.
- El sistema debe estar adecuadamente documentado tanto para el usuario final como para el Administrador, de forma de garantizar su mantenibilidad en el tiempo.

4.4 Requerimientos de interfaces

- Se deberá mantener la interfaz de la Plataforma U-Campus
- El sistema deberá ser compatible con cualquier Browser o navegador Web.

4.5 Requerimientos operacionales

- El sistema debe ser seguro y responsable con el manejo y despliegue de la información, considerando el nivel crítico de la información que maneja.
- Los tiempos de respuestas deben considerar el poco tiempo y paciencia de los usuarios, los cuales esperan tiempos no superiores a diez segundos
- El sistema debe ser capaz de informar a todos los usuarios de las acciones completadas con éxito, interrumpidas y finalizadas con errores.
- El sistema debe garantizar un desempeño adecuado, veloz y confiable, independientemente del volumen de datos que maneje.
- La información utilizada en cada reporte deberá ser pre-calculada, utilizando procesos batch que generen la tabla necesaria para almacenar la información de cada reporte. También podrán realizarse consultas en línea sin utilizar tablas pre-calculadas.

4.6 Requerimientos tecnológicos

La Escuela de Ingeniería y Ciencias y la Facultad de Ciencias Físicas y Matemáticas, en los últimos años han llevado a plataformas Web una gran cantidad de servicios y procesos como una forma de simplificarlos. Siguiendo esa tendencia el sistema a desarrollar en este trabajo deberá ser implementado como una plataforma Web. Entre las razones para seleccionar esta plataforma se encuentra el grado de ubicuidad y acostumbramiento que entrega a los usuarios hacia los cuales va dirigido.

Sus requerimientos tecnológicos son los siguientes:

- El sistema debe estar implementado utilizando el lenguaje de programación PHP.
- El sistema debe abstraerse del motor de bases de datos.
- El sistema debe ejecutarse sobre un sistema operativo Linux.
- El sistema debe utilizar el motor MySQL para su base de datos local.

5 Diseño

El diseño e implementación consistió en generar una nueva versión de la plataforma SIESTA, cuyo nombre es SIAGER (Sistema de apoyo a la generación de reportes). La arquitectura física del sistema SIESTA no fue modificada, a diferencia de la arquitectura lógica, sobre la cual fue necesario rediseñar la capa de datos, desde un modelo de datos en el cual se concebía una única tabla para contener la información de cada reporte a un modelo ROLAP estrella. También fue necesario implementar las funciones necesarias para controlar este nuevo modelo en la capa lógica. Además se rediseñó el proceso de actualización de la información de los reportes, entre otros puntos.

5.1 Capa de datos

Es la capa de permanencia de datos o base de datos, está constituida por un modelo de datos relacional de información local (desde ahora en adelante BD Local) y un modelo de datos ROLAP tipo estrella local (desde ahora en adelante BDEE, acrónimo de Base de datos Estructura Estrella) asociado a cada uno de los reportes, y donde se almacena la información recuperada desde las fuentes externas, por ejemplo SISPER, SIMBAD, UCURRICULUM, entre otras.

El modelo de datos de la BD Local mantiene las relaciones entre las diferentes dimensiones y tablas de hechos del modelo ROLAP estrella de la BDEE, además de definir atributos como los nombres, visibilidades, o descripciones de las diferentes columnas de cada una de las tablas. Este modelo se asemeja al catálogo del sistema de los motores de bases de datos, puesto que a través de él es posible acceder a las relaciones entre las tablas, sus atributos, índices, etc.

Los modelos ROLAP de esquema estrella permiten realizar consultas más rápidas que en un modelo relacional con un esquema diferente, puesto que primero se rescatan desde las dimensiones las claves que satisfacen ciertos filtros, para luego recuperar la información desde la tabla de hechos. Algunos motores de bases de datos pueden realizar esta acción en un solo paso por la tabla de hechos. También permiten un mejor control de cambios al modificar un campo o la referencia a una dimensión.

5.1.1 Tipo de tabla utilizada

El tipo de tabla utilizada en la implementación del sistema fue MyISAM estática. Este tipo de tablas están basadas en el antiguo tipo ISAM que fue utilizada por defecto en las antiguas versiones de MySQL. Al ser estática cada fila tendrá un largo fijo, lo que permitirá recuperarse de fallas fácilmente, pues es sencillo determinar el término de una fila. Otra ventaja es su mayor velocidad y simplicidad para ser almacenadas en cache.

En cuanto a los índices, existen los tipos BTREE y RTREE. El primero de ellos es un índice unidimensional y corresponde a las llaves primarias (Primary key), a los índices únicos (UNIQUE) y a los índices de texto FULLTEXT. El índice RTREE es un índice multidimensional de tipo espacial.

Algunas de las razones que justifican la elección de este tipo de tabla son las siguientes:

- Son las tablas basadas en disco más rápidas. El scan de tablas estáticas de largo fijo es veloz, ya que multiplica el número de la fila por el largo fijo para encontrar el número de índice de la fila (claves subrogadas).
- Pueden ejecutarse en máquinas con relativamente pocos recursos.
- A diferencia de InnoDB es compacta en disco.
- No realiza chequeos de integridad. Si bien esto podría ser un problema para la construcción de un Data Warehouse, esta tarea la asume el programador, que deberá velar por que los datos cargados en estas tablas sean correctos. Este sacrificio se hace con el objetivo de obtener un mejor rendimiento.
- Pueden combinarse varias tablas MyISAM en una única tabla Merge.
- Es recomendada en variadas bibliografías como una buena elección para un tradicional Data Warehouse.
- Permiten utilizar índices del tipo FULLTEXT, útil para realizar búsquedas sobre textos.

Algunos casos en los cuales se justifica su utilización son los siguientes:

- Datos que se regeneran rápidamente
- Tablas que deben contener información 100% correcta y siempre disponible.
- Tablas que pueden formarse a partir de datos exportados desde una base de datos OLTP para mejorar el rendimiento del análisis de los datos.
- Tablas solo para lectura.
- Tablas dispensables.
- Tablas temporales usadas para procesos batch.

5.1.2 Base de datos de información local (BD Local)

El modelo de datos de la BD Local es sencillo y almacena información sobre las tablas de los reportes y sus columnas. También se almacenan los parámetros de los filtros de las consultas estadísticas guardadas en el sistema por el usuario.

A continuación se describen las tablas que constituyen el modelo de datos de información local del sistema:

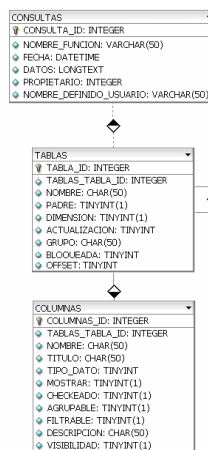


Fig 5.1 Modelo de datos de la BD local

1. **Tabla TABLAS:** Esta tabla contiene las referencias entre las tablas de dimensiones y las tablas de hechos. Sus campos son los siguientes:

- NOMBRE: Define el nombre de la tabla.
- PADRE: Define con un 1 si es una tabla de hechos o con un 0 si no lo es.
- DIMENSION: Define con un 1 si es una tabla dimensión o con un 0 si no lo es.
- ACTUALIZACIÓN: Define con un número el tipo de actualización que se realiza sobre el reporte, el valor 2 define la actualización en línea y el 1 por procesos
- GRUPO: Corresponde al grupo al cual pertenece el reporte.
- BLOQUEADA: Define con un 1 si la tabla está bloqueada por que se está actualizando, con un 0 si está disponible para ser consultada y con un 2 si está en proceso de ser actualizada.
- TABLAS_TABLA_ID: Corresponde a la clave primaria de la tabla de hechos asociado a la dimensión. En el caso de las tablas de hechos este entero toma el
- OFFSET: Contiene el número de la última fila recuperada desde las fuentes externas, útil para realizar una recuperación de información incremental.

Existe una relación 1:N recursiva en ella, debido a que una dimensión está relacionado con una única tabla de hechos, sin embargo una tabla de hechos está relacionada con una o más dimensiones, todas ellas pertenecientes al mismo reporte.

2. **Tabla COLUMNAS:** Esta tabla define los atributos de cada columna de una dimensión o tabla de hechos. Sus campos son los siguientes:

- NOMBRE: Corresponde al nombre de la columnas en la dimensión o tabla de hechos del modelo de datos ROLAP estrella de cada reporte. Estos nombres cumplen con el estándar definido por el ADI.
- DESCRIPION: Corresponde a un texto descriptivo que el usuario puede leer en el módulo de campos visibles.
- TITULO: Define un nombre o título que el usuario puede entender.
- TIPO_DATO: Define el tipo de dato que almacena la columna, el número 1 define a los números y el 2 a las cadenas.
- MOSTRAR: Define con un 1 si el campo se muestra por defecto en el módulo de consultas o 0 si no se muestra.
- CHECKEADO: Define con un 1 si el campo será mostrado por defecto en los resultados de las consultas o 0 si no.
- AGRUPABLE: Define con un 1 si el campo puede ser parte de una consulta de agrupación o 0 si no es agrupable.
- FILTRABLE: Define con un 1 si sobre el campo pueden seleccionarse los diferentes valores que puede tomar. Esta opción permite definir que los campos con alta cardinalidad no sean seleccionables.
- VISIBILIDAD: Define con un 1 si el campo existe para el usuario o con 0 si el campo está oculto.
- TABLAS_TABLAS_ID: Corresponde a la clave primaria de la tabla a la cual pertenece la columna.

Cada columna pertenece a una única tabla. Las tablas tiene una o más columnas, esto explica la relación 1:N entre la tabla TABLAS y COLUMNAS.

3. **Tabla CONSULTAS:** En esta tabla se almacenan las consultas estadísticas guardadas en el sistema por un usuario propietario. Sus campos son los siguientes:

- **NOMBRE_FUNCION:** Es el nombre de la tabla de hechos del reporte.
- **FECHA:** Es la fecha en la cual fue guardada la consulta.
- **DATOS:** Corresponde a los parámetros de la consulta, por ejemplo los filtros, los campos seleccionados, etc.
- **PROPIETARIO:** Corresponde al rut (sin dígito verificador) del usuario propietario de la consulta.
- **NOMBRE_DEFINIDO_USUARIO:** Corresponde al nombre asignado por el usuario a la consulta, es el que el usuario recuerda.

Las consultas se aplican a una única tabla de hechos, cada tabla pueden tener una o más consultas asociadas, esto explica la relación 1:N entre las tablas CONSULTAS y TABLAS.

5.1.3 Base de datos de estructuras estrella (BDEE)

El diseño de estas tablas se basa en el modelo ROLAP estrella. Todos los modelos de datos ROLAP estrella de los reportes cuentan con al menos una dimensión y una tabla de hechos. Cada dimensión forma parte de una única estrella y una tabla de hechos no es compartida por dimensiones de diferentes estrellas.

Para cada reporte existirá una función asociada, la que consiste en una función definida para recuperar la información desde las fuentes externas. Al definir una de estas funciones el sistema SIAGER la reconoce automáticamente y permite que un diseñador cree la estructura de datos ROLAP estrella asociada.

Al utilizar el reporte de ejemplo descrito en la sección 3.3.3.2.1.1 Base de datos de los reportes e implementarlo con un modelo ROLAP estrella las dimensiones y la utilización de espacio en disco de cada una de ellas es la siguiente:

1. Alumno: 952.620 Bytes (Datos + 1 clave primaria simple)
2. Tiempo: 83.176 Bytes (Datos + 1 clave primaria simple)
3. Docente: 186.968 Bytes (Datos+ 1 clave primaria simple)
4. Datos Docente: 4.125 Bytes (Datos+ 1 clave primaria simple)
5. Departamento: 54.465 Bytes (Datos +1 clave primaria simple)
6. Tema (de la memoria): 3.228 KBytes (Datos + 1 clave primaria simple)
7. Tabla de hechos: 2.202 KBytes (Datos+1 clave primaria compuesta)

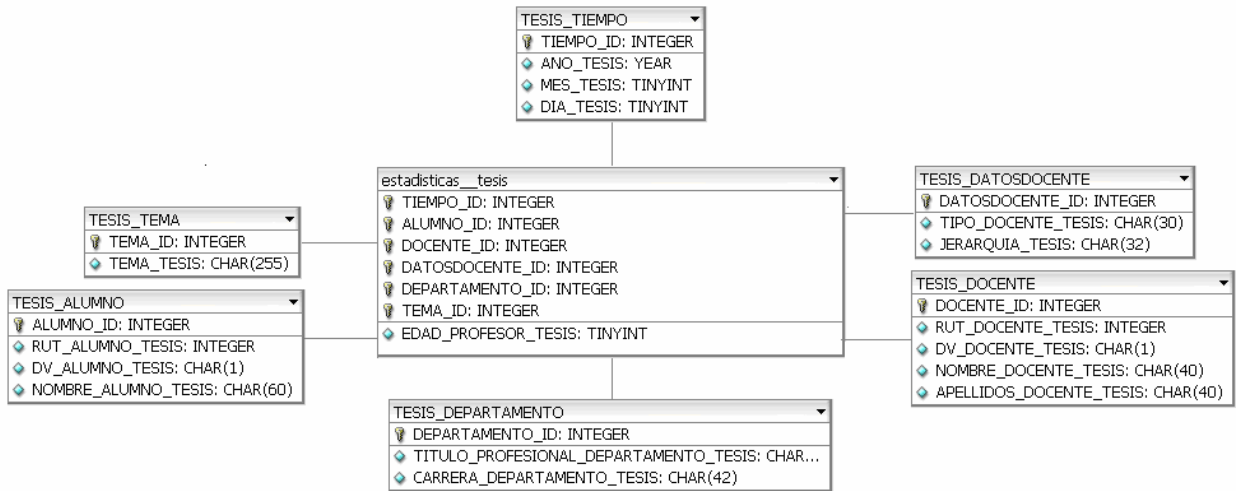


Fig 5.2 Modelo de datos del reporte tesis y memorias de la Facultad año 2007 en SIAGER

La tabla con la información de cada reporte en el sistema SIESTA cuenta con 39.611 filas distintas, las cuales utilizan 15.61 Mb de espacio en disco, incluyendo índices. Su estructura es la siguiente:

Índices: ⓘ				
Nombre de la clave	Tipo	Cardinalidad	Acción	Campo
PRIMARY	PRIMARY	39612		id
c_fecha	INDEX	4951		c_fecha
c_rut_alumno	INDEX	13204		c_rut_alumno
c_alumno	INDEX	13204		c_alumno
c_titulo	INDEX	69		c_titulo
c_rut_profesor	INDEX	1886		c_rut_profesor
c_nombres_profesor	INDEX	1365		c_nombres_profesor
c_apellidos_profesor	INDEX	1800		c_apellidos_profesor
c_tipo_profesor	INDEX	2		c_tipo_profesor
c_memoria	INDEX	13204		c_memoria
c_edad	INDEX	61		c_edad
c_departamento	INDEX	14		c_departamento
c_categoria	INDEX	17		c_categoria

Crear un índice en columnas

Fig 5.3 Modelo de datos del reporte tesis y memorias de la Facultad año 2007 en SIESTA

El espacio total utilizado por el modelo de datos ROLAP estrella de SIAGER es de 6.52Mb, es decir, una reducción del orden del 58%. En este caso también se considera la opción de compresión de claves (PACK_KEYS=1) en la tabla de hechos, lo que reduce el tamaño de esta tabla en 0.36 Mbytes versus la versión sin comprimir, ver anexo 13.3 Análisis de opciones PACK_KEYS. La utilización de memoria también se redujo, como se puede ver en las pruebas realizadas en la sección 5.2.1.7 Comparación de los tiempos de respuesta de SIESTA v/s SIAGER

Cada dimensión tendrá un nombre compuesto por el nombre de la dimensión definida por el diseñador (en el módulo de creación de un reporte) junto con el nombre del reporte como prefijo, ambos en mayúscula y separados por una barra baja (underscore ‘_’). El nombre

de la tabla de hechos estará compuesto por el prefijo ‘estadísticas__’ seguido del nombre del reporte en minúscula. Para evitar colisiones entre nombres de columnas de reportes distintos, cada una de ellas tendrá un sufijo compuesto de un underscore seguido por el nombre del reporte en mayúsculas. Esta estandarización es totalmente transparente para el usuario, el que no necesita conocerla ni entenderla para lograr generar reportes ROLAP estrella.

Con este esquema, cuando un diseñador cree una nueva estructura de datos para un reporte, se garantizará que no habrá colisiones entre los nombres de las tablas o las columnas de otros reportes.

La estructura de las tablas de cada reporte fue declarada siguiendo un conjunto de criterios, con el propósito de resolver efectivamente los problemas que la plataforma SIESTA presentaba. Estos criterios fueron:

1. **Las columnas son declaradas como NOT NULL:** Al declarar una columna como NOT NULL esta ahorra 1 bit, además la lectura de la tabla se ejecuta más rápido.
2. **Las tablas son declaradas como estáticas:** Las tablas estáticas mejoran el scan de tablas, puesto que al multiplicar el largo fijo de las filas por el número de la fila que se requiere es posible obtenerla eficientemente (basada en claves subrogadas). Este tipo de tabla no permite utilizar campos de largo variable como VARCHAR o TEXT, lo cual no constituye un problema, puesto que ninguno de estos tipos es utilizado en ninguno de los modelos de datos ROLAP estrella propuestos ni eventualmente generados por el sistema.
3. **Los identificadores de cada fila son enteros:** Esto facilita la identificación de cada fila, además de hacerlo más eficiente, puesto que los tipos enteros son de largo fijo y pequeños (4 bytes), todo lo cual permite almacenar un mayor número en el cache de índices.

Las claves enteras o subrogadas permiten un control de cambios más sencillo, tablas de hechos y dimensiones de menores tamaño, manejar datos sin códigos operacionales o identificadores naturales que puedan crear conflictos o colisiones, mapear billones (1.000 millones) de filas, recuperar un mayor número de filas de la tabla de hechos por cada operación de Input/Output de bloques, etc.

4. **Utilizar el tipo más pequeño posible:** Utilizar adecuadamente los tipos de datos disponibles, de esta manera se evita ocupar innecesariamente espacio en disco, por ejemplo para valores entre 0 y 65.535 un tipo Smallint debería preferirse sobre el tipo int (entero).
5. **Utilización de la opción PACK_KEYS con valor 1 en la tabla de hechos:** La opción PACK_KEYS tiene tres posibles valores: 1,0 y DEFAULT. El número 1 indica que todas las claves deben comprimirse de manera forzada, el número 0 no compacta ninguna clave y el valor DEFAULT compacta las columnas de caracteres pero no los números, esta es la principal razón por la cual las uniones con claves de cadenas son más lentas comparadas con las de enteros.

la base de datos BD local contiene la información sobre la actualización de los reportes, las relaciones entre las tablas de hechos y las dimensiones, entre otros datos.

Diferentes estrategias se evaluaron para realizar la actualización de los datos contenidos en el modelo de datos ROLAP estrella de cada reporte, estas estrategias son procesos batch y actualización en línea.

El primer método se recomienda para grandes bases de datos o cuando se actualizan un gran número de registros. Intentar realizar un proceso en línea en estos casos resultaría frustrante para el usuario, dado el largo periodo de tiempo que podría tomar la actualización.

El segundo método, está pensado para la actualización de un número reducido de registros, de manera que no pueda ser interrumpido por un time out, y sea tolerable por el usuario que realiza consultas. Para asegurar que el proceso de actualización termine en un tiempo razonable, se ha diseñado un parámetro que limita el número máximo de tuplas que pueden ser actualizadas. Este parámetro se define en el archivo *config.php*.

El usuario cuenta con funciones para manejar el número OFFSET de un reporte y así llevar a cabo una actualización incremental, estas funciones pueden ser utilizadas al interior de las funciones asociadas para recuperar una menor cantidad de tuplas desde las fuentes externas. Estas funciones son las siguientes:

- **updOffset(\$reporte,\$nuevo_offset):** Actualiza el número de OFFSET de un reporte.
- **getOffset(\$reporte):** Obtiene el número de OFFSET de un reporte.

5.2.1.1 Funciones asociadas para actualización en línea

Estas funciones son una parte importante en el proceso de creación de un reporte, pues rescatan la información desde las fuentes externas y también permiten crear la base de datos del reporte.

Para que el sistema reconozca estas funciones es necesario conocer el grupo al cual pertenece un reporte, esto se refiere al grupo de usuarios para los cuales estará disponible un reporte. Cada grupo de usuarios define una estructura de directorios jerárquica en la cual un grupo puede ser parte de otro más general. Por ejemplo, el grupo **dai** es parte del grupo **decanato**, entonces para crear un reporte en el grupo **dai** es necesario manipular el archivo *decanato.dai.php*. Si se quiere definir un nuevo reporte en el grupo **dai** cuyo nombre sea Académicos, entonces debe declararse la siguiente función asociada:

```
1  Function estadisticas__academicos($per_id=",$saccion=REPORTE_CARGAR){
2      If($saccion ==REPORTES_PROPIOS)
3      return validar_creacion_reporte($per_id,RUT_CREADOR,REPORTES_PROPIOS);
4      elseif($saccion ==NOMBRES_CAMPOS_FUNCION)
5      return array('CAMPO1','CAMPO2');
6      elseif($saccion ==REPORTE_CARGAR)
7      validar_creacion_reporte(RUT_CREADOR,RUT_CREADOR,REPORTE_CARGAR);
8      $datos_r=array(0=>array('CAMPO1'=>VALOR,'CAMPO2' =>VALOR),
9          1=>array('CAMPO1'=>VALOR,'CAMPO2'=>VALOR)
10         );
11     return $datos_r;
12 }
```

(En la tabla anterior se ha marcado con negrita las partes que cambian en cada reporte)

Como se puede ver en la tabla anterior, el nombre de todas las funciones asociadas debe comenzar con el prefijo ‘estadisticas__’ seguido del nombre del reporte en minúscula, tal como se dijo anteriormente. Cada función recibe dos argumentos: El identificador de la persona (\$per_id) y la acción (\$accion) que se quiere realizar sobre el reporte. Estos parámetros toman valores por defecto si la función es llamada sin ninguno de ellos (vacío en el primero caso y un valor constante identificando la actualización del reporte en el segundo).

Puesto que cada reporte tiene un administrador o creador, en la línea número 2 y a través de la función **validar_creacion_reporte()** se comprueba que el usuario (\$per_id) que la llama sea el usuario autorizado para crear el modelo de datos del reporte, validando la variable \$per_id contra el valor ingresado por el programador de esta función en el parámetro **RUT_CREADOR**, de esta manera si el creador del reporte tiene como rut 1 (“uno”) la línea número tres se vería de la siguiente forma:

```
return validar_creacion_reporte($per_id, 1 ,REPORTES_PROPIOS);
```

La acción **NOMBRES_CAMPOS_FUNCION** definida en la línea cuatro se refiere a los nombres de los campos del reporte, los cuales serán utilizados en la interfaz de creación de un reporte. En el ejemplo se han definido en la línea cinco los campos **CAMPO1** y **CAMPO2**.

La línea seis se refiere a la acción **REPORTE_CARGAR**, que corresponde a la actualización de los datos en línea, y puede realizarla cualquier usuario conectado al sistema, como se puede ver en la línea siete. Cada vez que un usuario acceda a un reporte e invoca a la función **validar_creacion_reporte()** el reporte se marca como ‘en proceso de actualización’, de esta manera otros usuarios no intentarán actualizarlo y continuarán utilizando el sistema sin ser bloqueados ni sufrir retardos.

Las líneas ocho a la diez definen el arreglo de actualización con los nombres de los campos definidos en la línea cinco. El arreglo debe definir cada fila de datos enumerándolas desde el número cero en adelante.

5.2.1.2 Funciones asociadas para actualización en procesos batch

En este caso el formato que deberá cumplir la función es el siguiente:

```
1 function estadisticas__academicos($per_id=",$accion=REPORTE_CARGAR){
2   if($accion ==REPORTES_PROPIOS)
3     return validar_creacion_reporte($per_id,RUT_CREADOR,REPORTES_PROPIOS);
4     elseif($accion ==NOMBRES_CAMPOS_FUNCION)
5     return array('CAMPO1','CAMPO2');
6     elseif($accion ==REPORTE_CARGAR)
7     validar_creacion_reporte($per_id,RUT_CREADOR,REPORTE_CARGAR);
8 }
```

(En la tabla anterior se ha marcado con negrita las partes que cambian en cada reporte)

Esta función no realiza recuperación de información desde las fuentes externas, sin embargo hasta la línea siete el formato es el mismo al descrito anteriormente. Para recuperar la información desde las fuentes externas el programador deberá crear un archivo separado del sistema, que puede ser llamado de manera automática como un Cron

(programa, comando o script programado para ejecutar una determinada acción a una hora determinada). Un ejemplo de este archivo es el siguiente:

```
1 <?php
2 Include_once('config.php');
3 estadisticas__academicos(RUT, REPORTE_CARGAR);
4 $datos=array(0=>array('CAMPO1'=>VALOR,'CAMPO2' =>VALOR),
5                 1=>array('CAMPO1'=>VALOR,'CAMPO2'=>VALOR)
6                 );
7 guardarDatosReporte('estadisticas__academicos',$datos);
8 ?>
```

(En la tabla anterior se ha marcado con negrita las partes que cambian en cada reporte)

La línea tres realiza el llamado a la función asociada para actualizar la información, esta función validará al usuario y marcará el reporte como 'en proceso de actualización', lo cual evitará que otros usuarios también puedan actualizarlo.

El arreglo \$datos es similar al utilizado en el punto anterior, es un ejemplo de dos tuplas recuperadas desde las fuentes externas y con dos campos insertados en las columnas CAMPO1 y CAMPO2 en la BDEE del reporte.

La línea siete es la función que se encarga de actualizar la base de datos con los datos del arreglo \$datos. Dado que estas funciones pertenecen al sistema, los archivos batch deben ser guardados en alguna carpeta dentro del sistema, de tal manera de tener visibilidad del archivo *config.php*.

5.2.1.3 Actualización en línea

El pseudo-código del algoritmo de actualización en línea es el siguiente:

```
1 Si (estadoLock('estadisticas__academicos') == 'bloqueado') {
2     Error!
3 }
4 Al contrario Si (estadoLock('estadisticas__academicos') == 'no bloqueado'){
5     $datos=estadisticas__academicos()
6     Si (No Vacio $datos){
7         guardarDatosReporte('estadisticas__academicos',$datos)
8     } Al contrario {
9         unlockReporte('estadisticas__academicos')
10    }
11 }
```

Los datos se actualizan solo si el estado de la tabla del reporte es no bloqueado. La línea cinco cambia el estado del bloqueo a 'en proceso de actualización' para luego cargar los datos en la tabla local del reporte llamando a la función **guardarDatosReporte()**, esta función además de cargar los datos desbloqueará la tabla y actualizará el número OFFSET.

Como puede verse en la línea ocho, si los datos que deben ser actualizados en los reportes son vacíos se desbloqueará inmediatamente la tabla del reporte, dado que no se justifica evitar que otros usuarios puedan consultarla.

5.2.1.4 Actualización por procesos batch

No existe un algoritmo de actualización, simplemente debe definirse un archivo batch con un formato similar al ejemplo de la sección 5.2.1.2 Funciones asociadas para actualización en procesos batch.

5.2.1.5 Insertar resultados en las tablas de los reportes en la BDEE

Esta tarea es realizada por la función **guardarDatosReporte()**, utilizada tanto en los procesos batch como en la actualización en línea. Esta función se encarga de consultar la existencia de una tupla en una dimensión o tabla de hechos, para proceder a insertarla si esta no existe.

Algunos casos especiales que considera esta función son los siguientes:

1. **Caracteres especiales:** Algunos campos podrían contener caracteres especiales difíciles de tratar, como el caso de algunos apellidos extranjeros con comillas simple entre sus letras (por ejemplo O'Rayan, O'Connor, etc.), estos casos deben ser considerados para evitar problemas.
2. **Tuplas repetidas:** Los datos rescatados desde el origen podrían tener tuplas repetidas, lo cual provocaría colisión de claves primarias en la tabla de hechos, se hace necesario detectar e ignorar la tupla repetida, insertándolas solo la primera vez.

Otra opción considerada para comprobar la existencia de una tupla fue crear un índice UNIQUE compuesto sobre todos los campos que no cuenten con claves primarias tanto en las dimensiones como en la tabla de hechos, de tal manera de ejecutar una operación INSERT IGNORE y posteriormente un SELECT para obtener la referencia. La sentencia INSERT IGNORE es una extensión de MySQL al SQL estándar que ignora la inserción si existe duplicación. El punto débil de esta opción es la necesidad de crear nuevos índices en todas las tablas, además de no ser una solución estándar.

El pseudo-código de esta función es el siguiente:

```
1  Función guardarDatosReporte($nombre_reporte,$datos){
2  Si(No existe $nombre_reporte en la tabla TABLAS){
3    Error!
4  }
5  Si( estadoLock($nombre_reporte)== 'en proceso de actualización' ){
6    lockReporte ($nombre_reporte)
7    Si (Es Vacío $datos){
8      unlockReporte($nombre_reporte)
9      Salir función
10   }
11   $n_tuplas=min(contar($datos),MAXIMO)
12   $NUEVO_OFFSET=getOffset($nombre_reporte)+$n_tuplas
13   $datos = Limitar el número de tuplas cargadas($datos,$n_tuplas)
14   $mapeo=getMapeoCampo($nombre_reporte ,$datos)
15   Para (cada fila de $datos como $fila){
16     Para (cada campo de $fila como $campo){
17       $dimension_campo=Separar por dimensión($campo, $mapeo)
18     }
19     Para (cada $dimension_campo como $dimension){
20       Insertar $fila en la dimension $dimension sin duplicación
21       $DIMENSION_ID[]=obtener referencia a la dimensión
```

```

22     }
23     Si(No existen referencias $DIMENSION_ID[] en tabla de hechos){
24         Insertar referencias ($DIMENSION_ID[]) y hechos en la tabla de hechos
25     }
26 }
27 unlockReporte($nombre_reporte)
28 updOffset($nombre_reporte, $NUEVO_OFFSET)
29 } Al contrario Si (estadoLock($nombre_reporte)=='bloqueado') {
30     Error!
31 }
32 }

```

Como puede verse en la línea cinco, para poder realizar la carga de los datos en las tablas dimensiones y hechos del reporte, es necesario haber realizado una llamada a la función **estadisticas__academicos()** (considerando el reporte 'Academicos') de tal manera de cambiar el estado del reporte a 'en proceso de actualización'. Una vez que se comprueba el estado del reporte, este se bloquea para evitar que otros usuarios o procesos realicen una nueva consulta a las fuentes externas.

Las líneas siete a la diez se encargan de manejar una condición de borde. Si el usuario intenta cargar un arreglo de datos vacío la tabla se desbloqueará y finalmente se abandonará la función.

Entre las líneas once a la trece se obtiene el número de OFFSET y se limita el número de tuplas máximo a actualizar. Con estos dos valores se determina el nuevo OFFSET que será utilizado en una futura actualización.

La función **getMapeoCampo()**, en la línea catorce, se encarga de mapear o determinar a qué tabla corresponde cada uno de los campos que deben ser actualizados desde la fuente externa. Luego, entre las líneas quince a la dieciocho, y utilizando el resultado de esta función se separará cada uno de los campos de acuerdo a si pertenece a una dimensión o a la tabla de hechos.

Las líneas diecinueve a la veintiuno verifican si la información que debe ser insertada en cada dimensión ya existe, de no existir se procede a la inserción. Luego se recuperan las referencias a las dimensiones que serán utilizadas en la línea veinticuatro para referenciar las dimensiones en la tabla de hechos.

Finalmente, las líneas veintisiete y veintiocho se encargarán de desbloquear el reporte y actualizar el OFFSET en la tabla TABLAS de la BD Local con el valor de la suma entre el antiguo valor y el número de tuplas recuperadas.

5.2.1.6 Ejecución de consultas estadísticas del usuario

A continuación se profundizará en la lógica implementada para la ejecución de las consultas estadísticas ejecutadas por el usuario sobre la información contenida en la base de datos de estructuras estrella.

5.2.1.6.1 Paginamiento de la consulta

La implementación del paginamiento en la plataforma SIESTA se realizaba en memoria, es decir, se ejecutaba la consulta en el servidor MySQL a través de la función **getConsulta()** para luego filtrar los resultados en el script PHP, determinando si el número de la tupla del

resultado se encontraba en la página solicitada por el usuario. Esta implementación presenta una serie de problemas:

1. **Memoria utilizada:** La cantidad de memoria utilizada para cargar los resultados en el arreglo de salida correspondía prácticamente al tamaño de la base de datos del reporte, con el consiguiente riesgo de necesitar más memoria de la disponible.
2. **Velocidad de consulta:** Realizar una consulta que recupere todas las tuplas de una tabla, se vuelve lento si el número de ellas alcanza números grandes.

Para resolver estos y otros problemas se decidió realizar el paginamiento en el motor de bases de datos, recuperando desde este solo los resultados de la página solicitada por el usuario. Para lograr este objetivo, se aprovechó la optimización de la opción LIMIT del SQL en MySQL, las cuales son [12.6 MySQL 33]:

1. Si el número de tuplas son pocas, MySQL puede usar índices cuando normalmente realiza un scan completo de la tabla.
2. Al utilizarse con ORDER BY, la consulta termina cuando se encuentra el largo de resultados solicitados, si se tiene índices, es muy rápido. Si no están ordenados los resultados, MySQL ordena los resultados hasta encontrar el número de tuplas solicitadas, el resto de las filas no son ordenadas.
3. Al agrupar y utilizar LIMIT, MySQL puede leer las claves en orden y detectar los cambios de estas, así no realiza cálculos innecesarios.
4. Cuando una consulta con LIMIT ha sido respondida, el motor aborta todo los procesos relacionados con ella.

Además de las ventajas anteriormente descritas, se puede considerar que el espacio en memoria utilizado por los resultados solo dependerá del tamaño de cada página y no del tamaño de la base de datos, como ocurría en la implementación del sistema SIESTA.

5.2.1.6.2 Determinar el número de filas resultado de una consulta estadística

En relación al punto anterior, para determinar el número total de resultados de una consulta ejecutada por un usuario se definió la función **getCount()**. De esta manera, el número de resultados de una consulta no dependerá del número de resultados de la consulta **getConsulta()**, sino de una consulta SQL que lo calcula.

Visualizando 14 de 14 datos encontrados.

N°	nombi
1	NO DEFINIDO
2	DEPARTAMENTO DE ASTRONOMIA
3	DEPARTAMENTO DE CIENCIAS DE LA COMPUTACION
4	DEPARTAMENTO DE CIENCIAS DE LOS MATERIALES

Fig 5.5 Número de filas por página y de filas totales para una consulta estadística

5.2.1.6.3 Determinar número total de cantidades de la consulta estadística de agrupación

Para determinar el número de cantidades totales de resultados (Totales) para todos los grupos se definió la función **getTotal()**, la cual ejecuta la suma del total de cantidades para todos los grupos en una consulta SQL.

N°	nombre departamento	Cantidad
1	NO DEFINIDO	20512
2	DEPARTAMENTO DE ASTRONOMIA	108
3	DEPARTAMENTO DE CIENCIAS DE LA COMPUTACION	2118
4	DEPARTAMENTO DE CIENCIAS DE LOS MATERIALES	251
5	DEPARTAMENTO DE FISICA	263
6	DEPARTAMENTO DE GEOFISICA	363
7	DEPARTAMENTO DE GEOLOGIA	1252
8	DEPARTAMENTO DE INGENIERIA CIVIL	3145
9	DEPARTAMENTO DE INGENIERIA DE MINAS	715
10	DEPARTAMENTO DE INGENIERIA ELECTRICA	2041
11	DEPARTAMENTO DE INGENIERIA INDUSTRIAL	5815
12	DEPARTAMENTO DE INGENIERIA MATEMATICA	673
13	DEPARTAMENTO DE INGENIERIA MECANICA	1311
14	DEPARTAMENTO DE INGENIERIA QUIMICA	1044
TOTALES		39611

Fig 5.6 Número total de cantidades para todos los grupos (Total: 39611)

5.2.1.6.4 Resultados de una consulta estadística

La función **getConsulta()** se encarga de ejecutar la consulta estadística y al mismo tiempo paginarla utilizando la opción de SQL LIMIT. Esta función también implementa la transposición, opción que fue mejorada pues el número de resultados no correspondía necesariamente al número de filas por página. En otras palabras, si la fila N°1 contara con resultados en 10 columnas transpuestas solo se mostraría esa fila en los resultados.

N°	rut del profesor	día del mes	año	mes del año(1)	mes del año(7)	mes del año(8)	mes del año(10)	mes del año(12)
1	562	1	1972		1			
2	562	2	1972				2	
3	562	3	1973	1				
4	562	5	1973					1
5	562	6	1971	1				
6	562	12	1973					1
7	562	19	1973					1
8	562	28	1974			1		
9	562	29	1975	1				
10	2132	11	1974	1				

Fig 5.7 Transposición del campo mes del año

La imagen muestra 10 resultados o filas, lo que es correcto puesto que el número de resultados debe considerar las filas y no el número de columnas transpuestas con resultados. Para lograrlo, primero se ejecutó la consulta estadística de agrupación por los campos no transpuestos, para posteriormente recuperar todos los campos transpuestos por cada fila agrupada.

5.2.1.6.5 Ejecutar consultas estadísticas agregadas

SIAGER permite ejecutar hasta un máximo de 3 funciones agregadas, las cuales son: Promedio, máximo, mínimo, desviación Estándar y varianza. Esta funcionalidad es importante en cualquier sistema de generación de reportes, puesto que permite realizar estudios sobre la información disponible.

La función que realiza el cálculo de las funciones agregadas es **getConsulta_agregadas()**, la cual realiza la agrupación por los campos definidos por el usuario, de esta manera puede aplicar las funciones agregadas sobre los diferentes campos numéricos seleccionados.

5.2.1.6.6 Automatización de la creación del modelo ROLAP estrella de un reporte

La automatización de la creación de los modelos ROLAP estrellas de los reportes sigue las siguientes etapas:

1. **Declarar las funciones asociadas de cada reporte:** Se debe declarar la función asociada al reporte cumpliendo el formato definido anteriormente, de esta manera podrá ser utilizada para recuperar la información desde fuentes externas, validar al diseñador y auto-completar los nombres de los campos.
2. **Acceder al reporte para crear la estructura ROLAP estrella:** Al acceder a un reporte sin un modelo de datos definido, el usuario es redirigido al módulo de creación de un reporte, donde los campos del reporte son auto-completados utilizando los nombres definidos por el diseñador en la función asociada.

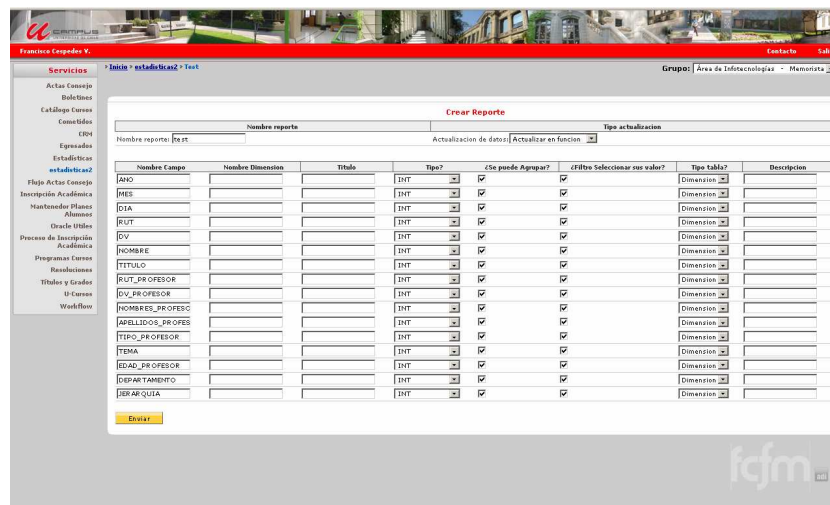


Fig 5.8 Auto-completado de los campos de un reporte utilizando la función asociada

Una vez que el usuario ha completado todos los campos del módulo de creación de un reporte, se realiza una llamada a la función **crearReporte()** con los datos del módulo como parámetros. El pseudo-código de esta función es el siguiente:

```

1  funcion crearReporte($datos){
2  $nombre_reporte=Obtener nombre reporte($datos)
3  $tipo_actualizacion=Obtener tipo actualización($datos)
4  Para (cada $datos como $fila){
5      $tabla=Obtener nombre dimensión($fila)
6      Si ($tabla es Dimensión){
7          $tabla = $nombre_reporte+'_'+$tabla
8          $dimensiones[]=$tabla
9      }
10     $datos_por_tabla[$tabla][]= $fila
11 }
12 Para (cada $datos_por_tabla como $tabla){
13     Si ($tabla es Dimensión){
14         crearDimensionReporte($nombre_reporte,$datos_por_tabla[$tabla])
15     }
16 }
17 crearHechoReporte($nombre_reporte,$tipo_actualizacion, $datos_por_tabla[$nombre_reporte],
    $dimensiones)

```

18 }

En la línea dos se obtiene el nombre del reporte, que será utilizado para definir posteriormente el nombre de las dimensiones utilizando una estandarización, y en la cual el nombre del reporte corresponde a un prefijo.

En la línea tres se obtiene el tipo de proceso de actualización que se realizará sobre el reporte, ya sea a través de un proceso batch o mediante una actualización en línea. Este tipo de actualización será registrada en la BD Local del reporte.

Entre las líneas cinco y nueve se almacenan los nombres de las dimensiones renombradas en un arreglo. En la línea diez se separan los campos por tabla, es decir el arreglo \$datos_por_tabla[\$tabla] contiene todos los campos de la tabla \$tabla.

Finalmente, entre las líneas doce y quince se creará cada una de las dimensiones con todos los campos definidos por el diseñador del modelo. Por otra parte, en la línea diecisiete se creará la tabla de hechos con todas las referencias a las dimensiones.

El pseudo-código de la función **crearDimensionReporte()** utilizada por la función anterior es el siguiente:

```
1 Funcion crearDimensionReporte($nombre_reporte,$campos){
2   $tabla= Obtener nombre de la dimensión ($campos)
3   $nombre_dimension=$nombre_reporte+'_'+$tabla
4   Crear tabla dimensión($nombre_dimension,$campos)
5   $id=Referenciar dimensión en tabla TABLAS($nombre_dimension)
6   Para (Cada $campos como $campo){
7     Referencia columna en tabla COLUMNAS($campo,$id)
8   }}
```

La línea dos obtiene el nombre de la dimensión a partir de los campos ingresados por el usuario, los nombres de cada dimensión son renombrados en la línea tres al igual que en el caso anterior.

Luego de crear físicamente la tabla en la línea cuatro, se inserta su información en la tabla TABLAS en la BD Local. En la línea siete se insertan las referencias para cada una de las columnas de la dimensión en la tabla COLUMNAS, donde la variable \$id es el identificador de la tabla dimensión en la tabla TABLAS.

El pseudo-código de la función **crearHechoReporte()** utilizado en la primera función es el siguiente:

```
1 funcion crearHechoReporte($nombre_reporte,$actualizacion,$campos,$dimensiones){
2   Crear tabla de hechos($nombre_reporte, $dimensiones, $campos)
3   $id=Referenciar tabla de hechos TABLAS($nombre_reporte)
4   Actualizar campo TABLAS_TABLA_ID de dimensiones TABLAS($id)
5   Para (Cada $campos como $campo){
6     Actualizar campo TABLAS_TABLA_ID en tabla COLUMNAS($campo,$id)
7   }
8 }
```

La línea dos crea la tabla de hechos físicamente con las referencias a las dimensiones y los hechos. En la línea tres se insertan las referencias de la tabla de hechos en la tabla TABLAS en la BD Local. En la línea 4 se utiliza TABLAS_TABLAS_ID de cada dimensión para actualizar las referencias de las dimensiones a la tabla de hechos.

Finalmente, en la línea cinco y seis, para cada hecho se registrará la referencia a la tabla de hechos a la cual pertenece mediante la definición del campo TABLAS_TABLAS_ID de la tabla COLUMNAS con el valor de la variable \$id.

3. **Actualización de la información:** Dependiendo de la forma de actualización seleccionada para el reporte, se podrá aplicar la lógica de actualización en línea o con procesos batch.

5.2.1.6.7 Guardar en disco la configuración de la consulta estadística

Esta opción permite almacenar en el disco del usuario, la configuración del módulo de consultas a través de un archivo de texto, el cual brinda al usuario la posibilidad de acceder a un reporte solo contando con el archivo.

En la siguiente imagen se puede ver el resultado de seleccionar la opción de guardar en disco la configuración de un reporte con el nombre nuevo_reporte:

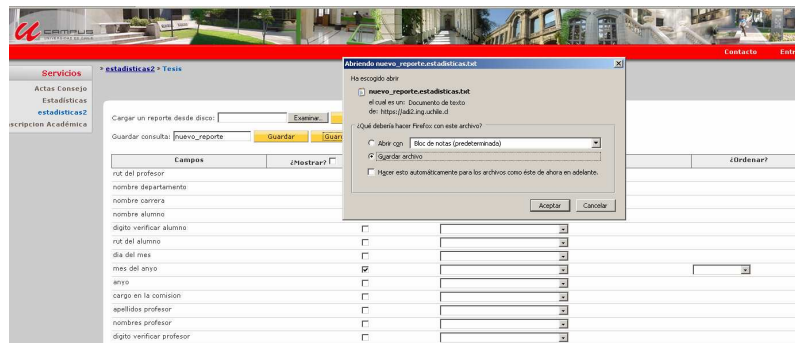


Fig 5.9 Guardar parámetros de una consulta estadística en disco

Este archivo tiene el siguiente formato:

```

nuevo_reporte.estadisticas.bit - WordPad
Archivo Edición Ver Insertar Formato Ayuda

estadisticas_cesis
nuevo_reporte
personas:0
0
57
NEWY {
  '_modulo' => 'estadisticas2',
  '_script' => 'funcion.php',
  'funcion' => 'estadisticas_cesis',
  'bajar' => '1',
  'debug' => 0,
  'browser_global' => '',
  'pagina' => '0',
  'filtro' =>
array (
  'DOC_RUT_PROFESOR_TESIS' => '',
  'DEP_DEPARTAMENTO_TESIS' => '',
  'TDEF_TITULO_TESIS' => '',
  'TALU_NOMBRE_TESIS' => '',
  'TALU_DV_TESIS' => '',
  'TALU_RUT_TESIS' => '',
  'TTIE_DIA_TESIS' => '',
  'TTIE_MES_TESIS' => '',
  'TTIE_AÑO_TESIS' => '',
  'TES_DAT_MEM_TIPO_PROFESOR_TESIS' => '',
  'TDOC_APELLIDOS_PROFESOR_TESIS' => '',
  'TDOC_NOMBRE_PROFESOR_TESIS' => '',
  'TDOC_IV_PROFESOR_TESIS' => '',
  'TES_DAT_MEM_HERARQUIA_TESIS' => '',
  'TES_TEMA_TESIS' => '',
  'TES_EDAD_PROFESOR_TESIS' => '',
)
)
select3' =>
array (
  'DOC_RUT_PROFESOR_TESIS' => '',
  'DEP_DEPARTAMENTO_TESIS' => '',
  'TDEF_TITULO_TESIS' => '',
  'TALU_NOMBRE_TESIS' => '',
  'TALU_DV_TESIS' => '',
  'TALU_RUT_TESIS' => '',
  'TTIE_DIA_TESIS' => '',
  'TTIE_MES_TESIS' => '',
)

```

Fig 5.10 Formato archivo guardado en disco

El nombre del reporte en la primera línea, el nombre con el cual fue guardado en la segunda, el grupo al cual pertenece el reporte en la tercera, el identificador de la persona en la cuarta, el identificador de la consulta estadística en la quinta y los parámetros de configuración de la consulta desde la sexta línea en adelante.

5.2.1.6.8 Cargar la configuración de un reporte desde disco

Esto corresponde a subir el archivo explicado en el punto anterior para cargar la configuración de la consulta estadística que contiene, de manera de ser ejecutada en el módulo de consultas.



Fig 5.11 Opciones para guardar y cargar consultas estadísticas

5.2.1.6.9 Consultas estadísticas almacenadas en el sistema

El sistema cuenta con la posibilidad de cargar una consulta estadística almacenada en el sistema. Dicha consulta es guardada en la tabla CONSULTAS, donde se define un propietario, un nombre y un reporte asociado a ella. Además de guardar las consultas, es posible borrarlas o bien actualizarlas.

5.2.1.6.10 Generación de gráficos automáticamente

Dependiendo de la configuración y las opciones de la consulta estadística, se generarán diferentes gráficos tanto para la interfaz Web como para archivos exportados. Para el primer caso se utilizó la librería Open Flash Chart para generar gráficos animados en Flash y para los archivos exportados la librería Jpgraph para generar imágenes jpeg. Estas librerías fueron seleccionadas debido a que son utilizadas por el ADI. En el caso de las animaciones en Flash se utilizaron las siguientes funciones:

- **barras_agrupadas()**
- **lineas_juntas_agrupadas()**
- **torta_agrupar()**
- **barras_juntas_transpuestas()**
- **barras_transpuestas()**
- **lineas_juntas_transpuestas()**
- **torta_transponer()**

Para generar las imágenes jpeg se realiza un llamado a la función **graficar_export()**, la cual utiliza las siete funciones anteriores para generar las imagen que serán embebidas en cada uno de los reportes finales.

Los gráficos son generados automáticamente de acuerdo a las opciones agrupar o transponer seleccionadas en el módulo de consultas. También dependen del número de

campos agrupados y de las opciones de funciones agregadas, cálculo de porcentajes o totales. Los gráficos serán vistos con más detalle más adelante.

5.2.1.6.11 Exportar reportes finales en archivos Excel

Al igual que en los reportes finales en otros formatos, se distinguirán dos casos: Reportes finales con y sin gráficos. Los reportes finales sin gráficos se generan cuando no existen campos agrupados en la consulta estadística, en estos casos se llama a la función **tabla_simple()**, esta función genera una tabla idéntica a la que ve el usuario en la tabla de resultados del módulo de consulta, para luego ser insertada en el archivo Excel.

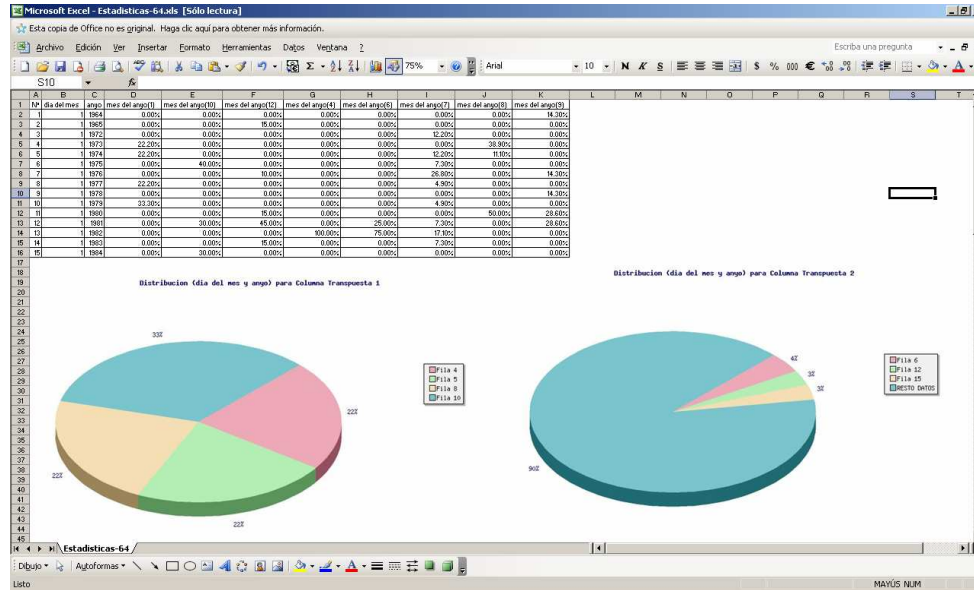


Fig 5.12 Tabla y gráficos de torta de un reporte final como archivo Excel

Los gráficos de los reportes se generan cuando se selecciona uno o más campos de agrupación, en estos casos es la función **graficar_export()** la encargada de generar las imágenes, ya sean gráficos de barras, tortas o líneas. Cuando existe al menos un campo agrupado, es la función **tabla_graficos()** la encargada de generar la tabla de resultado de la consulta estadística que será insertada en el archivo Excel.

Un objetivo al generar un reporte final en Excel, es contar con imágenes embebidas en el archivo. Las opciones consideradas para lograr este objetivo fueron las siguientes:

1. **Creación de un archivo en formato Open XML:** SpreadsheetML es un formato XML utilizado por Microsoft que abarca la gran mayoría de las funcionalidades, como formulas, datos, formato, etc. Es parte de la especificación Open XML y ha sido ratificado por ECMA como un estándar.

Algunas de sus ventajas son las siguientes:

- Puede ser abierto, modificado y exportado libremente en Excel.
- Portable, puede ser generado y procesado en cualquier computador con las facilidades para XML.
- Es un formato no binario que permite procesar los documentos.

Sin embargo, solo funciona en computadores con Windows XP o Windows 2000 y si bien permite incluso la creación de gráficos directamente, no permite embeber imágenes.

2. **Exportar un documento Excel con formato HTML:** Si bien esta solución permite generar documentos Excel reconocidos por la gran mayoría de los computadores, las imágenes solo pueden ser incluidas como links utilizando la etiqueta HTML img.

Existe un protocolo para embeber imágenes como link dinámicamente, el que se encuentra definido en el RFC 2397, sin embargo las herramientas de Microsoft, como Excel e IExplorer, no reconocen este tipo de imágenes.

Finalmente, la solución seleccionada fue la creación de un archivo MHTML y la implementación de la librería excelHTML. Este formato, es reconocido e interpretado adecuadamente por Excel y permite embeber imágenes en él. Estos archivos MHTML no son reconocidos por Excel en computadores Mac, razón por la cual se podrá exportar informes finales en Excel sin gráficos utilizando una librería disponible en el ADI.

5.2.1.6.12 Exportar reportes finales en archivos PDF

Para generar reportes en este formato se utilizó la librería disponible en el ADI llamada **Cezpdf**, la cual puede genera reportes con tablas y gráficos. El formato PDF puede ser visualizado en cualquier máquina sin problemas debido a que se encuentra estandarizado.

Para generar el reporte final en este formato, se realizan llamadas a las mismas funciones descritas en el caso de los archivos Excel, solo cambia la clase que genera el reporte y sus métodos.

The screenshot shows the Adobe Acrobat Professional interface with a PDF document titled 'Estadísticas(2).pdf'. The main content is a table with the following data:

Estadísticas										
#*	dia	del	mes	anho	mes	del	anho	mes	del	anho
1	1		1964	0.0 %	10.0 %	0.0 %	0.0 %	0.0 %	0.0 %	14.3 %
2	1		1965	0.0 %	0.0 %	15.0 %	0.0 %	0.0 %	0.0 %	0.0 %
3	1		1972	0.0 %	0.0 %	0.0 %	0.0 %	0.0 %	12.2 %	0.0 %
4	1		1973	22.2 %	0.0 %	0.0 %	0.0 %	0.0 %	0.0 %	0.0 %
5	1		1974	22.2 %	0.0 %	0.0 %	0.0 %	0.0 %	12.2 %	11.1 %
6	1		1975	0.0 %	10.0 %	0.0 %	0.0 %	0.0 %	7.3 %	0.0 %
7	1		1976	0.0 %	0.0 %	10.0 %	0.0 %	0.0 %	26.8 %	14.3 %
8	1		1977	22.2 %	0.0 %	0.0 %	0.0 %	0.0 %	4.9 %	0.0 %
9	1		1978	0.0 %	10.0 %	0.0 %	0.0 %	0.0 %	0.0 %	14.3 %
10	1		1979	33.3 %	0.0 %	0.0 %	0.0 %	0.0 %	4.9 %	0.0 %
11	1		1980	0.0 %	0.0 %	15.0 %	0.0 %	0.0 %	0.0 %	24.6 %
12	1		1981	0.0 %	20.0 %	45.0 %	0.0 %	25.0 %	7.3 %	0.0 %
13	1		1982	0.0 %	10.0 %	0.0 %	100.0 %	75.0 %	17.1 %	0.0 %
14	1		1983	0.0 %	0.0 %	15.0 %	0.0 %	0.0 %	7.3 %	0.0 %
15	1		1984	0.0 %	30.0 %	0.0 %	0.0 %	0.0 %	0.0 %	0.0 %

Fig 5.13 Tabla de resultados de un reporte final como archivo PDF

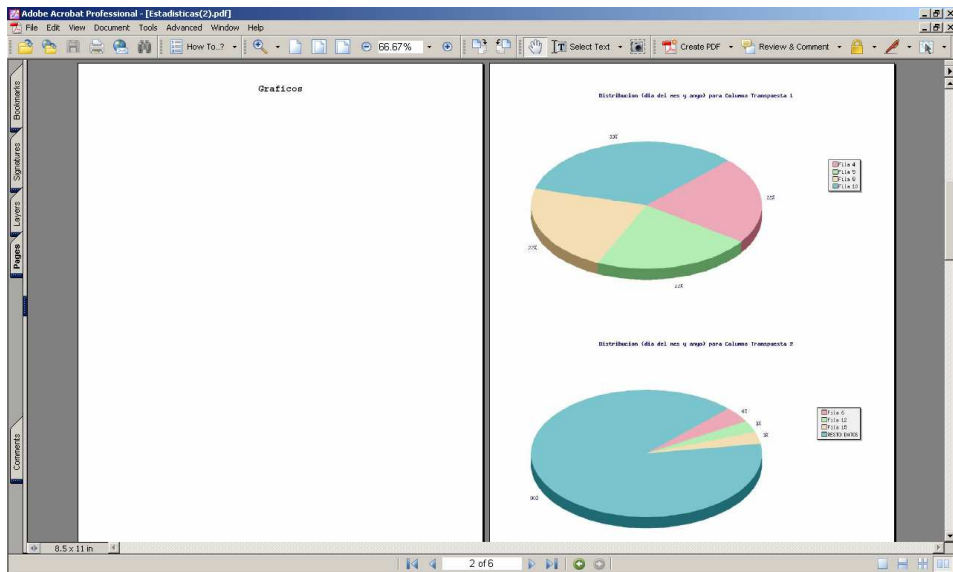


Fig 5.14 Gráficos de torta de un reporte final como archivo PDF

5.2.1.6.13 Exportar reportes finales en archivos WORD y RTF

Esta opción se implementó utilizando el estándar Open XML y WordprocessingML, este tipo de archivos tiene las mismas ventajas de las señaladas anteriormente para el caso de SpreadsheetML, además de permitir embeber imágenes directamente en el archivo. Para generar este tipo de archivo se implementó la librería wordML, sin embargo este formato solo es reconocido por computadores Windows XP o Windows 2000, por lo tanto, también se cuenta con la posibilidad de exportar los reportes finales en formato RTF utilizando una librería disponible en el ADI.

Para generar el reporte final en este formato, se realizan llamadas a las mismas funciones descritas para el caso de los archivos Excel, solo cambia la clase que genera el reporte y sus métodos.

N	día del mes	año	mes del año(1)	mes del año(2)	mes del año(3)	mes del año(4)	mes del año(5)	mes del año(6)	mes del año(7)	mes del año(8)	mes del año(9)
1	1	1964	0.0%	0.0%	0.0%	0.0%	0.0%	0.0%	0.0%	0.0%	14.3%
2	1	1965	0.0%	0.0%	12.8%	0.0%	0.0%	0.0%	0.0%	0.0%	0.0%
3	1	1972	0.0%	0.0%	0.0%	0.0%	0.0%	0.0%	12.2%	0.0%	0.0%
4	1	1973	22.2%	0.0%	0.0%	0.0%	0.0%	0.0%	0.0%	38.9%	0.0%
5	1	1974	22.2%	0.0%	0.0%	0.0%	0.0%	0.0%	12.2%	11.1%	0.0%
6	1	1975	0.0%	40.0%	0.0%	0.0%	0.0%	0.0%	7.3%	0.0%	0.0%
7	1	1976	0.0%	0.0%	10.0%	0.0%	0.0%	0.0%	26.8%	0.0%	14.3%
8	1	1977	22.2%	0.0%	0.0%	0.0%	0.0%	0.0%	4.9%	0.0%	0.0%
9	1	1978	0.0%	0.0%	0.0%	0.0%	0.0%	0.0%	0.0%	0.0%	14.3%
10	1	1979	33.3%	0.0%	0.0%	0.0%	0.0%	0.0%	4.9%	0.0%	0.0%
11	1	1980	0.0%	0.0%	15.0%	0.0%	0.0%	0.0%	0.0%	30.0%	28.6%
12	1	1981	0.0%	30.0%	45.0%	0.0%	0.0%	25.0%	7.3%	0.0%	28.6%
13	1	1982	0.0%	0.0%	0.0%	100.0%	25.0%	12.1%	0.0%	0.0%	0.0%
14	1	1983	0.0%	0.0%	15.0%	0.0%	0.0%	0.0%	7.3%	0.0%	0.0%
15	1	1984	0.0%	30.0%	0.0%	0.0%	0.0%	0.0%	0.0%	0.0%	0.0%

Fig 5.15 Tabla de resultados para un reporte final como archivo WORD

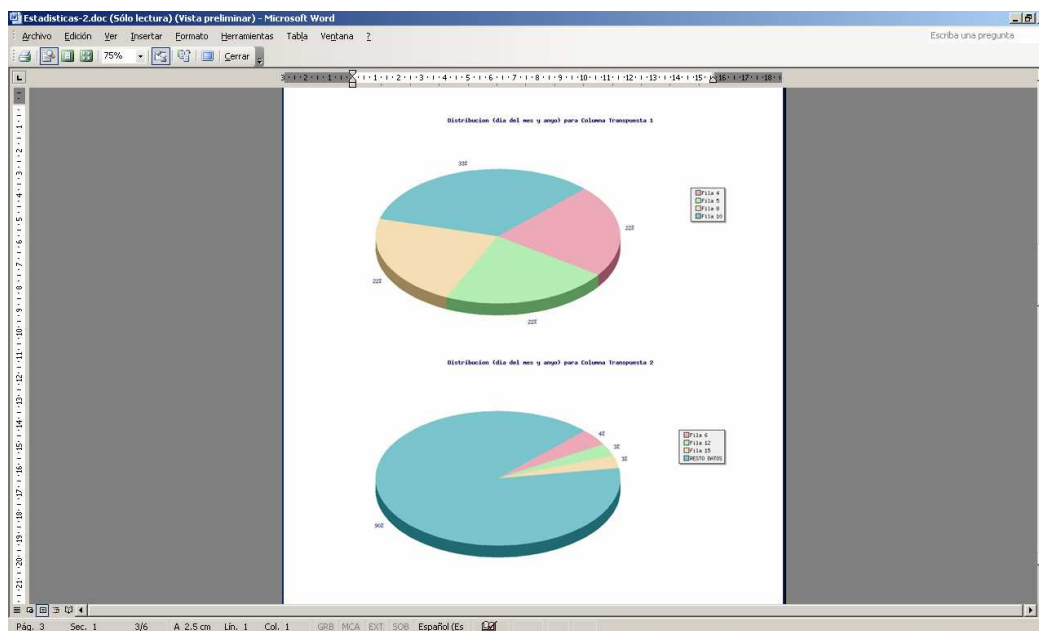


Fig 5.16 Gráficos de torta para un reporte final como archivo WORD

5.2.1.7 Comparación de los tiempos de respuesta de SIESTA v/s SIAGER

Además de lograrse un ahorro del espacio utilizado en disco por cada reporte, se mejoraron los tiempos de respuesta a las consultas estadísticas de los usuarios en el módulo de consultas, como se puede ver en la siguiente tabla (las pruebas fueron hechas en los servidores del ADI sobre el reporte tesis y memorias de la facultad del año 2007 descrito anteriormente). Los tiempos de respuesta a algunas consultas son los siguientes:

Consulta	SIESTA (segundos)	SIAGER (segundos)
(1) Mostrar todos los campos	5	1
(2) Agrupar por fecha	2	1
(3) Agrupar por rut del alumno	2	0
(4) Agrupar por rut del profesor	1	1
(5) Agrupar por todos los campos de texto	17	6
(6) Agrupar por todos los campos	55	7

También se navegó a través de las diferentes páginas de resultados para cada una de estas consultas, con los siguientes resultados por consulta y hoja:

SIESTA:

Hoja	Consulta (1)	Consulta (2)	Consulta (3)	Consulta (4)	Consulta (5)	Consulta (6)
1	5	2	3	1	82	54
7	3	2	2	1	57	59

10	3	2	3	1	27	57
13	3	2	2	1	18	65
16	3	1	2	1	17	72
19	3	2	2	2	19	61
22	3	1	2	1	21	58
25	6	2	2	1	20	52
28	3	1	2	2	19	52
31	3	2	2	2	18	60
última	3	2	3	2	15	75

SIAGER:

Hoja	Consulta (1)	Consulta (2)	Consulta (3)	Consulta (4)	Consulta (5)	Consulta (6)
1	2	0	1	1	6	7
7	1	1	1	1	6	6
10	1	0	0	1	6	7
13	0	1	1	1	8	7
16	1	0	0	1	7	6
19	1	1	1	1	7	7
22	1	1	0	1	7	7
25	1	1	1	1	7	7
28	1	0	1	1	7	7
31	1	1	1	1	7	6
última	1	1	0	1	7	7

Con respecto a la utilización de la memoria total disponible (aproximadamente 514.172 Kilobytes al momento de realizar la prueba), al seleccionar todos los campos y rescatar los datos desde la base de datos los porcentajes de utilización son los siguientes:

SIESTA: Utiliza alrededor del 35% de la memoria disponible.

SIAGER: Utiliza alrededor del 3% de la memoria disponible.

En conclusión, basado en las pruebas y en los hechos antes mencionados, SIAGER se presenta como una solución eficiente en cuanto a la utilización de memoria, disco y a los tiempos de respuestas cuando se compara con SIESTA, puesto que:

- Para un reporte de ejemplo **permitió reducir en un 58%** la utilización del espacio en disco utilizado por SIESTA.
- Para un reporte de ejemplo **permitió reducir en más de 11 veces** la utilización de memoria de SIESTA al consultar los reportes.
- Para un reporte de ejemplo permitió reducir los tiempos de respuestas y ejecutar consultas que implican una carga importante en el servidor de base de datos en **SIAGER** de una manera muy buena en cuanto a la utilización de disco, memoria y tiempos de respuesta para bases de datos de gran tamaño y consultas pesadas.

5.3 Capa de presentación

El sistema SIAGER utiliza el diseño de templates para implementar la capa de presentación. Estos templates son manejados por un controlador de la capa lógica del mismo nombre, por ejemplo el template *funcion.html* es controlado por el script PHP *funcion.php*. Una norma de diseño del ADI establece que las variables y las llamadas a funciones deben ser realizadas en el controlador, cada template solo debe imprimir los valores de las variables.

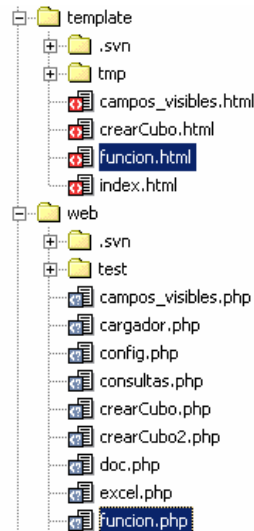


Fig 5.17 Controlador y template del módulo de consulta (funcion.php y funcion.html)

Para manejar la información que se muestra en la capa de presentación, cada template puede ejecutar código PHP utilizando las marcas “{}”, es decir, si se quiere imprimir el valor de una variable basta con realizar un “{\$variable}” en el template. Estas marcas son remplazadas por el código `<?php print $variable; ?>` al cargar los templates, utilizando la función `template()` disponible en las librerías del ADI.

Si es necesario utilizar estructuras de control de flujo o iterar utilizando PHP, deben utilizarse los comentarios HTML para cada línea de código PHP que se quiera insertar en el template, por ejemplo, el siguiente código imprimirá los números del cero al cinco:

```
<!--for($i=0;$i<5;$i++) { -->  
    El número es {$i} <br>  
<!-- } -->
```

De esta manera, cada template contendrá código HTML que puede ser editado por una persona sin conocimientos de HTML, a la vez que cuenta con el dinamismo necesario para generar la interfaz.

5.4 Modelos ROLAP estrella propuestos para reportes frecuentes

En este punto se presenta la información que los usuarios frecuentes necesitan consultar. Posteriormente se presentan los modelos de datos ROLAP estrella que entregan dicha información, y que permiten responder a consultas estadísticas sobre la misma.

5.4.1 Análisis de la información requerida

Los siguientes casos corresponden a los requerimientos de información que los usuarios del sistema necesitan, separados por proceso o entidad. Estos se obtuvieron a partir de entrevistas y reuniones realizadas con los usuarios que frecuentemente solicitan estadísticas.

5.4.1.1 Estadísticas de Inscripción

Estas estadísticas se refieren al proceso de inscripción académica. A continuación se detalla la granularidad y la información que necesitan conocer los usuarios que consultan estas estadísticas:

Granularidad: Por alumno, semestre y ramo.

Datos Requeridos:

1. Código del ramo
2. Sección del ramo
3. Rut del alumno
4. Nombres y apellidos del alumno
5. Fecha de nacimiento del alumno
6. Edad del alumno
7. Género o sexo del alumno
8. Estado civil del alumno
9. Nota del alumno en el ramo
10. Número de la inscripción del ramo por parte del alumno
11. Año y semestre de la inscripción.

5.4.1.2 Estadísticas de Docente

Estas estadísticas se refieren a los datos personales, de contratación y académicos de un docente. A continuación se detalla la granularidad y la información que necesitan conocer los usuarios que consultan estas estadísticas:

Granularidad: Por departamento, docente y año

Datos Requeridos:

1. Rut del docente
2. Nombres y apellidos del docente
3. Fecha de nacimiento del docente
4. Edad del docente
5. Género o sexo del docente
6. Estado civil del docente
7. Renta del docente
8. Jerarquía del docente
9. Horas contratadas semanales del docente
10. Tipo de contrato del docente
11. Año ingreso grado académico.
12. Grado académico del docente: Cada año ocurre una única actualización del grado académico de un docente.

13. Número de UD'S dictados por docente en plan común.
14. Número de memorias guiadas por docente
15. Número de memorias, por docente, en las cuales ha sido parte en la comisión.

5.4.1.3 Estadísticas de Alumnos

Estas estadísticas se refieren a los datos personales, antecedentes escolares y universitarios de un alumno. A continuación se detalla la granularidad y la información que necesitan conocer los usuarios que consultan estas estadísticas:

Granularidad: Por alumno y semestre

Datos Requeridos:

1. Rut del alumno
2. Nombres y apellidos del alumno
3. Fecha de nacimiento del alumno
4. Edad del alumno
5. Género o sexo del alumno
6. Estado civil del alumno
7. Semestre y año
8. Promedio ponderado PSU del alumno
9. Promedio ponderado de ingreso del alumno
10. Posición de ingreso del alumno.
11. Colegio de procedencia del alumno.
12. Tipo de colegio de procedencia del alumno.
13. Tipo de ingreso del alumno
14. Puntaje NEM de ingreso del alumno.
15. Región, comuna y ciudad de procedencia del alumno.
16. Año de ingreso del alumno.
17. Número de semestres cursados acumulados del alumno.
18. Número de UD'S inscritas en el semestre del alumno.
19. Número de UD'S aprobadas en el semestre del alumno.
20. Número de UD'S eliminadas en el semestre del alumno.
21. Número de UD'S reprobadas en el semestre del alumno.
22. Número de UD'S inscritas acumuladas del alumno.
23. Número de UD'S aprobadas acumuladas del alumno.
24. Número de UD'S eliminadas acumuladas del alumno.
25. Número de UD'S reprobadas acumuladas del alumno.
26. Número de cursos aprobados en el semestre del alumno.
27. Número de cursos reprobados en el semestre del alumno.
28. Número de cursos eliminados en el semestre del alumno.
29. Número de cursos aprobados acumulados del alumno.
30. Número de cursos reprobados acumulados del alumno.
31. Número de cursos eliminados acumulados del alumno.
32. Índice de aprobación por semestre del alumno.
33. Índice de aprobación acumulado del alumno.
34. Promedio de cursos aprobados acumulados del alumno.
35. Promedio ponderado por UDS de cursos aprobados acumulados por alumno.

5.4.1.4 Estadísticas de Cursos

Estas estadísticas se refieren a los datos de rendimiento de los cursos. A continuación se detalla la granularidad y la información que necesitan conocer los usuarios que consultan estas estadísticas:

Granularidad: Por docente, semestre y ramo.

Datos Requeridos:

1. Número de UDS del ramo
2. Código del ramo
3. Sección del ramo
4. Departamento del ramo.
5. Promedio de nota del ramo.
6. Docente del ramo.
7. Promedio de notas del ramo considerando alumnos aprobados del ramo
8. Número de alumnos inscritos en el ramo
9. Número de encuestas recibidas del ramo
10. Número de encuestas en blanco del ramo
11. Número de eliminados del ramo
12. Número de reprobados del ramo
13. Número de aprobados del ramo
14. Número de notas en blanco del ramo

5.4.1.5 Estadísticas de Carrera

En este caso, primer año y plan común corresponde a casos especiales de carreras, junto con las carreras o departamentos con que cuenta la Escuela de ingeniería y Ciencias. La información requerida y su granularidad es la siguiente:

Granularidad: Por alumno, carrera y año.

Datos Requeridos:

1. Año de ingreso a carrera del alumno.
2. Número de alumnos matriculados en la carrera.
3. Nombre de la carrera.
4. Número de alumnos de la carrera provenientes de la región metropolitana.
5. Número de alumnos de la carrera provenientes de regiones.
6. Número de alumnos de la carrera provenientes de colegios municipales.
7. Número de alumnos de la carrera provenientes de colegios subvencionados.
8. Número de alumnos de la carrera provenientes de colegios particulares pagados.
9. Número de alumnos de la carrera por tipo de ingreso deportistas.
10. Número de alumnos de la carrera por tipo de ingreso extranjero.
11. Puntaje promedio ponderado PSU alumnos de la carrera.
12. Máximo puntaje promedio PSU alumnos de la carrera.
13. Mínimo puntaje promedio PSU alumnos de la carrera.
14. Desviación estándar puntaje promedio ponderado PSU alumnos de la carrera.
15. Número de alumnos con crédito AFI en la carrera.
16. Número de alumnos con crédito universitario en la carrera.
17. Puntaje promedio ponderado ingreso alumnos de la carrera.

18. Máximo promedio ponderado ingreso alumnos de la carrera.
19. Mínimo promedio ponderado ingreso alumnos de la carrera.
20. Promedio NEM en la carrera.
21. Desviación estándar NEM en la carrera.
22. Alumnos con edad menor o igual a 18 años en la carrera.
23. Alumnos con edad igual a 19 años en la carrera.
24. Alumnos con edad igual a 20 años en la carrera.
25. Alumnos con edad igual a 21 años en la carrera.
26. Alumnos con edad igual a 22 años en la carrera.
27. Alumnos con edad igual a 23 años en la carrera.
28. Alumnos con edad igual a 24 años en la carrera.
29. Alumnos con edad igual o superior a 25 años en la carrera.
30. Número de alumnos hombres en la carrera
31. Número de alumnos mujeres en la carrera

5.4.2 Consideraciones para el diseño de un modelo de datos ROLAP estrella

Tanto la experiencia empírica como bibliográfica entregan recomendaciones o buenas prácticas, que deben ser consideradas en el diseño físico de modelos de datos ROLAP estrella, algunas de ellas son las siguientes:

1. **No incluir tuplas nulas en la tabla de hechos:** Con esto se refiere a referenciar tuplas que no existen en las dimensiones desde la tabla de hechos. Un problema de tuplas nulas pone en duda la confiabilidad de los datos, por cuanto se entiende que no se ha realizado un control ni un chequeo de integridad referencial efectivo.

Por otra parte, el problema de tuplas nulas podría arrojar resultados incoherentes o incorrectos al momento de realizar agregación de la información, por ejemplo contar el número de tuplas que satisfacen una determinada condición o bien incluir tuplas que no deberían considerarse.

2. **No incluir valores “no ocurre” en la tabla de hechos:** Las razones que justifican evitar estas situaciones son similares al punto anterior. Sin embargo, es posible agregar estas referencias en ciertos casos, por ejemplo, antecedentes personales incompletos de algunas personas.
3. **Incluir los campos de texto en las tablas dimensiones:** En general no es recomendable almacenar datos de este tipo, puesto que son de largo variable y pueden reducir el rendimiento de los motores de bases de datos al desplazarse recorriendo las filas. De ser necesario, es recomendable agregarlo como campo de una dimensión, puesto que el número de tuplas de los hechos es geoméricamente mayor que una dimensión.
4. **Los valores de las columnas no deben estar codificados:** Tanto el nombre de las columnas como el valor que pueden tomar deben ser desplegados al usuario en un lenguaje natural, evitándose codificaciones. En este sentido, valores como 1 ó 0 para representar el cumplimiento o no cumplimiento de una determinada condición deberían ser presentados con un “Sí” o un “No”.

En este trabajo se ha preferido almacenar los valores directamente en las tablas por sobre convertirlos antes de desplegar la información al usuario, privilegiando contar con tablas auto-explicativas; el espacio extra requerido generalmente es despreciable.

5. **Una fila en los hechos corresponde a una medida:** Al fijar la granularidad de una tabla de hechos, esta entregará una única fila. Cada fila debe contener medidas que estén relacionadas solo con esta granularidad y con ninguna otra, de no ser así, se podría suponer mezcla de granularidades.
6. **Valores constantes deben almacenarse en las dimensiones:** Si se considera un ejemplo en el cual un alumno ingresa a una universidad con ciertos puntajes obtenidos en las diferentes pruebas rendidas en la PSU (Prueba de selección universitaria) y se quiere conocer la distribución de los alumnos de la carrera de Ingeniería Civil en Computación a través del tiempo por puntaje ponderado de ingreso por PSU, podría pensarse en almacenar los puntajes obtenidos por cada alumno en la tabla de hechos, sin embargo, estos valores son independientes de otras dimensiones y corresponden a constantes propias de cada alumno, razón por lo cual deberían constituir atributos de la dimensión Alumno. Incluir estos atributos como campos de hechos podría aumentar la utilización de espacio en disco, puesto que los hechos cuentan con un número de tuplas geoméricamente mayor al de sus dimensiones.
7. **Todas las medidas en la tabla de hechos deben tener la misma granularidad:** Este punto se refiere a no incluir en una misma dimensión información cuya granularidad es distinta, por ejemplo, datos históricos anuales con semestrales. También se refiere a no mezclar granularidades similares pero distintas en la tabla de hechos, por ejemplo, una tabla de hechos cuya granularidad es por Alumno, Ramo y Semestres combinada con otra cuya granularidad es por Docente, Ramo y Semestre, el resultado sería una granularidad por Alumno, Docente, Ramo y Semestre donde existirían medidas asociadas al alumno y al docente, esto crearía una confusión respecto a cual es la granularidad real de los hechos.
8. **Las medidas en los hechos deben ser numéricas, aditivas y continuas:** La continuidad se requiere básicamente porque ayuda a determinar que atributos pertenecen a una dimensión y cuales a un hecho.

La aditividad se refiere a que cada medida pueda ser sumada a través de todas las dimensiones, esto debido a que las funciones agregadas están basadas principalmente en la suma, como es el caso del promedio, la desviación estándar, la varianza, entre otros. Una medida semi-aditiva difícilmente tendrá validez o coherencia al agrupar los hechos por algunas dimensiones.

Entonces, dado que la tabla de hechos es usada para calcular operaciones agregadas sobre sus medidas, resulta recomendable que estas sean numéricas, continuas y aditivas puesto que al realizar consultas sobre miles o incluso millones de registros estos podrán ser condensados en unas cuantas tuplas que finalmente se desplegarán directamente al usuario.

Lamentablemente no en todos los casos las medidas pueden ser del tipo aditivas o incluso semi-aditivas, en esos casos podría ser recomendable almacenar esas medidas en una dimensión.

9. **Mantener la independencia entre atributos de diferentes dimensiones:** Este es un punto relevante para determinar que atributos deben separarse o almacenarse en dimensiones distintas. Dos campos independientes deben separarse, por ejemplo, el nombre de una persona podría almacenarse en una misma dimensión con su género. Por lo general la independencia de dos campos está basada en información estadística o bien en modelos de datos relacionales pre-existentes.
10. **Es valido agregar nuevas dimensiones a los hechos que no cambien su granularidad:** Estas nuevas dimensiones agregan información extra a la granularidad de la tabla de hechos, por ejemplo, los datos demográficos de una persona en un semestre almacenados en una mini-dimensión, agregan información adicional y no cambian la granularidad por Persona y Semestre.
11. **Evitar la normalización por tercera forma normal:** Al normalizar una dimensión en tercera forma normal esperando utilizar menos espacio, el impacto real de esta acción es mínima (la reducción de espacio es típicamente menor al 1% [12.4 Libros 1]). Por otra parte, al normalizar una tabla, el rendimiento al realizar Browsing sobre los campos disminuye, debido a las uniones que es necesario realizar para cruzar las diferentes tablas a fin de obtener los distintos valores.
12. **La redundancia de información es válida:** Existen casos en los cuales la información redundante permite mejorar el rendimiento de las consultas, por ejemplo, almacenar los datos demográficos en una dimensión distinta junto con almacenarlos en la dimensión Persona; al filtrar por la dimensión Demografía se obtendría una selectividad mayor, dado que muchas personas compartirán sus datos demográficos. Al mantener los datos demográficos también en Persona se mantiene la completitud de la información personal.
13. **Separar dimensiones con atributos compartidos en dos dimensiones distintas:** En ciertos casos una dimensión podría contener información incompleta sobre datos de Clientes, en estos casos podría convenir separar los datos incompletos de los clientes en una nueva dimensión en la cual se establezca debidamente una tupla “no definido” y en otra dimensión los datos completos de los clientes.
14. **Evitar tablas de hechos con un gran número de dimensiones:** Contar con una tabla de hechos con un gran número de dimensiones es una mala idea por una serie de razones, entre las cuales se encuentra el dramático crecimiento en la utilización de disco, el gran número de uniones necesarias dado el número de dimensiones y una consiguiente disminución del rendimiento. Existe una menor posibilidad de indexación de las llaves de dimensiones.

En la práctica, las tablas de hechos no tienen más de 15 dimensiones y en general el número promedio de dimensiones distintas en una tabla de hechos es de 4 ó 5 [12.4 Libros 1].

15. **Utilizar atributos como dimensiones es una mala decisión:** Es una mala decisión al igual que el punto anterior, puesto que el número de dimensiones en la tabla de hechos puede crecer innecesariamente.

5.4.3 Modelos de datos propuestos

A continuación se presentan los modelos de datos ROLAP estrella. En su diseño se han considerado las limitaciones del sistema, el marco teórico descrito en la sección 5.4.2 Consideraciones para el diseño de un modelo de datos ROLAP estrella y la bibliografía consultada.

5.4.3.1 Estadísticas de Inscripción

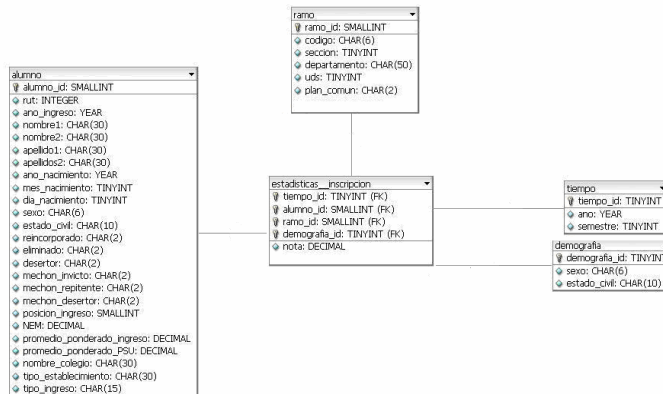


Fig 5.18 Modelo de datos ROLAP estrella de Estadísticas de Inscripción

Objetivo: Este modelo pretende responder consultas relacionadas con las notas y ramos inscritos por un alumno en un semestre. También es posible conocer antecedentes de estos alumnos, como sus puntajes en la PSU, algunos datos personales, etc.

Descripción: Cada ramo se caracteriza por tener un código, una sección, un número de UD'S y pertenecer a un único departamento. El campo plan_comun indica si el ramo es parte del plan de estudios del plan común, con el valor 'Si' o 'No'.

La tabla dimensión Alumno, no solo almacena los datos personales de cada alumno, también contiene algunos campos que entregan información sobre su desempeño a lo largo de su vida universitaria. Los campos mechon_invicto, mechon_repitente, mechon_desertor, eliminado, reincorporado y desertor contendrán los valores "SI" o "NO" dependiendo si el alumno aprobó todos sus ramos el primer año, si repitió completamente el primer año o primer semestre (dependiendo del año de ingreso, de acuerdo al plan de estudio según el cual ingreso), abandonó la Facultad por su propia decisión, fue eliminado en algún semestre y si fue reincorporado o desertor en algún semestre. También contendrá información sobre sus puntajes de ingreso a la universidad como también su NEM (notas enseñanza media). Por último, también contendrá información sobre el colegio, el tipo de ingreso y la posición de ingreso.

A partir de la fecha de nacimiento de los alumnos es posible determinar su edad, por esta razón no se incluye la edad directamente en el modelo, puesto que una fecha de nacimiento tiene una granularidad distinta a la tabla de hechos (Semestral).

La dimensión Demografía es una mini-dimensión que no cambia la granularidad de la tabla de hechos, puesto que va relacionada con cada Alumno. Esta dimensión mejora el rendimiento al realizar filtros por sexo o estado civil.

El único campo en la tabla de hechos es la nota obtenida por el alumno en el ramo, a pesar de ser no-aditiva se ha almacenado en esta tabla puesto que dependen de las dimensiones y no se justificaría definir una nueva dimensión solo para contener la nota.

Una dimensión relevante y presente en la mayoría de los modelos es el Tiempo, la razón de esto son los estudios y análisis que se realizan a lo largo de meses, semestres o años; no es común un modelo sin alguna dimensión de Tiempo. Al ser una dimensión es posible definir su granularidad en un solo lugar y agregar una detallada descripción, como por ejemplo días laborales, feriados, fines de semana, último día del mes, etc. así se evita tener más de un campo en la tabla de hechos para definirlo.

En esta dimensión no se utiliza el tipo de datos de fecha y tiempo disponible en los motores de bases datos, puesto que estos no los manejan eficientemente al realizar uniones. Algunos motores incluso no los indexan.

5.4.3.2 Estadísticas de Docente

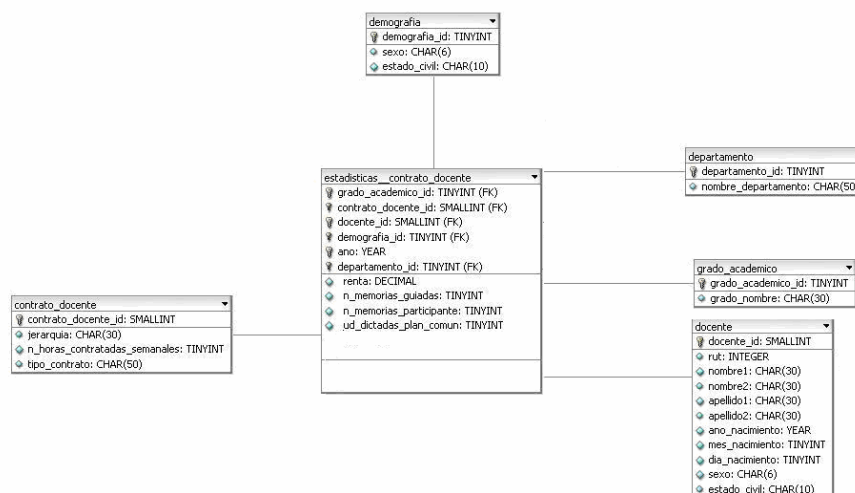


Fig 5.19 Modelo de datos ROLAP estrella de Estadísticas de Docente

Objetivo: Este modelo responde a las consultas sobre los grados académicos y el contrato docente para un académico, por departamento y por año. Contiene información resumen de los cursos y horas docentes dedicadas por el académico.

Descripción: La dimensión Departamento, solo tiene el campo no numérico nombre_departamento, por esta razón se ha creado como una nueva dimensión y no como una dimensión degenerada. Si el nombre del departamento fuera un campo de texto dentro de la llave compuesta de los hechos la utilización de espacio en disco sería mayor junto con disminuir los tiempos de respuesta a las consultas, debido a las uniones entre las tablas por campos de texto. La alta selectividad y baja cardinalidad, sumado a que un docente puede pertenecer a uno o más departamentos son buenas razones para justificarla como una dimensión.

La dimensión Grado Académico, presenta una situación similar, contiene solo el grado académico asociado al docente en el año. Dado que el grado académico cambia solo una vez por año estará asociado a la dimensión degenerada del tiempo (el campo año en la tabla de hechos).

La dimensión Contrato Docente contiene campos altamente relacionados con el contrato del docente en el departamento del cual forma parte, como por ejemplo el número de horas semanales contratadas, el tipo de jerarquía (profesor titular, académico jornada completa, etc) y el tipo de jornada (completa, parcial, etc).

En la dimensión Docente se encuentran los datos personales del profesor, como su rut y nombres, la fecha de nacimiento y sus datos demográficos. Los datos demográficos también se encuentran en la mini-dimensión Demografía, que no modifica la granularidad del modelo, puesto que está asociada al Docente.

La tabla de hechos solo contiene campos aditivos, como la renta percibida por el docente y algunos datos pre-calculados que resumen su historia docente hasta el año considerado, como por ejemplo, el número de memorias dirigidas, el número de UD's dictadas en plan común, etc.

5.4.3.3 Estadísticas de Alumno

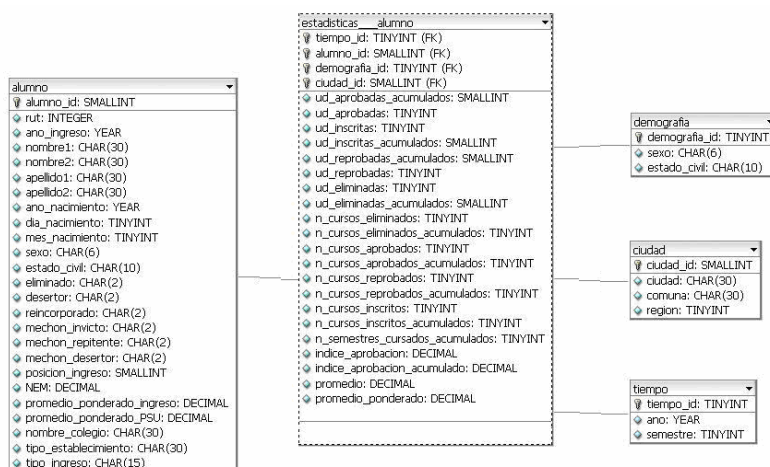


Fig 5.20 Modelo de datos ROLAP estrella de Estadísticas de Alumno

Objetivo: Este modelo contiene los indicadores de rendimiento para los alumnos por semestre. Es posible conocer el desempeño del alumno a lo largo del tiempo y obtener información personal más detallada, como la ciudad o comuna donde vive.

Descripción: Las dimensiones Alumno, Tiempo, y Demografía corresponden a las mismas analizadas anteriormente. La dimensión Ciudad, sin embargo, permite obtener información de los alumnos de acuerdo a su distribución geográfica en el país. Tanto la dimensión Ciudad como Demografía no modifican la granularidad del modelo, puesto que están asociadas al Alumno.

La tabla de hechos cuenta con campos aditivos y no aditivos. Los campos relacionadas con las unidades docentes aprobadas, eliminadas, reprobadas e inscritas, junto con el número de cursos eliminados, reprobados, inscritos y aprobados corresponden a los campos

aditivos. Todos ellos considerando tanto semestralmente como acumulado desde su inicio como alumno.

Los campos no aditivos corresponden a los promedios y al índice de aprobación. Los campos promedio ponderado e índice de aprobación no pueden ser cálculos utilizando el sistema, razón por la cual son almacenados en esta tabla directamente. Si bien el promedio de cursos aprobados podrían ser obtenidos a partir del cálculo del promedio sobre la nota de los cursos aprobados, los datos necesarios para esto se encuentran en otro modelo.

5.4.3.4 Estadísticas de Curso

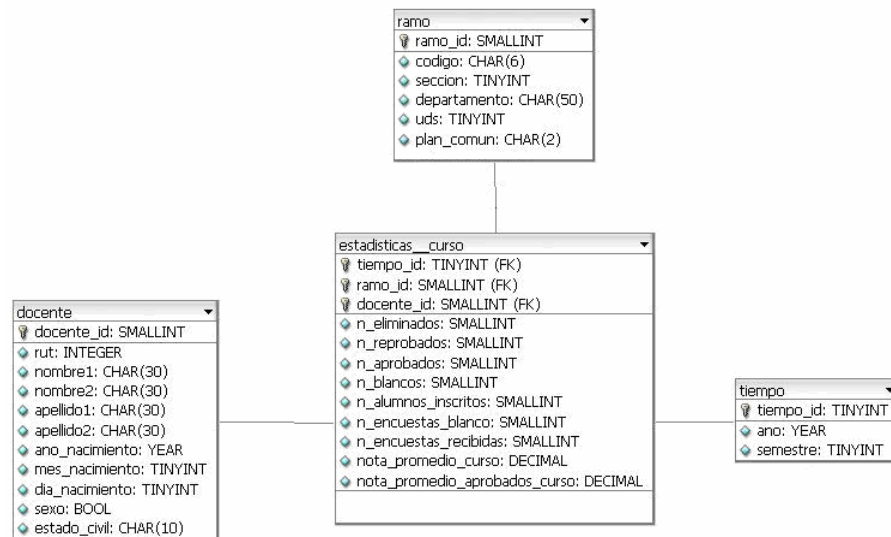


Fig 5.21 Modelo de datos ROLAP estrella de Estadísticas de Curso

Objetivo: Este modelo responde a consultas sobre los ramos dictados por un docente en un semestre y sobre sus antecedentes. Resume la información sobre el resultado final del curso, como el promedio del curso, el número de reprobados, encuestas docentes enviadas, etc.

Descripción: Las dimensiones Tiempo, Ramo y Docente son similares en estructura y significado a como fueron descritas anteriormente, razón por la cual no se realizará una explicación más detallada de las mismas.

La tabla de hechos cuenta solo con dos campos no aditivos, la nota promedio del curso y la nota promedio del curso calculada sobre los alumnos que aprobaron. Al igual que en modelos anteriores, estos promedios podrían ser calculadas como el promedio sobre la nota obtenida en el curso por todos los alumnos que tienen una nota y sobre los que tienen nota mayor o igual a 4.0, sin embargo, dichas notas se encuentran en el modelo **Estadísticas de Inscripción**.

La granularidad del modelo **Estadísticas de Inscripción** es por Alumno, Semestre y año (Tiempo) y los hechos determinan la nota de un alumno; la granularidad de este modelo es por Ramo, Docentes y Semestre (Tiempo) y determina los hechos de un curso. De haber unido ambos modelos, por cada alumno habría uno o varios profesores, mezclándose los datos del curso con los del alumno.

Con respecto a los hechos aditivos se cuenta con el número de alumnos reprobados, aprobados, que eliminaron el curso y con nota en blanco. También se tiene el número de alumnos que inscribieron el curso, el número de encuestas docentes enviadas y el número de ellas enviadas en blanco.

5.4.3.5 Estadísticas de Carrera

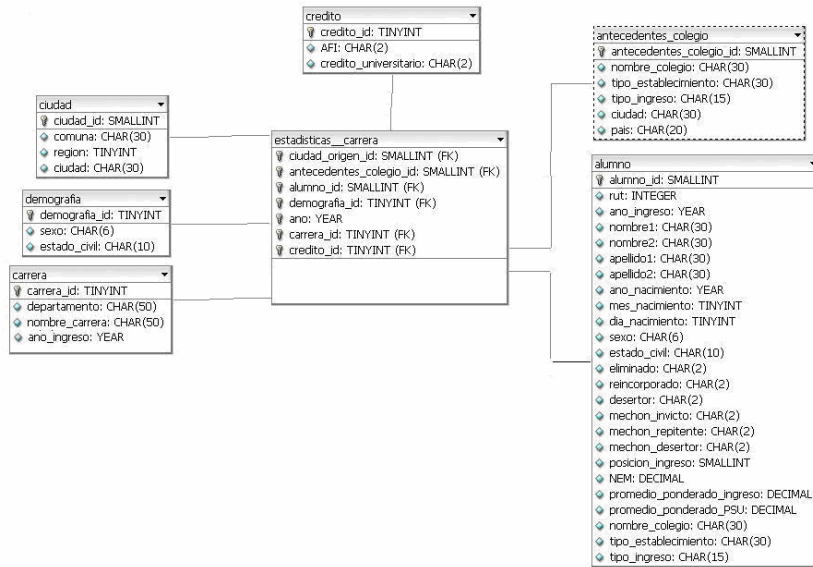


Fig 5.22 Modelo de datos ROLAP estrella de Estadísticas de Carrera

Objetivo: El objetivo de este modelo es permitir conocer los antecedentes académicos y escolares, de financiamiento y origen de los alumnos, por carrera y a lo largo del transcurso de ella.

Descripción: Las dimensiones Demografía, Alumno y Ciudad son similares en estructura y significado a como fueron explicadas anteriormente, por lo cual no requieren una descripción más detallada.

La dimensión Tiempo por su parte corresponde a una dimensión especial puesto que se encuentra degenerada en la tabla de hechos, esto debido a que su granularidad es anual y por lo tanto se hace necesario definirla como una dimensión normal.

La dimensión Crédito permite conocer los créditos que dispone el alumno en el año, de esta manera cuenta con campos que indican con un “Si” o un “No” si el alumno cuenta con el crédito de aporte fiscal indirecto (AFI) o el crédito universitario.

La dimensión Antecedentes Colegio contiene algunos datos en común con la dimensión Alumno, como son los campos tipo de ingreso (Deportista, PAA o PSU, Alumno de intercambio, etc), tipo de establecimiento (Técnico profesional, científico humanista, etc) y el nombre del colegio. Otros campos con que cuenta son la ciudad y el país, con estos datos podrá filtrarse a los alumnos chilenos de regiones, santiaguinos o a los extranjeros.

Las dimensiones Crédito, Antecedentes Colegio, Demografía y Ciudad no agregan cambios a la granularidad del modelo, esto debido a que están relacionadas con el Alumno en un año.

La dimensión Carrera determina a las generaciones (año de ingreso) de una carrera de un departamento. El campo año de ingreso no se almacena como un hecho puesto que no es un campo aditivo. Al almacenarlo en esta dimensión se tendrán agrupados a todos los alumnos de una misma generación de una carrera. Esta dimensión no contará con un gran número de tuplas, por que el número de carreras es bajo, al igual que el número de generaciones.

A priori no es posible conocer el número de carreras que podría estudiar un alumno, sin embargo, sería posible definir un nuevo campo en la dimensión Alumno por cada carrera por la cual haya pasado un alumno, sin embargo con esto se perdería información sobre el tiempo en el cual estuvo en ella.

5.4.4 Automatización del diseño de modelos de datos ROLAP estrella

En el presente trabajo se realizaron intentos por automatizar el diseño físico de los modelos de datos ROLAP estrella. El objetivo era obtener una lista de dimensiones con sus campos a partir de una lista de filas y campos obtenidos desde las fuentes externas. Para lograr este objetivo se consideraron los siguientes puntos.

1. Separar los campos en dos grupos: Cadenas y números.
2. Calcular la cardinalidad de cada uno de los campos.
3. Realizar test de independencia Chi-cuadrado o Fisher para variables cualitativas.

Sin embargo, los resultados obtenidos no fueron coherentes dado que la obtención de un modelo de datos efectivo, depende de variables propias de la definición y utilización del modelo de datos relacional original de la fuente externa. A su vez también depende de la granularidad de variables, como el tiempo que deben ser definidas por el usuario dependiendo del proceso de negocio.

Cualquier intento de ingeniería inversa que no haga uso de la información del catalogo del modelo de datos original, se vuelve complejo y poco útil. Los trabajos en el área de la automatización son informales, la mayoría de ellos solo utilizan guidelines o están basados en reglas. En trabajos como [12.7 Paper's 4] y [12.7 Paper's 5] sin embargo, se presentan algoritmos y variables relevantes a considerar, algunas de ellas son:

1. Los campos numéricos y aditivos, que no sean llaves primarias en relaciones **n:m** del modelo relacional son candidatos a ser almacenadas en la tabla de hechos.
2. Reconocer tres tipos de datos: String, números y fechas. El resto se ataca recursivamente.
3. Analizar los valores de medidas en función del tiempo y dado las necesidades del negocio.
4. Encontrar hechos basados en la entidad actualizada con mayor frecuencia. Los valores numéricos constantes o discretos son candidatos a ser almacenados en dimensiones. Valores numéricos con cambios constantes o muy repetidos deberían ser almacenados en las tablas de hechos.
5. Las relaciones **n:m** entre tablas deben ir en una tabla de hechos, puesto que cuentan con llaves compuestas. Esto entrega un principio, cada tabla con llaves primarias compuestas es una tabla de hechos.

Utilizando estas y otras consideraciones, el algoritmo descrito en [12.7 Paper's 4] sigue los siguientes pasos:

1. Ordenar las entidades de acuerdo al número de columnas numéricas en orden descendente. Ordenarlos descendientemente mejora el resultado del algoritmo.
2. Crear atributos numéricos de la entidad en dimensiones.
3. Identificar los campos de fecha.
4. Si aun quedan campos por definir se crea un nuevo nivel.
5. Analizar recursivamente las relaciones de las entidades para agregar más niveles.

Posteriormente a la aplicación de estos pasos, se aplica un refinamiento que realiza los siguientes pasos:

1. Verificar que todos los campos definidos como medidas lo sean.
2. Realizar cálculos comunes y almacenarlos en una nueva columna.
3. Eliminar campos innecesarios.

Sin embargo, ningún trabajo estudiado elimina la componente humana del todo y el hecho de acceder al catalogo del modelo de datos, lo que hace que su aplicación en nuestro trabajo pierda la motivación inicial, por lo cual no se aborda.

6 Implementación

En la implementación del sistema SIAGER se utilizaron librerías, funciones y clases, además de archivos de script PHP y plantillas HTML. Algunas de las librerías fueron desarrolladas especialmente para este trabajo, otras en cambio se encuentran disponibles en el repositorio de funciones con que cuenta el ADI, conocido bajo el nombre de LIB_PHP.

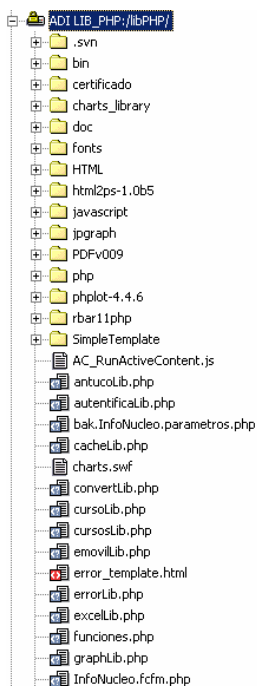


Fig 6.1 Carpeta LIB_PHP

6.1 Archivos template

Cada archivo template está asociado a su controlador del mismo nombre y corresponden a los archivos que cuentan con los códigos HTML para generar las interfaces de usuario. Los templates con que cuenta el sistema son los siguientes:

1. **Campos_visible.html:** Es el archivo HTML asociado al controlador *campos_visibles.php*. Este archivo muestra los campos que el usuario podrá ver en la interfaz de consultas.
2. **CrearReporte.html:** Es el archivo HTML asociado al controlador *crearReporte.php*. Este archivo contiene la interfaz que el usuario podrá ver para crear un reporte ROLAP estrella.
3. **Funcion.html:** Es el archivo HTML asociado al controlador *funcion.php*. Este archivo contiene la interfaz que el usuario podrá ver al acceder al módulo de consultas.
4. **Index.html:** Es un archivo HTML asociado al controlador *index.html*. Este archivo contiene la interfaz que muestra los reportes disponibles.

6.2 Scripts del sistema

Estos corresponden a los controlares de la capa lógica y son los archivos PHP con que cuenta el sistema. Estos archivos se encargan de controlar la interfaz de usuario e interactuar con la capa de datos, a continuación se explican cada uno de ellos:

1. **Campos_visibles.php:** Genera la interfaz de campos visibles
2. **Cargador.php:** Rescata la información desde la BD local y la carga en sesión, posteriormente redirecciona al módulo de consultas o de creación del modelo ROLAP estrella si corresponde.
3. **Config.php:** Archivo de configuración del módulo. En este archivo se deben declarar todas las variables que tengan que ver con la configuración del servidor donde está corriendo el módulo. Una de las líneas más importante y con la cual se incluye el sistema de parámetros del módulo es la siguiente:

```
include_once( 'InfoNucleo.php' )
```

4. **Index.php:** Genera la interfaz de inicio del módulo, donde se pueden ver los reportes disponibles.
5. **Consultas.php:** Controla las consultas estadísticas guardadas, actualizadas y eliminadas.
6. **crearReporte.php:** Genera la interfaz de creación del modelo ROLAP estrella.
7. **crearReporte2.php:** Crea el modelo de ROLAP estrella al ser llamado por *crearReporte.php*.
8. **Funcion.php:** Este archivo es llamado cuando se accede al módulo de consultas. Este archivo se encarga de ejecutar las consultas, de actualizar los reportes ROLAP estrella, de controlar la interfaz de consultas y de llamar a los archivos *doc.php*, *pdf.php*, *rtf.php* y *Excel.php*. Además puede exportar los resultados de la consulta a un archivo Excel que no incluya imágenes.
9. **doc.php:** Exporta las consultas a un archivo WORD, estos reportes pueden incluir tanto gráficos como tablas.
10. **pdf.php:** Exporta las consultas a un archivo PDF, estos reportes pueden incluir tanto gráficos como tablas.
11. **rtf.php:** Exporta las consultas a un archivo RTF, estos reportes pueden incluir tanto gráficos como tablas.
12. **excel.php:** Exporta las consultas a un archivo EXCEL, estos reportes pueden incluir tanto gráficos como tablas.
13. **Graficar.php:** Realiza los gráficos Flash, este archivo es incluido en *funcion.php*

6.3 Funciones

Las funciones también forman parte de la capa de lógica del sistema, puesto que controlan la generación de gráficos, permiten exportar los resultados en diferentes formatos, entre otras tareas. A continuación se explican cada una de ellas de acuerdo a la función que desempeña.

6.3.1 Funciones de gráficos

Para la generación de gráficos se dispone de funciones genéricas las cuales se encuentran en el archivo *funciones_graficos.php* en la carpeta *include* del sistema. Estas funciones son las siguientes:

6.3.1.1 String graficar(\$datos,\$stipo='barras',\$tit='',\$_l_y='',\$_l_x='', \$sw='50%', \$sh= 450)

Retorna el código HTML para insertar un gráfico flash en una página Web. Sus parámetros son los siguientes:

- \$datos: Los datos del gráfico
- \$stipo: El tipo de gráfico, que pueden ser:
 - torta
 - barras
 - lineas_juntas
 - barras_juntas
- \$tit: El título del gráfico
- \$_l_x: Título del eje X
- \$_l_y: Título del eje Y
- \$width: Ancho del gráfico en porcentaje
- \$height. El alto del gráfico

El formato de una salida es el siguiente:

```
<div style="text-align:center">
<object width="50%" height="450" id="chart_2039576836">
<param name="movie"
value="?chart=2039576836&accion=swf&data=%3Fchart%3D2039576836%26accion%3Ddatos%26rand%3D1073969494" />
<embed src="?chart=2039576836&accion=swf&data=%3Fchart%3D2039576836%26accion%3Ddatos%26rand%3D1988875680"
width="50%" height="450">
</embed>
</object>
</div>
```

6.3.1.2 Array torta_agrupar(\$datos,\$datos_form,\$columnas,\$pagina)

Retorna los datos necesarios para generar un gráfico de torta para la opción agrupar desde el módulo de consultas. Sus parámetros son los siguientes:

- \$datos: Corresponde al resultado de ejecutar la consulta
- \$datos_form: Corresponde a los parámetros de la consulta del usuario
- \$columnas: Corresponde a los datos de las columnas del reporte ROLAP estrella sobre el cual se ejecuta la consulta

- \$pagina: El número de página de la consulta. Gracias a este número es posible saber cuales son los números de filas que corresponde desplegar.

Retorna un arreglo asociativo de la forma siguiente:

```

Array
(
  [datos] => Array
  (
    [Fila 4] => 0.2
    [Fila 5] => 0.2
    [Fila 6] => 0.3
    [Fila 7] => 0.4
    [Fila 8] => 0.3
    [Fila 9] => 0.2
    [Fila 10] => 0.3
    [Fila 11] => 0.2
    [Fila 14] => 0.2
    [Fila 15] => 0.2
    [RESTO DATOS] => 97.5
  )

  [titulo] => Agrupados por (dia del mes y mes del año)
  [campos] => Array
  (
    [dia del mes] => dia del mes
    [mes del año] => mes del año
  )
)

```

La descripción para cada una de las claves de este arreglo son las siguientes:

- titulo: Corresponde al título del gráfico.
- campos: Es un arreglo de los campos agrupados
- datos: Contiene los porcentajes que serán graficados en una torta. En este caso se grafica cada fila con un porcentaje distinto de cero. Si la suma de porcentajes no es cien por ciento se crea un nuevo valor llamado 'RESTO DATOS'.

6.3.1.3 Array torta_transponer(\$datos,\$datos_form,\$columnas,\$pagina)

Retorna los datos necesarios para generar un gráfico de torta para la opción transponer desde el módulo de consultas. Sus parámetros son los siguientes:

- \$datos: Corresponde al resultado de ejecutar la consulta
- \$datos_form: Corresponde a los parámetros de la consulta del usuario
- \$columnas: Corresponde a los datos de las columnas del reporte ROLAP estrella sobre el cual se ejecuta la consulta
- \$pagina: El número de página de la consulta. Gracias a este número es posible saber cuales son los números de filas que corresponde desplegar.

Retorna un arreglo asociativo de la forma siguiente:

```

Array
(
  [datos] => Array

```

```

(
  [1] => Array
  (
    [Fila 1] => 0.4
    [Fila 2] => 4.4 )
[10] => Array
  (
    [Fila 1] => 4.4
    [Fila 2] => 9.1
  )
)

[campos] => Array
(
  [dia del mes] => dia del mes
)

```

La descripción para cada una de las claves de este arreglo son las siguientes:

- campos: Es un arreglo con los campos agrupados
- datos: Contiene un arreglo con los porcentajes por cada columna transpuesta. El valor de la clave es el valor del campo transpuesto.

6.3.1.4 Array `lineas_juntas_transpuestas($datos,$datos_form,$columnas)`

Retorna los datos necesarios para generar un gráfico de líneas para la opción transponer desde el módulo de consultas. Sus parámetros son los siguientes:

- \$datos: Corresponde al resultado de ejecutar la consulta
- \$datos_form: Corresponde a los parámetros de la consulta del usuario
- \$columnas: Corresponde a los datos de las columnas del reporte ROLAP estrella sobre el cual se ejecuta la consulta

Retorna un arreglo asociativo de la forma siguiente:

```

Array
(
  [datos] => Array
  (
    [(562)] => Array
    (
      [Columna Transpuesta 1] => 3
      [Columna Transpuesta 2] => 2
    )
  )
  [titulo] => Distribuido por (rut del profesor)
  [campos] => Array
  (
    [TDOC_RUT_PROFESOR_TESIS] => ' rut del profesor'
  )
)

```

Cada una de las claves del arreglo datos corresponde a una línea que será graficada a lo largo de cada uno de los valores de las columnas transpuestas. En el ejemplo anterior, la línea asociada al rut 562 toma los valores tres y dos en las columnas transpuestas.

La descripción para cada una de las claves de este arreglo son las siguientes:

- campos: Es un arreglo con los campos agrupados
- titulo: Es el título del gráfico

6.3.1.5 Array barras_agrupadas(\$datos,\$datos_form,\$columnas,\$pagina)

Retorna los datos necesarios para generar un gráfico de barra para la opción agrupar desde el módulo de consulta. Sus parámetros son los siguientes:

- \$datos: Corresponde al resultado de ejecutar la consulta.
- \$datos_form: Corresponde a los parámetros de la consulta del usuario.
- \$columnas: Corresponde a los datos de las columnas del reporte ROLAP estrella sobre el cual se ejecuta la consulta
- \$pagina: El número de página de la consulta. Gracias a este número es posible saber cuales son los números de filas que corresponde desplegar.

Retorna un arreglo asociativo de la forma siguiente:

```
Array
(
  [datos] => Array
  (
    [Fila 1] => 9
    [Fila 2] => 5
    [Fila 3] => 6
    [Fila 4] => 93
    [Fila 5] => 73
    [Fila 6] => 116
    [Fila 7] => 149
    [Fila 8] => 112
    [Fila 9] => 71
    [Fila 10] => 124
    [Fila 11] => 97
    [Fila 12] => 2
    [Fila 13] => 2
    [Fila 14] => 90
    [Fila 15] => 85 ) [titulo] => Agrupados por (dia del mes y mes del año)
  [labels] => Filas agrupadas
  [campos] => Array
  (
    [dia del mes] => dia del mes
    [mes del año] => mes del año
  )
)
```

La descripción para cada una de las claves de este arreglo son las siguientes:

- datos: Los cantidad de datos agrupados para cada fila
- titulo: El título del gráfico
- campos: Arreglo con los campos agrupados.
- labels: El nombre del eje X

6.3.1.6 Array `lineas_juntas_agrupadas($datos,$datos_form,$columnas)`

Retorna los datos necesarios para generar un gráfico de líneas para la opción agrupar desde el módulo de consultas. Sus parámetros son los siguientes:

- `$datos`: Corresponde al resultado de ejecutar la consulta
- `$datos_form`: Corresponde a los parámetros de la consulta del usuario.
- `$columnas`: Corresponde a los datos de las columnas del reporte ROLAP estrella sobre el cual se ejecuta la consulta

Retorna un arreglo asociativo de la forma siguiente:

```
Array
(
    [titulo] => Distribuidos por (dia del mes)
    [datos] => Array
        (
            [(1)] => Array
                (
                    [1] => 9
                    [2] => 5
                )
            [(2)] => Array
                (
                    [1] => 97
                    [2] => 2
                )
        )
    [labels] => mes del año
    [campos] => Array
        (
            [dia del mes] => dia del mes
            [0] => mes del año
        )
)
```

La descripción para cada una de las claves de este arreglo son las siguientes:

- `label`: El nombre del eje x
- `campos`: Los campos agrupados
- `titulo`: El título del gráfico
- `datos`: Cada clave del arreglo define una línea distinta que es graficada a lo largo de los valores de ese arreglo.

6.3.1.7 Array `tabla_simple($datos,$columnas,$trans,$pagina)`

Retorna una tabla simple que es utilizada cuando no se realiza una agrupación en el modelo de consultas. Sus parámetros son los siguientes:

- `$datos`: Corresponde al resultado de ejecutar la consulta
- `$columnas`: Corresponde a los datos de las columnas del reporte ROLAP estrella sobre el cual se ejecuta la consulta
- `$trans`: Indica si es una tabla transpuesta o no
- `$pagina`: El número de la página

Retorna un arreglo asociativo de la siguiente forma:

```
Array
(
  [0] => Array
    (
      [0] => N°
      [1] => día del mes
      [2] => mes del anyo
    )
  [1] => Array
    (
      [0] => 1
      [1] => 30
      [2] => 6
    )
  [2] => Array
    (
      [0] => 2
      [1] => 30
      [2] => 6
    )
)
```

6.3.1.8 Array graficar_export(\$dat,\$d_form,\$col,\$pag,\$preTitulo="", \$path="")

Esta función se encarga de generar las imágenes JPEG utilizadas en los archivos Excel, WORD, RTF y PDF exportados como reportes finales. Internamente realiza llamados a las funciones de gráficos anteriores.

Sus parámetros son los siguientes:

- \$dat: Corresponde al resultado de ejecutar la consulta
- \$d_form: Corresponde a los parámetros de la consulta del usuario.
- \$col: Corresponde a los datos de las columnas del reporte ROLAP estrella sobre el cual se ejecuta la consulta
- \$pag: El número de la página
- \$preTitulo: Corresponde a un texto que es usado como prefijo en los títulos de los gráficos.
- \$path: Cambia la ruta donde se almacena la imagen.

Retorna un arreglo asociativo de la siguiente forma:

```
Array
(
  [lineas_agrupar] => Array
    (
      [path] => Array
        (
          [0] => /tmp/fm_4c58db4856363ebf1095a75aed96b1bd.jpeg
        )
    )
  [barras_verticales_agrupar] => Array
    (
      [path] => Array
```



```
(
  (
    [0] => /tmp/fm_1f27aac7b34a028557dd25f8f4dde8d.jpeg
  )
)
)
```

En este ejemplo se generan dos tipos de gráficos, uno de líneas y otro de barras, sus imágenes son almacenadas en un archivo jpeg, ubicadas físicamente en el archivo definido en el sub-arreglo path

6.3.2 Funciones de consulta

Estas funciones se encargan de ejecutar las consultas estadísticas ejecutadas por el usuario del sistema. Estas funciones se encuentran implementadas en el archivo *funciones_reportes.php* y son las descritas a continuación.

6.3.2.1 Array getConsulta_agregadas(\$nombre,\$cam,\$c_fil,\$f_agre,\$c_agre,\$cond,\$pag,\$orden=array())

Esta función se encarga de calcular las funciones agregadas. Sus argumentos son los siguientes:

- \$nombre: Es el nombre del reporte ROLAP estrella
- \$cam: Arreglo de campos sobre los cuales se agrupa la consulta
- \$f_agre: Arreglo de funciones asociadas que quieren calcularse.
- \$c_agre: Arreglo de campos sobre los cuales se quieren calcular las funciones
- \$c_fil: Arreglo de campos sobre los cuales se aplican los filtros de la consulta.
- \$cond: Arreglo de las condiciones de filtrado.
- \$pag: El número de la página que está visualizando el usuario.
- \$orden: Arreglo que contiene los campos y la forma de la ordenación de los resultados.

Su resultado es un arreglo asociativo, para el cálculo del promedio de edad de los profesores el arreglo es el siguiente:

```
Array
(
  [0] => Array
    (
      [AVG_TES_EDAD_PROFESOR_TESIS] => 37.3478
      [cantidad] => 46
    )
  [1] => Array
    (
      [AVG_TES_EDAD_PROFESOR_TESIS] => 39.6618
      [cantidad] => 68
    )
  ....
)
```

6.3.2.2 Array getConsulta(\$nombre, \$cam,\$c_filtro, \$con,\$pag,\$strans, \$agrupar= false, \$orden = array())

Esta es la función que ejecuta las consultas que retornan los resultados que ve el usuario del reporte ROLAP estrella en el módulo de consultas. Esta función realiza tres tipos de consultas: Las agrupadas, las transpuestas y las de solo consulta.

Sus argumentos son los siguientes:

- \$nombre: Nombre del reporte ROLAP estrella
- \$campos: Campos sobre los cuales se esta agrupando y que se están seleccionando
- \$c_filtro: Campos sobre los cuales se realiza un filtro
- \$con: Condiciones de filtrado sobre los campos anteriores
- \$pag: El número de página que está visualizando el usuario
- \$trans: Contiene el nombre de la columna que se está transponiendo.
- \$agrupar: Determina si se está o no agrupando
- \$orden: Contiene los campos y la forma de ordenar cada uno de ellos.

Para las consultas con la opción de transponer una columna la función retorna un arreglo asociativo como el siguiente:

```
Array
(
  [datos] => Array
    (
      [0 y 1] => Array
        (
          [TTIE_DIA_TESIS] => 1
          [cantidad] => 758
          [mes del año(1)] => 9
          [mes del año(2)] => 5
        )
      [0 y 2] => Array
        (
          [TTIE_DIA_TESIS] => 2
          [cantidad] => 1048
          [mes del año(1)] => 97
          [mes del año(2)] => 2
        )
    )
  [columnas] => Array
    (
      [mes del año(1)] => mes del año(1)
      [mes del año(2)] => mes del año(2)
    )
  [totales_trans] => Array
    (
      [mes del año(1)] => 5098
      [mes del año(2)] => 13
    )
  [totales_columna_trans] => Array
    (
      [mes del año(1)] => 2183
      [mes del año(2)] => 10
    )
)
```

Los campos de la función son los siguientes:

- datos: Arreglo de datos, la clave del arreglo será la combinación de los valores de los campos agrupados, por ejemplo la clave [0 y 1]
- columnas: Los campos transpuestos y sus valores entre paréntesis

- `totales_trans`: Los totales al suma una columnas transpuesta
- `totales_columna_trans`: Los totales para una columna transpuesta considerando todas las tuplas de la tabla del reporte ROLAP estrella que pertenecen a la columna transpuesta.

Para las consultas no agrupadas la función retorna un arreglo asociativo como el siguiente:

```
Array
(
  [0] => Array
    (
      [TDOC_RUT_PROFESOR_TESIS] => 9978
      [TDEP_DEPARTAMENTO_TESIS] =>
    )
  [1] => Array
    (
      [TDOC_RUT_PROFESOR_TESIS] => 34
      [TDEP_DEPARTAMENTO_TESIS] =>
    )...
)
```

Para la salida anterior, cada arreglo corresponde a una fila, y cuenta con los campos del rut del profesor y el nombre del departamento al cual pertenece. Por ejemplo, el rut 9978 pertenece a un departamento que no se conoce y corresponde a una fila de la tabla de resultados.

Para las consultas agrupadas la función retorna un arreglo asociativo como el siguiente:

```
Array
(
  [datos] => Array
    (
      [0] => Array
        (
          [TALU_DV_TESIS] => 0
          [TTIE_DIA_TESIS] => 1
          [cantidad] => 46
        )
      [1] => Array
        (
          [TALU_DV_TESIS] => 0
          [TTIE_DIA_TESIS] => 2
          [cantidad] => 68
        )
      ..
    )
  [total] => 114
)
```

En la salida anterior se puede ver el número total de resultados de la consulta, que corresponde a 114, este valor se calcula como la suma de las cantidades para cada una de las filas que constituyen la tabla de resultados.

6.3.2.3 `Int getCount($nombre,$campos,$campos_filtro, $condicion,$agru,$stranspuesto='')`

Determina el número total de tuplas que satisfacen una consulta del usuario. Sus parámetros son los siguientes:

- \$nombre: Nombre del reporte ROLAP estrella.
- \$campos: Campos de agrupación
- \$campos_filtro: Corresponde a los campos sobre los cuales se realiza un filtrado
- \$condición: Corresponde a las condiciones de filtrado para los \$campos_filtro
- \$agru: Determina si se debe agrupar por \$campos
- \$transpuesto: Distinto de vacío indica que se requiere calcular el número total de resultados para el caso transpuesto.

6.3.2.4 Int getTotal(\$nombre, \$campos,\$campos_filtro, \$agrupar, \$condicion)

Similar a la anterior pero se encarga de calcular la suma del número de tuplas de todos los grupos, en SQL es similar a realizar un SUM(COUNT(*)) por grupo de tuplas.

Sus parámetros son los siguientes:

- \$nombre: Nombre del reporte ROLAP estrella.
- \$campos: Corresponde a los campos de agrupación.
- \$campos_filtro: Corresponde a los campos sobre los cuales se realizan filtros.
- \$agrupar: Determina si se agrupan o no los campos.
- \$condicion: Corresponde a las condiciones de los filtros.

6.3.2.5 Array getSelect(\$nombre, \$columna)

Se encarga de seleccionar todos los valores distintos posibles de una columna. Sus parámetros son los siguientes:

- \$nombre: El nombre del reporte ROLAP estrella
- \$columna: Columna sobre la cual se quiere obtener la lista de valores

El arreglo asociativo retornado por esta función es de la siguiente forma:

```

Array
(
    [1-9] => 1-9
    [10-8] => 10-8
    [1000-6] => 1000-6
    [10000-5] => 10000-5
    [10000000-8] => 10000000-8
    [1000100-5] => 1000100-5
    [10010001-0] => 10010001-0
    [10101000-7] => 10101000-7
    [10203551-8] => 10203551-8
    [11111111-1] => 11111111-1
    [11111111-8] => 11111111-8
    [11628196-1] => 11628196-1
    [117181985-3] => 117181985-3
    [11737231-6] => 11737231-6
    [117956974-0] => 117956974-0
    [118303702-8] => 118303702-8
    .....
)
(El arreglo se encuentra ordenado.)

```

6.3.2.6 Array getRelacionesReporte(\$nombre, \$campos,\$campos_filtro, \$condicion)

Esta función obtiene las tablas y relaciones de unión entre las diferentes tablas involucradas en una consulta del reporte ROLAP estrella. Sus parámetros son los siguientes:

- \$nombre: Nombre del reporte ROLAP estrella
- \$campos: Campos que se quieren recuperar
- \$campos_filtro: Los campos sobre los cuales se aplican las condiciones de filtro
- \$condicion: Condición de filtrado sobre los campos anteriores

El arreglo asociativo de retorno de la función tiene la siguiente forma:

```
Array
(
    [tablas] => Array
        (
            [0] => estadisticas__difusion
            [1] => DIFUSION_PERSONA
            [25] => DIFUSION_CIUADAD
            [29] => DIFUSION_TIEMPO
            [31] => DIFUSION_ACTIVIDAD
        )
    [condicion_join] => Array
        (
            [0] => 1
            [1] => DIFUSION_PERSONA.DPER_ID=estadisticas__difusion.DPER_ID
            [13] => DIFUSION_CIUADAD.DCIU_ID=estadisticas__difusion.DCIU_ID
            [15] => DIFUSION_TIEMPO.DTIE_ID=estadisticas__difusion.DTIE_ID
            [16] => DIFUSION_ACTIVIDAD.DACT_ID=estadisticas__difusion.DACT_ID
        )
)
```

6.3.2.7 Array getDimensiones(\$nombre,\$visibles=COLUMNA_TODAS,\$campo='')

Entrega ciertas columnas de las dimensiones de un reporte. Sus parámetros son los siguientes:

- \$nombre: Nombre del reporte ROLAP estrella
- \$visibilidad: Define la visibilidad de la columna recuperada. Esta puede tener los siguientes valores (definidos en el archivo *config.php*):
 - COLUMNA_INVISIBLES
 - COLUMNA_VISIBLE
 - COLUMNA_TODAS
- \$campo: Define el nombre de la columna que se quiere seleccionar. Si es vacío retorna todos los campos que cumplan la visibilidad.

Retorna el siguiente arreglo asociativo:

```
Array
(
    [0] => Array
```

```
(
  [TAB_ID] => 199
  [TABLA_NOMBRE]=> DIFUSION_PERSONA
)
[1] => Array
(
  [TAB_ID] => 199
  [TABLA_NOMBRE]=> DIFUSION_PERSONA
)
...
)
```

6.3.2.8 Array function getColumnas(\$nombre,\$visible=COLUMNA_VISIBLE)

Entrega la información sobre las columnas del reporte ROLAP estrella. Sus argumentos son:

- \$nombre: Define el nombre del reporte ROLAP estrella
- \$visible: Define la visibilidad de las columnas rescatadas. Los posibles valores son los siguientes (definidos en el archivo *config.php*):
 - COLUMNA_INVISIBLES
 - COLUMNA_VISIBLE
 - COLUMNA_TODA

Retorna un arreglo asociativo de la siguiente forma:

```
Array
(
  [DPER_RUT_DIFUSION] => Array
    (
      [name] => DPER_RUT_DIFUSION
      [title] => RUT
      [type] => string
      [show] => 0
      [checked] => 1
      [group] => 1
      [select] => 1
      [desc] => RUT
      [tabla] => DIFUSION_PERSONA
      [prefijo_tabla] => DPER
    )
  [DPER_NOMBRE1_DIFUSION] => Array
    (
      [name] => DPER_NOMBRE1_DIFUSION
      [title] => NOMBRE1
      [type] => string
      [show] => 1
      [checked] => 1
      [group] => 1
      [select] => 1
      [desc] => NOMBRE1
      [tabla] => DIFUSION_PERSONA
      [prefijo_tabla] => DPER
    )
  ).....)
```

Cada campo tiene como clave su propio nombre en la tabla a la que pertenece. Los datos prefijo_tabla son los primeros caracteres de las tablas, de esta manera se cumple con el

estándar del ADI para declarar columnas de tablas, el resto de los datos fueron explicados en la sección 5.1.2 Base de datos de información local (BD Local)

6.3.3 Funciones de estadísticas almacenadas

Estas funciones son utilizadas para almacenar las consultas estadísticas ejecutadas por los usuarios de la aplicación internamente en el sistema, de manera de poder re-utilizarlas posteriormente.

6.3.3.1 String saveEstadistica(\$n_reporte,\$nombre,\$datos,\$pers_id,\$est_id,\$gru_id)

Retorna un archivo con la extensión “.estadisticas.txt” que contiene la estadística seleccionada para ser descargado por el usuario. Sus parámetros son los siguientes:

- \$n_reporte: El nombre el reporte ROLAP estrella
- \$nombre: El nombre del archivo ingresado por el usuario
- \$datos: Los datos de la estadística
- \$pers_id: El rut de la persona que guarda la estadística
- \$est_id: El identificador de la estadística en el sistema
- \$gru_id: El grupo al cual pertenece la estadística

6.3.3.2 Array getEstadistica(\$est_id, \$pers_id)

Obtiene los datos de una estadística almacenada en el sistema. Sus parámetros son los siguientes:

- \$est_id: El identificador de la estadística en el sistema
- \$pers_id: El rut de la persona que solicita la estadística, debe ser el propietario de ella.

Retorna un arreglo asociativo como el siguiente:

```
Array
(
    [CON_ID] => 3
    [CON_FUNCION] => estadisticas__tesis
    [CON_NOMBRE] => consulta_prueba
    [CON_PERS_ID] => 15819341
    [CON_FECHA] => 2008-11-30 19:36:48
    [CON_DATOS] => array (
        '_modulo' => 'estadisticas2',
        '_script' => 'funcion.php',
        'funcion' => 'estadisticas__tesis',
        'bajar' => '1',
        'debug' => 0,
        'buscador_global' => "",
        'pagina' => '0',
        'filtro' =>
        array (
            'TDOC_RUT_PROFESOR_TESIS' => "",
            .....
        )
    )
(Ha sido cortado debido a su gran tamaño)
```

6.3.3.3 Boolean addEstadistica(\$nombre_reporte, \$nombre, \$datos, \$pers_id)

Almacena en el sistema una estadística. Sus parámetros son los siguientes:

- \$nombre_reporte: Nombre del reporte ROLAP estrella
- \$nombre: El nombre con que el usuario guarda la estadística
- \$datos: Los datos de la estadística
- \$pers_id: El rut de la persona que almacena la estadística

Retorna TRUE en caso de éxito o FALSE en caso de algún error.

6.3.3.4 boolean updEstadistica(\$est_id, \$pers_id, \$datos)

Actualiza una estadística. Sus parámetros son los siguientes:

- \$est_id: El identificador de la estadística
- \$pers_id: El rut de la persona que actualiza la estadística, debe ser el mismo que la creó
- \$datos: Datos que actualizarán la estadística

Retorna TRUE si se efectuó correctamente la actualización o FALSE si ocurrió algún error

6.3.3.5 boolean delEstadistica(\$est_id, \$pers_id)

Elimina una estadística del sistema. Sus parámetros son:

- \$est_id: El identificador de la estadística
- \$pers_id: El rut de la persona que borra la estadística. Debe ser el mismo rut de la persona que la creó.

Retorna TRUE si se eliminó la estadística o FALSE en caso de problemas

6.3.3.6 Array getSelectEstadisticas(\$nombre, \$pers_id)

Obtiene las estadísticas almacenadas en el sistema por una persona. Sus parámetros son los siguientes:

- \$nombre: El nombre de la función sobre la cual se aplica la estadística
- \$pers_id: El propietario de las estadísticas.

Retorna un arreglo asociativo con los nombres de las estadísticas disponibles:

```
Array
(
    [3] => consulta_prueba
    [4] => consulta_prueba_disco
    [8] => generando_graficos
    [6] => ssdd
    [5] => sss
    [7] => sss)
```


6.3.4 Funciones de creación y actualización de reportes ROLAP estrella

Estas funciones son utilizadas para crear el modelo de datos ROLAP estrella en base a las especificaciones del diseñador. También permiten realizar el proceso de actualización de estos modelos, ya sea de manera incremental, en línea o con procesos batch.

6.3.4.1 boolean `isReporte($nombre)`

Comprueba si un reporte existe. Esta función es utilizada para evitar conflictos en los nombres de los reportes. Sus argumentos son:

- `$nombre`: El nombre del reporte ROLAP estrella

Retorna TRUE si el reporte cuenta con una modelo de datos ROLAP estrella o FALSE si no.

6.3.4.2 int `getUpdReporte($nombre)`

Obtiene el tipo de actualización del reporte. Esta función se utiliza para verificar si la información del reporte debe actualizarse en línea. Sus argumentos son:

- `$nombre`: El nombre del reporte ROLAP estrella

Los tipos retornados son los siguientes (definidos en el archivo *config.php*):

- `ACTUALIZACION_CRON = 1`
- `ACTUALIZACION_LINEA = 2`

6.3.4.3 void `crearDimensionReporte($nombre,$campos)`

Explicada en detalle en la sección 5.2.1.6.6 Automatización de la creación del modelo ROLAP estrella de un reporte.

6.3.4.4 void `crearHechoReporte($nombre,$actualizacion,$campos,$dimensiones)`

Explicada en detalle en la sección 5.2.1.6.6 Automatización de la creación del modelo ROLAP estrella de un reporte.

6.3.4.5 void `crearReporte($datos)`

Explicada en detalle en la sección 5.2.1.6.6 Automatización de la creación del modelo ROLAP estrella de un reporte.

6.3.4.6 Array `getMapeoCampo($nombre,$datos_r)`

Determina a qué dimensión del reporte ROLAP estrella pertenece cada columna de los datos que serán insertados en él. Sus argumentos son los siguientes:

- `$nombre`: Nombre del reporte ROLAP estrella

- \$datos_r: Datos a ser insertado en el reporte ROLAP estrella

Retorna un arreglo asociativo de la forma siguiente:

```

Array
(
    [campos_tabla] => Array
        (
            [RUT_DIFUSION] => DIFUSION_PERSONA
        )
    [campos_columna] => Array
        (
            [RUT_DIFUSION] => DPER_RUT_DIFUSION
        )
    [tabla_id] => Array
        (
            [DPER_ID] => DIFUSION_PERSONA
            [EST_DIF_ID] => estadísticas_difusion
        )
)

```

Los campos del arreglo CAMPOS_TABLA relacionan cada campo con la tabla a la cual pertenece. Los campos del arreglo TABLE_ID relacionan los nombres de las claves primarias de las dimensiones con los nombres de las dimensiones. Los campos del arreglo CAMPOS_COLUMNA relacionan el nombre del campo con su nombre en la base de datos.

6.3.4.7 void guardarDatosReporte(\$nombre_reporte,\$datos_r)

Explicada detalladamente en la sección 5.2.1.5 Insertar resultados en las tablas de los reportes en la BDEE

6.3.4.8 mixed validar_creacion_reporte(\$per_id,\$slave,\$accion)

Valida el acceso al módulo de creación de un reporte y la actualización del reporte, también cambia el estado de actualización de un reporte ROLAP estrella. Sus argumentos son los siguientes:

- \$per_id: El rut de la persona que llama a la función.
- \$slave: El rut del propietario o diseñador del reporte.
- \$accion: La operación que se quiere realizar y validar.

Si se intenta acceder al módulo de creación de un reporte retorna TRUE si el usuario está autorizado o FALSE si no. Si se intenta actualizar la información de los reportes entonces terminará con un error si no es un usuario valido o procederá a cambiar el estado del reportes llamando a la funcion busyReporte().

6.3.4.9 boolean busyReporte(\$nombre)

Actualiza el estado del reporte ROLAP estrella si no esta bloqueado. Sus argumentos son:

- \$nombre: Nombre del reporte ROLAP estrella

Retorna TRUE si lo cambió o FALSE si se produce algún error.

6.3.4.10 int estadoLock(\$nombre)

Retorna el estado actual del reporte. Sus argumentos son:

- \$nombre: Nombre del reporte ROLAP estrella

En caso de error retorna el estado bloqueado.

6.3.4.11 bool lockReporte(\$nombre)

Esta función se encarga de bloquear el reporte ROLAP estrella. Sus argumentos son:

- \$nombre: Nombre del reporte ROLAP estrella

Retorna TRUE si el bloqueo se efectuó o FALSE si ocurrió algún problema

6.3.4.12 bool unlockReporte(\$nombre)

Esta función se encarga de desbloquear un reporte ROLAP estrella. Sus argumentos son:

- \$nombre: Nombre del reporte ROLAP estrella

Retorna TRUE si el desbloqueo se efectuó o FALSE si ocurrió algún problema

6.3.4.13 int getOffset(\$nombre)

Retorna el OFFSET del reporte ROLAP estrella. Sus argumentos son:

- \$nombre: Nombre del reporte ROLAP estrella

En caso de error o no encontrarse el OFFSET se retorna 0, al contrario retorna el OFFSET

6.3.4.14 boolean updOffset(\$nombre,\$nuevo_offset)

Esta función actualiza el OFFSET de actualización de la información del reporte ROLAP estrella. Sus argumentos son:

- \$nombre: Nombre del reporte ROLAP estrella

Retorna TRUE si la consulta se ejecutó con éxito o FALSE en caso de error.

6.3.5 Clases

Las clases son archivos PHP que permiten realizar ciertas funcionalidades de manera centralizada y más ordenada que con la utilización de funciones. Las clases que a continuación se describirán corresponden a las que permiten exportar los resultados a archivos WORD, Excel, generar las imágenes de gráficos, entre otras.

6.3.5.1 Clase wordML

Esta clase fue desarrollada completamente desde cero para efectos de este trabajo. Su objetivo era contar con una clase que permitiera generar archivo Word con tablas e imágenes embebidas, para lograrlo, esta clase genera un archivo de acuerdo al estándar OpenXML. Los métodos utilizados en esta clase son los siguientes.

- **Void WordML(\$title='', \$author='', \$company='')**: Constructor que permite definir un título, un autor y una compañía al documento.
- **Void setFont(\$newFont)**: Define el tipo de fuente a utilizar en el documento.
- **Void setMarginTop(\$new)**: Define el margen superior del documento.
- **Void setMarginBottom(\$new)**: Define el margen inferior del documento.
- **Void setMarginLeft(\$new)**: Define el margen izquierdo del documento.
- **Void setMarginRight(\$new)**: Define el margen derecho del documento.
- **Void setPageSizeH(\$new)**: Define el alto de la página.
- **Void setPageSizeW(\$new)**: Define el ancho de la página.
- **String output(\$name='', \$path='')**: Genera el archivo WORD, se le puede asignar un nombre \$name y una ruta \$path donde almacenarlo.
- **Void addTable(\$table_r, \$size=12, \$align='center', \$negrita=false)**: Inserta la tabla \$table_r en el documento con un tamaño de letra \$size y alineación \$align, también es posible definir su texto con negrita.
- **Void addImage(\$path, \$align='center', \$scale=0.8)**: Inserta una imagen ubicada en la ruta \$path en el documento con una cierta escala \$scale y con una cierta alineación \$align
- **Void newText(\$texto, \$size=28, \$align='center', \$negrita=false, \$cursiva=false)**: Inserta un nuevo texto \$texto en el documento con cierta alineación \$align y tamaño de letra \$size. También puede tener el formato de negrita o cursiva.
- **Void newPage(\$align='center')**: Agrega una nueva página al documento, alineando en el centro el cursor.

6.3.5.2 Class excelMHTML

Esta clase fue desarrollada completamente desde cero para efectos de este trabajo. Su objetivo era contar con una clase que permitiera generar archivos Excel con imágenes embebidas, para lograrlo esta clase genera un archivo MHTML. Los métodos utilizados en esta clase son los siguientes.

- **excelMHTML():** Constructor de la clase
- **addTable(\$tabla):** Inserta la tabla referenciada por \$tabla en el documento
- **addImage(\$path):** Agrega una imagen ubicada en la ruta \$path al archivo Excel
- **output(\$nombre='Estadisticas.xls'):** Genera el documento con el nombre por defecto Estadisticas.xls o cualquier otro definido en la variable \$nombre

6.3.5.3 Class Gráfico

Esta clase fue inicialmente desarrollada por el ADI pero fue mejorada para efectos de este trabajo. Esta clase utiliza la librería jgraph disponibles en la carpeta de librerías del ADI LIB_PHP. La función utilizada en este trabajo fue:

- **graficador(\$tipo,\$datos,\$path,\$tit=null,\$label_x='', \$label_y=''):** Sus parámetros son los siguientes:
 - \$tipo: Define el tipo de gráfico, puede ser:
 - barras_horizontales
 - barras_verticales
 - lineas
 - torta
 - \$datos: Contiene los datos a graficar
 - \$path: Define la ruta donde se guardará la imagen
 - \$tit: El título del gráfico
 - \$label_x: El nombre del eje x
 - \$label_y: El nombre del eje y

Esta función generará una imagen JPEG que será almacenada en la ruta definida en el parámetro \$path, de esta manera cualquier aplicación que utilice esta clase podrá insertar la imagen en el archivo exportado.

6.3.5.4 Class Excel

Se trata de una clase también disponible en el directorio LIB_PHP. Esta librería fue desarrollada completamente por el ADI y solo permite insertar tablas en el archivo Excel generado, no imágenes. Para utilizarla se realiza una llama estática a la siguiente función:

- **Excel:: streamTable(\$tabla,\$nombre, 'inline') :** Sus parámetros son los siguientes:
 - \$tabla: Es un arreglo con la tabla que se quiere insertar en el archivo.
 - \$nombre: Es el nombre del archivo que se podrá descargar una vez generado.
 - 'inline': Define que el archivo pueda ser descargado por el usuario mediante la opción de un cuadro emergente en su navegador.

6.3.5.5 Class RTF

Esta clase fue desarrollada por el ADI y se encuentra en la carpeta de librerías LIB_PHP. Esta clase permite generar archivo RTF con imágenes y tablas, es por esta razón que se

utilizó para generar informes finales. Los métodos importantes para generar estos gráficos y la tabla son los siguientes:

- **RTF()**: Es el constructor de la clase
- **boolean creaRTF()**: Prepara el archivo con los encabezados necesarios. Siempre retorna TRUE
- **boolean insertaTabla(\$datos, \$encabezados="", \$titulo="", \$lineas=0)**: Se encarga de insertar una tabla en el archivo. Sus parámetros son los siguientes:
 - \$datos: La tabla a ser insertada
 - \$encabezados: La fila de la cabecera
 - \$titulo: El título de la tabla
 - \$lineas: El número de líneas a insertar antes de la tabla
- **boolean nuevaPagina(\$cantidad=1)**: Genera 1 o más hojas dependiendo del número pasado como parámetro. Siempre retorna TRUE
- **Boolean insertaTexto(\$texto, \$tamano=12, \$alineacion='', \$sangria=0, \$lineas=0)**: Inserta un texto en el documento. Sus parámetros son los siguientes:
 - \$texto: El texto a insertar en el documento
 - \$tamano: El tamaño de la fuente, por defecto 12
 - \$alineacion: La alineación del texto ('centro','derecha', 'justificado', 'izquierda')
 - \$sangria: El número de veces a aplicar sangría
 - \$lineas: Número de líneas a insertar antes del texto

Retorna TRUE si el texto se inserta exitosamente, sino retorna FALSE

- **Boolean insertaImagen(\$imagen, \$alineacion="", \$lineas=0)**: Inserta una imagen en el archivo. Sus parámetros son los siguientes:
 - \$imagen: La ruta a la imagen
 - \$alineación: La alineación ('centro','derecha', 'justificado', 'izquierda')
 - \$lineas: Número de líneas a insertar antes del texto

Retorna TRUE si la imagen se inserta exitosamente, sino retorna FALSE

- **String enviaContenido()**: Envía al navegador el documento RTF para ser descargado por el usuario.

7 Interfaces y funcionalidades

En este capítulo se describen las diferentes interfaces de la aplicación SIAGER por módulos. Cada uno de estos módulos se dividirá en sus diferentes funcionalidades, detallando en cada una de ellas algunos ejemplos de sus resultados o salidas.

7.1 Módulo de inicio

Aquí solo se desplegarán los reportes con una función asociada definida. Desde este módulo es posible dirigirse directamente al de creación de reportes o al de consulta de reportes, dependiendo si ya fue creado su modelo de datos ROLAP estrella o no.

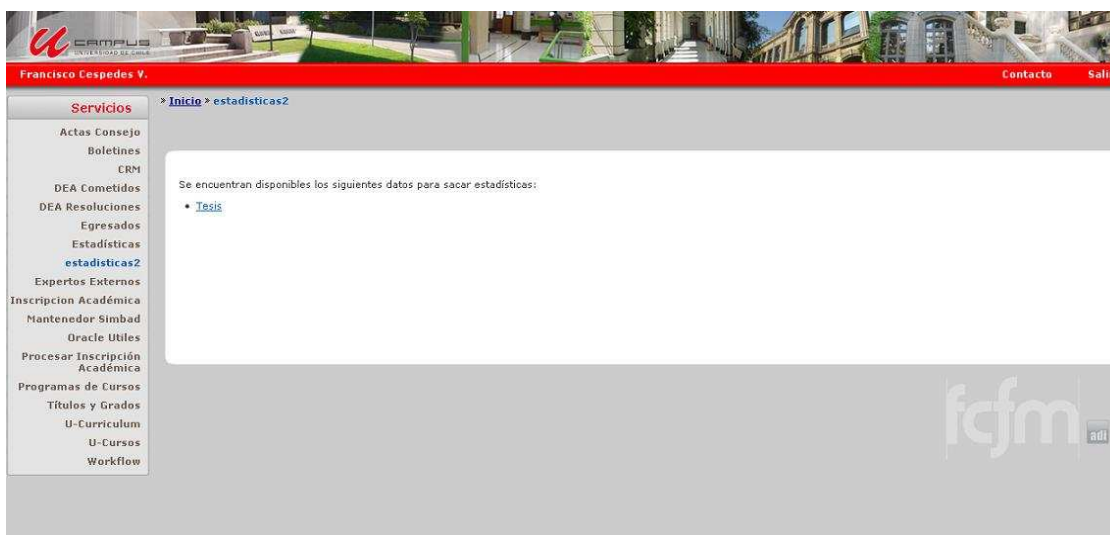


Fig 7.1 Módulo de inicio

7.2 Módulo de Consulta

Este modulo es el que le brinda al usuario la posibilidad de realizar consultas estadísticas sobre la información disponible. Además de realizar las consultas, se pueden visualizar sus resultados en gráficos de diferentes tipos, dependiendo del caso estos podrán ser de barras, líneas o torta.

7.2.1 Actualización de información

Cuando el tipo de actualización de un reporte es en línea se realiza una llamada a la función asociada para rescatar la nueva información y actualizarla. Si la información se actualiza exitosamente entonces el usuario del sistema verá un mensaje como el siguiente:

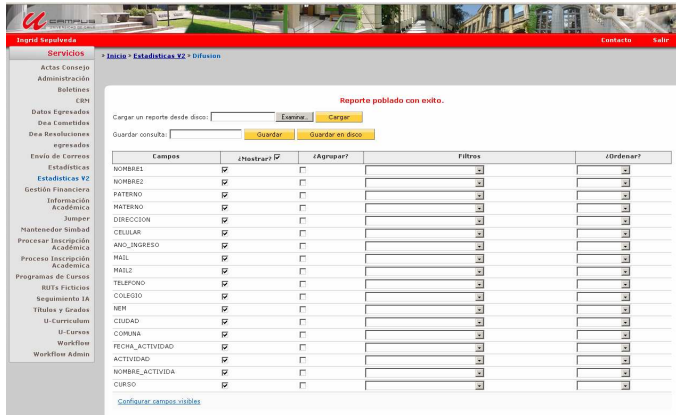


Fig 7.2 Resultado exitoso de la actualización en línea de un reporte

Durante el proceso de actualización de la información cualquier otro usuario que intente realizar una consulta no podrá acceder al módulo, puesto que la información está cambiando. En este caso el usuario debería ver un mensaje como el siguiente:

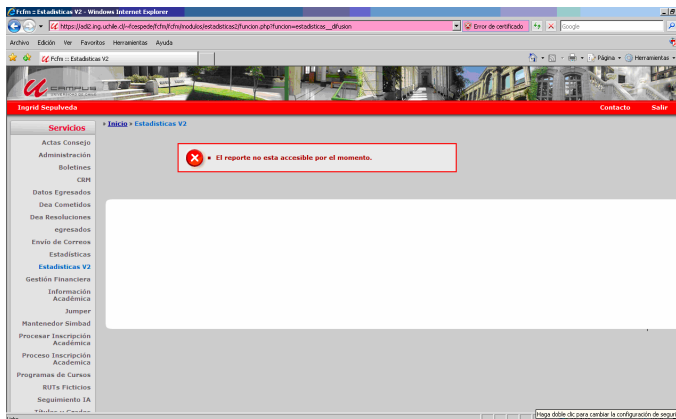


Fig 7.3 Acceso al módulo de consultas durante una actualización en curso

Si la información que debe ser actualizada en la base de datos es vacía, es decir, no hay nueva información para actualizar, el usuario debería recibir un mensaje como el siguiente:

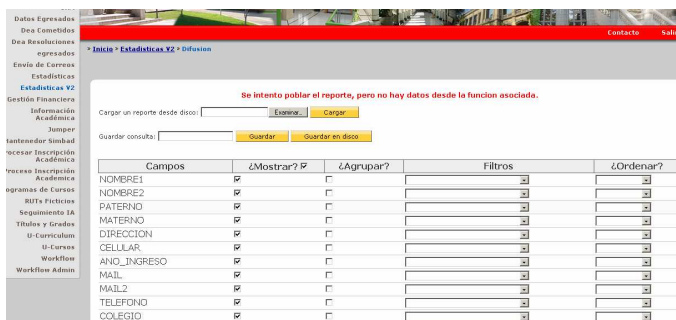


Fig 7.4 Actualización en línea de la información. No hay datos que actualizar

7.2.2 Consultar reporte

Este es el módulo donde el usuario define las consultas de acuerdo a ciertos filtros y condiciones. Es posible guardar consultas en disco o en el sistema, así como también cargar alguna consulta almacenada en disco.

Cargar consulta:

Cargar un reporte desde disco:

Guardar consulta:

Campos	¿Mostrar?	¿Agrupar?	Filtros	¿Ordenar?
rut del profesor	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="text"/>	<input type="text"/>
nombre departamento	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="text"/>	<input type="text"/>
nombre carrera	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="text"/>	<input type="text"/>
nombre alumno	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="text"/>	<input type="text"/>
digito verificar alumno	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="text"/>	<input type="text"/>
rut del alumno	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="text"/>	<input type="text"/>
dia del mes	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="text"/>	<input type="text"/>
mes del año	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="text"/>	<input type="text"/>
año	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="text"/>	<input type="text"/>
cargo en la comision	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="text"/>	<input type="text"/>
apellidos profesor	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="text"/>	<input type="text"/>
nombres profesor	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="text"/>	<input type="text"/>
digito verificar profesor	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="text"/>	<input type="text"/>
la jerarquia del profesor	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="text"/>	<input type="text"/>
el tema de la memoria	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="text"/>	<input type="text"/>
edad profesor	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="text"/>	<input type="text"/>

[Configurar campos visibles](#)

Fig 7.5 Opciones de guardar, cargar consulta y campos de la consulta

También se puede ver la lista de campos disponibles para efectuar filtros, agrupaciones o simplemente mostrar los valores de sus campos. Para cada campo es posible definir una forma de ordenación.

Actas Consejo
Administración
Boletines
CEN
Datos Egresados
Dea Comités
Dea Resoluciones
egresados
Envío de Correos
Estadísticas
Estadísticas V2
Gestión Financiera
Información Académica
Jumper
Mantenedor Simbad
Procesar Inscripción Académica
Proceso Inscripción Académica
Programas de Cursos
RUTs Ficticios
Seguimiento IA
Títulos y Grados
U-Curriculum
U-Cursos
Workflow
Workflow Admin

Se intento poblar el reporte, pero no hay datos desde la funcion asociado.

Cargar consulta:

Cargar un reporte desde disco:

Campos	¿Mostrar?	¿Agrupar?	Filtros	¿Ordenar?
el apellido materno	<input type="checkbox"/>	<input type="checkbox"/>	<input type="text"/>	<input type="text"/>
el apellido paterno	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/> Usar como títulos	<input type="text"/>	Ascendente
el segundo nombre	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/> Usar como títulos	<input type="text"/>	Ascendente
el primer nombre	<input type="checkbox"/>	<input type="checkbox"/>	<input type="text"/>	<input type="text"/>
el rut	<input type="checkbox"/>	<input type="checkbox"/>	<input type="text"/>	<input type="text"/>

Otros

Ningun título
 Totales
 Porcentajes %
 Sólo Contar

Aplicar: Sobre el campo:

Aplicar: Sobre el campo:

Aplicar: Sobre el campo:

Ordenar: la cantidad

[Configurar campos visibles](#)

Visualizando 15 de 8130 datos encontrados.

N°	el apellido paterno	el segundo nombre	Cantidad
1	NO DEFINIDO	NO DEFINIDO	3
2	NO DEFINIDO	ALBORNOZ	1
3	NO DEFINIDO	Alejandro	1
4	NO DEFINIDO	ANDRES	1
5	NO DEFINIDO	BUSCAGLIONE	1
6	NO DEFINIDO	CARRASCO	1
7	NO DEFINIDO	CAVIERES	1
8	NO DEFINIDO	COPIENTES	1
9	NO DEFINIDO	COHEN	1
10	NO DEFINIDO	DIEZ	1

Fig 7.6 Selección de la opción de ordenación por campos agrupados

Algunos campos pueden ser no agrupables, y otros no contar con la posibilidad de seleccionar entre sus valores, dado la gran cantidad de ellos. Por ejemplo, los campos de nombres o apellidos de una persona seleccionables podrían verse como se muestra en la siguiente imagen:

Servicios **Inicia** > Estadísticas V2 > nuevo

Actas Consejo
Administración
Boletines
CEN
Datos Egresados
Dea Comités
Dea Resoluciones
egresados
Envío de Correos
Estadísticas
Estadísticas V2
Gestión Financiera
Información Académica
Jumper
Mantenedor Simbad
Procesar Inscripción Académica
Proceso Inscripción Académica
Programas de Cursos
RUTs Ficticios
Seguimiento IA
Títulos y Grados
U-Curriculum
U-Cursos
Workflow
Workflow Admin

Se intento poblar el reporte, pero no hay datos desde la funcion asociado.

Cargar consulta:

Cargar un reporte desde disco:

Campos	¿Mostrar?	¿Agrupar?	Filtros	¿Ordenar?
el apellido materno	<input type="checkbox"/>	<input type="checkbox"/>	Seleccionar Uno	<input type="text"/>
el apellido paterno	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/> Usar como títulos	<input type="text"/>	Ascendente
el segundo nombre	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/> Usar como títulos	<input type="text"/>	Ascendente
el primer nombre	<input type="checkbox"/>	<input type="checkbox"/>	<input type="text"/>	<input type="text"/>
el rut	<input type="checkbox"/>	<input type="checkbox"/>	<input type="text"/>	<input type="text"/>

Otros

Ningun título
 Totales
 Porcentajes %
 Sólo Contar

Aplicar: Sobre el campo:

Aplicar: Sobre el campo:

Aplicar: Sobre el campo:

Ordenar: la cantidad

[Configurar campos visibles](#)

Visualizando 15 de 8130 datos encontrados.

N°	el apellido paterno	el segundo nombre	Cantidad
1	NO DEFINIDO	NO DEFINIDO	3
2	NO DEFINIDO	ALBORNOZ	1
3	NO DEFINIDO	Alejandro	1
4	NO DEFINIDO	ANDRES	1
5	NO DEFINIDO	BUSCAGLIONE	1
6	NO DEFINIDO	CARRASCO	1
7	NO DEFINIDO	CAVIERES	1
8	NO DEFINIDO	COPIENTES	1
9	NO DEFINIDO	COHEN	1
10	NO DEFINIDO	DIEZ	1

Fig 7.7 El campo apellido materno no es agrupable, pero puede filtrarse un valor específico

Al agrupar los campos se pueden calcular porcentajes, totales o contar el número de resultados por agrupación. También es posible definir funciones agregadas como el promedio, la desviación estándar o la varianza sobre los campos numéricos.

Otros

Totales
 Porcentajes %
 Sólo Contar

Aplicar: Promedio Sobre el campo: edad profesor
 Aplicar: Seleccione uno Sobre el campo: Seleccione uno
 Aplicar: Seleccione uno Sobre el campo: Seleccione uno
 Ordenar: la cantidad

[Configurar campos visibles](#)

Fig 7.8 Opciones de agrupación

La opción de transponer los datos ordena como columnas los diferentes valores que puede tomar el campo transpuesto. Esta opción resulta importante para realizar estudios de Cohorte o visualizar el comportamiento a lo largo del tiempo. El resultado de transponer un campo se puede ver en la siguiente imagen:

Nº	el primer nombre	el segundo nombre	el segundo nombre	el segundo nombre	el segundo nombre	el segundo nombre	el segundo nombre	el segundo nombre
1	NO DEFINIDO	VALERIA	0	0	0	0	0	0
2	NO DEFINIDO	SEBASTIAN	0	0	0	0	0	0
3	NO DEFINIDO	PAUL	0	0	0	1	0	0
4	NO DEFINIDO	PAULO	0	1	0	0	0	0
5	NO DEFINIDO	PAULETTE	0	0	0	0	0	0
6	NO DEFINIDO	NICOLE	0	0	0	0	0	0
7	NO DEFINIDO	NICOLAS	0	0	0	0	0	0
8	NO DEFINIDO	MAXIMILIANO 1	0	0	0	0	0	0
9	NO DEFINIDO	MARTIN	0	0	1	0	0	0
10	NO DEFINIDO	MARION	0	0	0	0	0	1
11	NO DEFINIDO	JORGE	0	0	0	0	0	1
12	NO DEFINIDO	SOEL	0	0	0	0	0	0
13	NO DEFINIDO	FRANCISCO	0	0	0	0	0	1
14	NO DEFINIDO	DANISA	0	0	0	0	0	0
15	NO DEFINIDO	CONSTANZA	0	0	0	0	0	0

Distribuido por (el segundo nombre y el primer nombre)
 --(NO DEFINIDO y VALERIA) --(NO DEFINIDO y SEBASTIAN)
 --(NO DEFINIDO y PAUL) --(NO DEFINIDO y PAULO)
 --(NO DEFINIDO y PAULETTE) --(NO DEFINIDO y NICOLE)
 --(NO DEFINIDO y NICOLAS) --(NO DEFINIDO y MAXIMILIANO)
 --(NO DEFINIDO y MARTIN) --(NO DEFINIDO y MARION)
 --(NO DEFINIDO y JORGE) --(NO DEFINIDO y SOEL)
 --(NO DEFINIDO y FRANCISCO) --(NO DEFINIDO y DANISA)

Fig 7.9 Opción de transponer los datos. Se ha transpuesto el campo “el segundo nombre”

Al pie de página es posible ver las diferentes páginas para navegar por los resultados, también es posible exportar los reportes en archivo Excel, PDF, WORD o RTF. Se muestra además el número de resultados de la página que se está visualizando y la cantidad total de resultados.

Consultar Exportar Excel Exportar Excel en graficos Exportar PDF Exportar DOC Exportar RTF

Visualizando 15 de 1286 datos encontrados.

1 2 3 4 5 6 7 2441 Signatura

Nº	rol del profesor	nombre departamento	nombre carrera	nombre alumno	digito verificador alumno	rol del alumno	edad del alumno	fecha de ingreso	tipo de curso	apellidos profesor	nombre profesor	digito verificador profesor	la materia del profesor	el tema de la materia	edad profesor	
1	4739978	NO DEFINIDO	INGENIERO DE EJECUCION ELECTRICA	VICTOR CORNEJO ANTONIO	9	4281694	30	6	1945	Profesor Data	VALDENEGRO ESPINOZA	ARIEL ANTONIO	2	NO DEFINIDO	"DEFERACION CONFIGURACIONES PAREL-ACEITE (PARTE EXPERIMENTAL"	64
2	5089687	NO DEFINIDO	INGENIERO DE EJECUCION ELECTRICA	VICTOR CORNEJO ANTONIO	9	4281694	30	6	1945	Profesor Integrante Comision	AVCASQUER HUET	AUGER EMILE	0	NO DEFINIDO	"DEFERACION CONFIGURACIONES PAREL-ACEITE (PARTE EXPERIMENTAL"	62
3	2038158	NO DEFINIDO	GEOLOGO	VERCEJO ROJAL CECILIA	0	500003111	27	12	1960	Profesor Integrante Comision	FUENZALIDA PONCE	HUMBERTO 2 ADRIAN	0	NO DEFINIDO	"ESTUDIO SEDIMENTOLOGICO DE ARENAS DE ELATA EN LA COSTA DE CHILE CENTRAL ENTRE VALPARAISO Y SANTIAGO"	72
4	500000001	NO DEFINIDO	GEOLOGO	VERCEJO ROJAL CECILIA	0	500003111	27	12	1960	Profesor Data	PROF. NO ACTUALIZADO		0	NO DEFINIDO	"ESTUDIO SEDIMENTOLOGICO DE ARENAS DE ELATA EN LA COSTA DE CHILE CENTRAL ENTRE VALPARAISO Y SANTIAGO"	0

Fig 7.10 Botones de exportar los resultados de la consulta, páginas y tabla de resultado

En el caso que al ejecutar una consulta estadística esta no arroje resultados, el usuario debería ver aparecer en pantalla un mensaje en el cual se le notifique de ello. En la siguiente imagen se puede ver el mensaje cuando no hay resultados para una consulta:

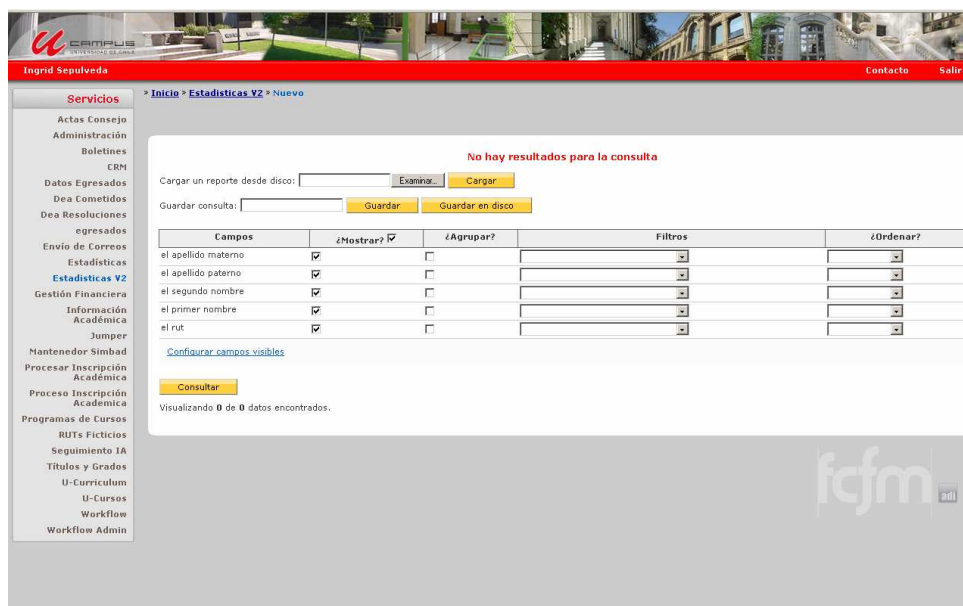


Fig 7.11 La consulta no arrojó resultados

7.2.3 Cargar una consulta desde el sistema

Si se selecciona una consulta almacenada en el sistema, y esta es satisfactoriamente ejecutada, el usuario debería ser notificado con un mensaje, además de desplegar los resultados de la consulta en pantalla. La siguiente imagen permite ver el resultado de cargar una consulta desde el sistema:

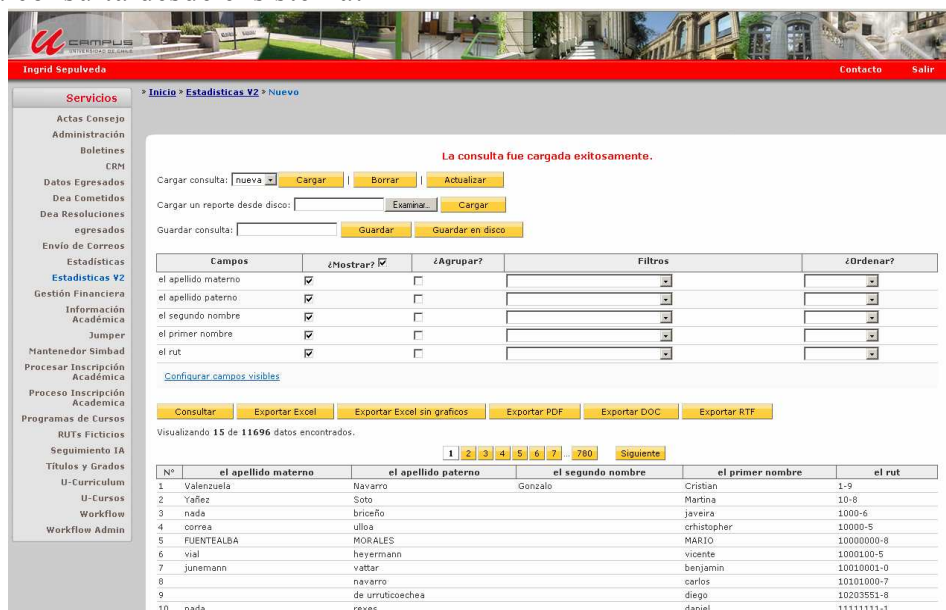


Fig 7.12 Cargar una consulta desde el sistema

Al momento de cargar una consulta desde el sistema pueden ocurrir una serie de problemas, por ejemplo no contar con los permisos para realizar la carga o que la consulta ya no se encuentre en el sistema, en estos casos se mostrará el siguiente mensaje al usuario:



Fig 7.13 Mensaje mostrado al usuario cuando ocurre un problema al cargar una consulta

7.2.4 Cargar una consulta desde un archivo en disco

Para cargar una consulta almacenada en el disco del usuario es necesario presionar el botón “Examinar”, de esta manera el navegador mostrará una ventana para seleccionar el archivo con la configuración de la consulta. Esta acción puede verse en la siguiente imagen:

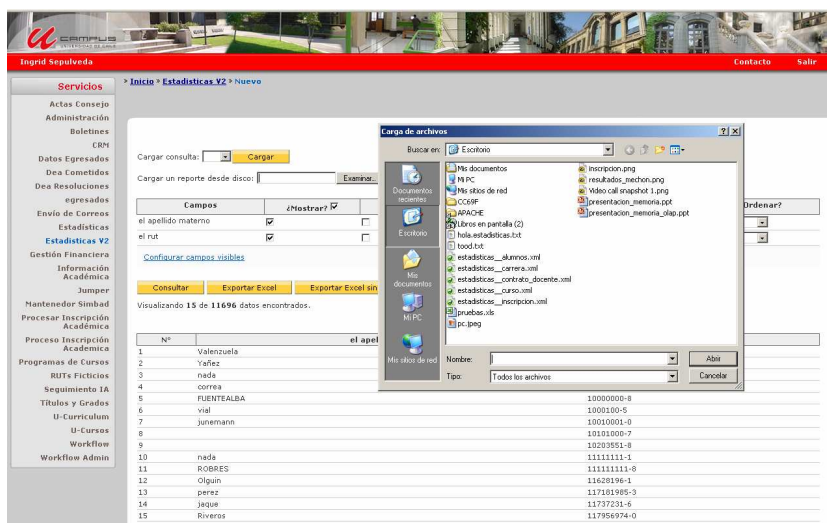


Fig 7.14 Seleccionar un archivo en disco con los parámetros de la consulta

Si la consulta es cargada exitosamente entonces se desplegará un mensaje similar al de la Fig 7.12. Si ocurre algún error y no es posible cargar la consulta desde el archivo en disco entonces se desplegará un mensaje similar al de la Fig 7.13.

7.2.5 Borrar una consulta desde el sistema

Si la consulta es borrada satisfactoriamente se despliega al usuario un mensaje como el de la Fig 7.12 pero sobre la acción de borrar una consulta. Si ocurre un error durante la actualización se despliega un mensaje similar al de la Fig 7.13 pero para la acción de borrar una consulta

Si no se cuenta con los permisos sobre la consulta como para borrarla el usuario debería ver en pantalla un mensaje como el siguiente:

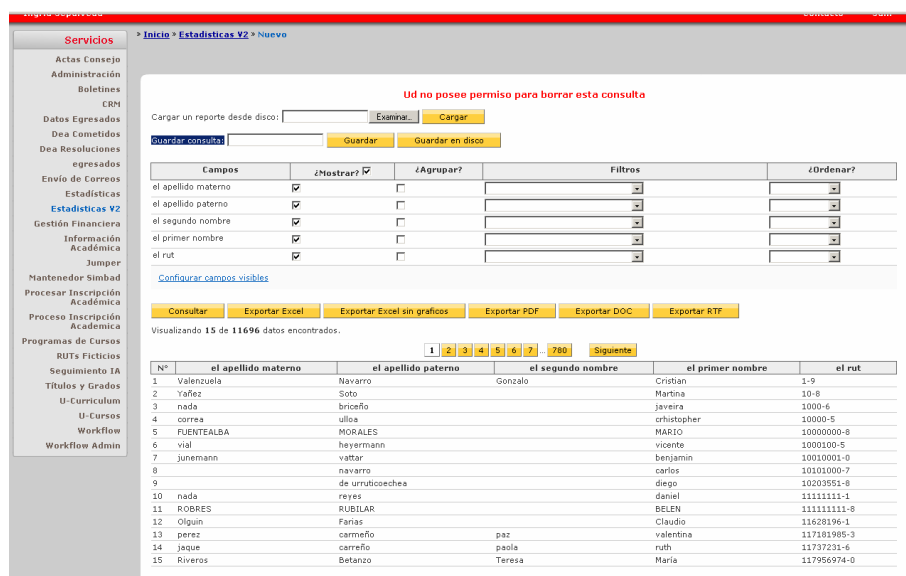


Fig 7.15 Mensaje desplegado al usuario cuando intenta borrar una consulta indebidamente

7.2.6 Actualizar una consulta en el sistema

Por ultimo, además de borrar o cargar una consulta es posible actualizarla. Para poder actualizar una consulta primero es necesario haberla cargada, posteriormente esta consulta podrá ser modificada y ejecutada para ver los resultados con las modificaciones. Si el usuario no cuenta con los permisos para actualizar una consulta entonces se le desplegará un mensaje similar a la Fig 7.15 pero para el caso de una actualización. Si un error se produce cuando se está actualizando una consulta, se desplegará un mensaje como el siguiente:

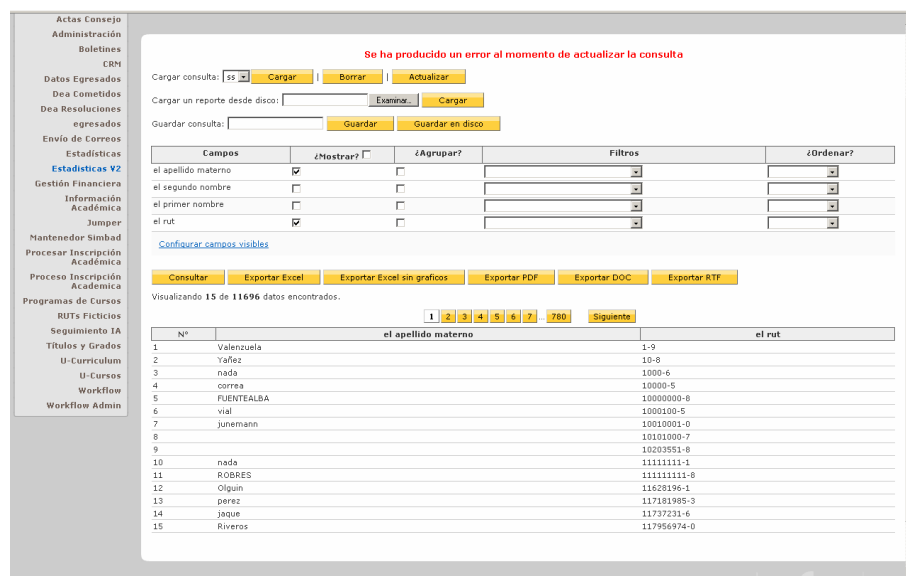


Fig 7.16 Mensaje desplegado al usuario cuando ocurre un error mientras se intentaba actualizar

7.2.7 Guardar una consulta en disco

Al seleccionar una consulta para guardarla en disco el navegador deberá mostrar un cuadro emergente con la opción para descargar un archivo con los parámetros de la consulta, en este cuadro se le ofrece al usuario la opción de guardar o abrir el archivo.

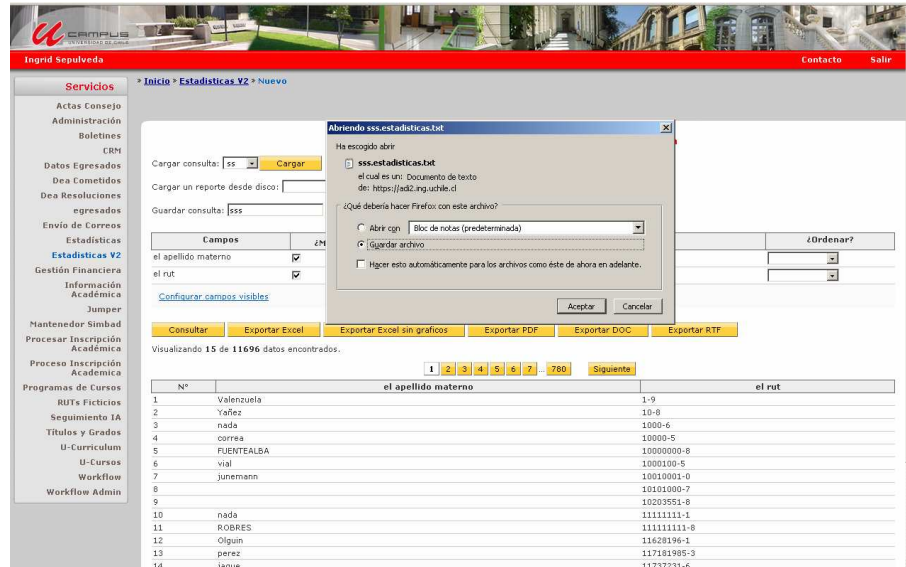


Fig 7.17 Archivo generado con los parámetros de la consulta

Si ocurriera algún error mientras se está generando el archivo se mostraría un mensaje similar al de la Fig 7.16, pero para el caso de guardar una consulta en disco. Si se intenta guardar una consulta sin ingresar un nombre al archivo se desplegará un mensaje como el siguiente:

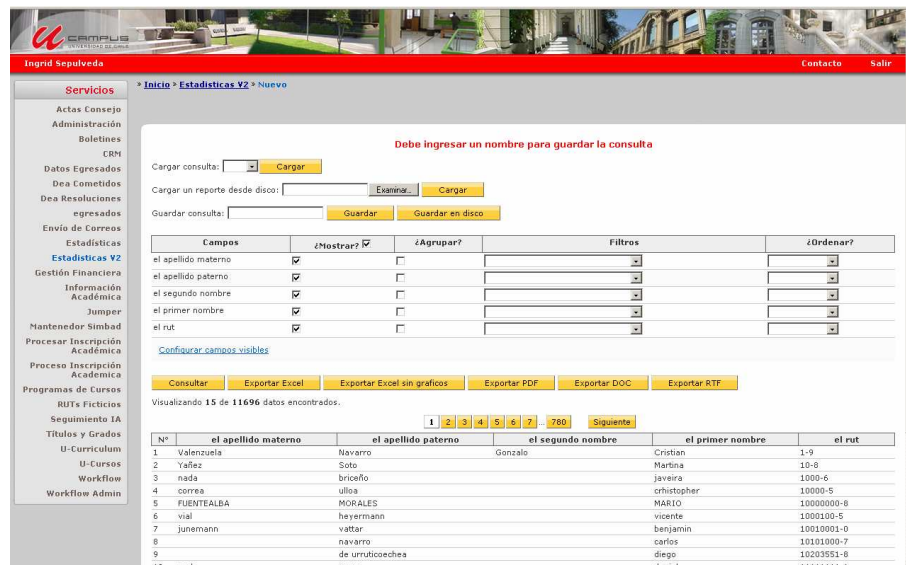


Fig 7.18 Mensaje al intentar guardar un archivo en disco sin un nombre

Si no se cuenta con el permiso para AGREGAR y se intenta almacenar una consulta en el disco, se desplegará un mensaje similar al de la Fig 7.15 pero para el caso de guardar una consulta en disco.

7.2.8 Guardar una consulta en el sistema

La opción para almacenar una consulta en el sistema se despliega en la parte superior de la interfaz, junto con la opción de cargar una consulta desde el disco o desde el mismo sistema. Para almacenar una consulta en el sistema se debe escribir un nombre en el cuadro de texto al lado de “Guardar consulta:” y luego presionar el botón “Guardar”.

Si la consulta es guardada exitosamente se muestra un mensaje similar al de la Fig 7.12, pero para el caso de guardar una consulta en el sistema. Si ocurriera un error mientras se intentaba guardar una consulta se debería desplegar un mensaje como el de la Fig 7.16, pero para el caso de guardar una consulta en el sistema. El nombre ingresado para la consulta debe ser distinto de vacío, de lo contrario se desplegará un mensaje similar al de la Fig 7.18, pero para el caso de guardar una consulta en el sistema. Además es necesario contar con los permisos adecuados, de lo contrario se desplegará un mensaje similar al de la Fig 7.15, pero para el caso de guardar una consulta en el sistema.

7.2.9 Graficar

Los gráficos se generarán cuando se seleccionen campos agrupados. Se generarán diferentes gráficos dependiendo si se transponen los datos o solo se agrupan. Por ejemplo, para la opción de cálculo de porcentajes se generarán gráficos de torta, para el resto de las opciones los gráficos generados pueden ser de barras o líneas.

En la siguiente imagen se puede ver el gráfico de barra generado al calcular el promedio de la edad de los profesores agrupados por el nombre del departamento. Cada una de las barras corresponde a una fila de los resultados:

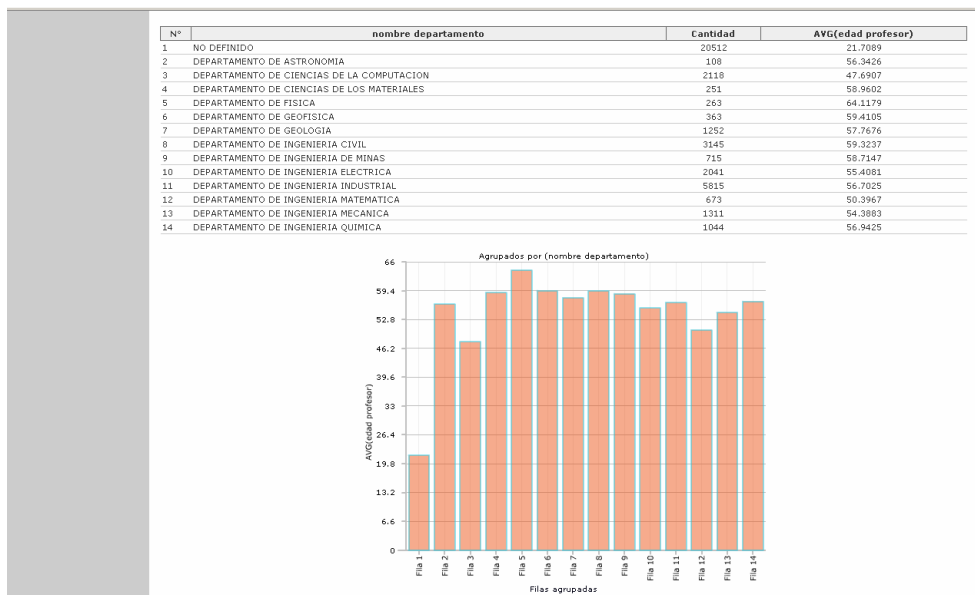


Fig 7.19 Cálculo de promedio de edad de profesores por departamento

Las etiquetas del eje X se han nombrado como Fila 1, Fila 2, etc. para evitar que los nombres largos sean truncados. Además de este gráfico, se genera otro del mismo estilo con el número de profesores por departamento:

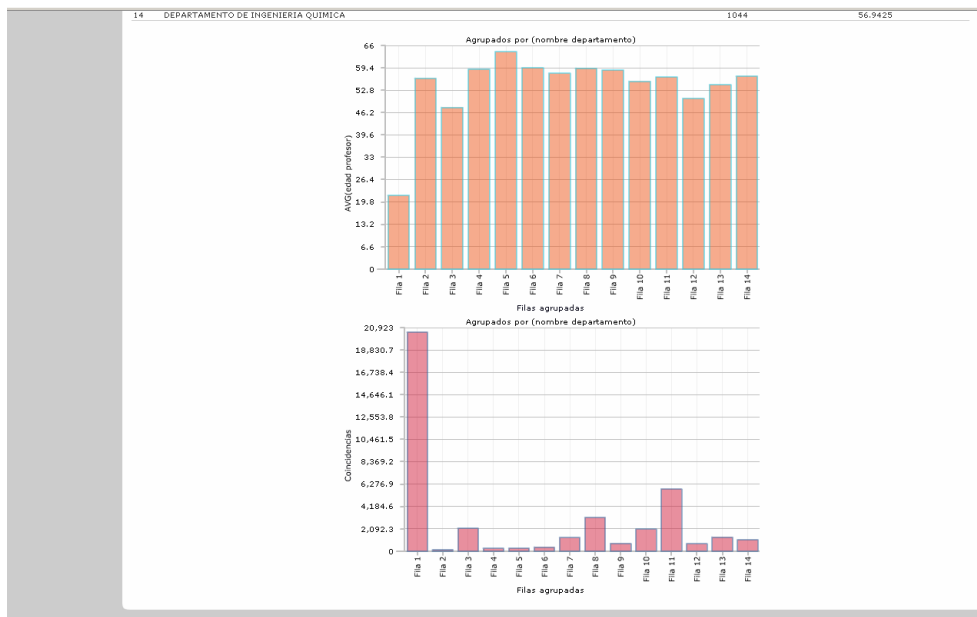


Fig 7.20 Gráfico de barras con el número de profesores por departamento

Si se agrupara por dos o más campos, además de gráficos similares a los anteriores se generaría uno nuevo de líneas, donde cada línea sería la agrupación de las columnas (exceptuando la última, que define las etiquetas del eje X). Por ejemplo, al agrupar por nombre del departamento y nombre de la carrera, los nuevos gráficos serían los siguientes:

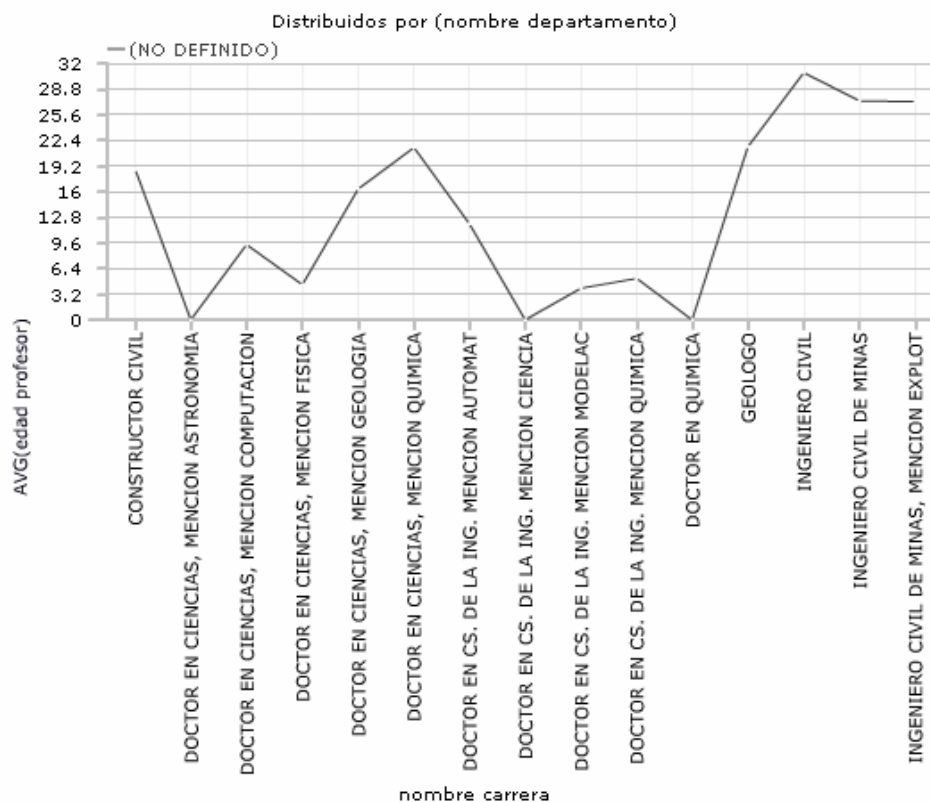


Fig 7.21 Gráfico de líneas de edad promedio de profesores de cada departamento por carrera

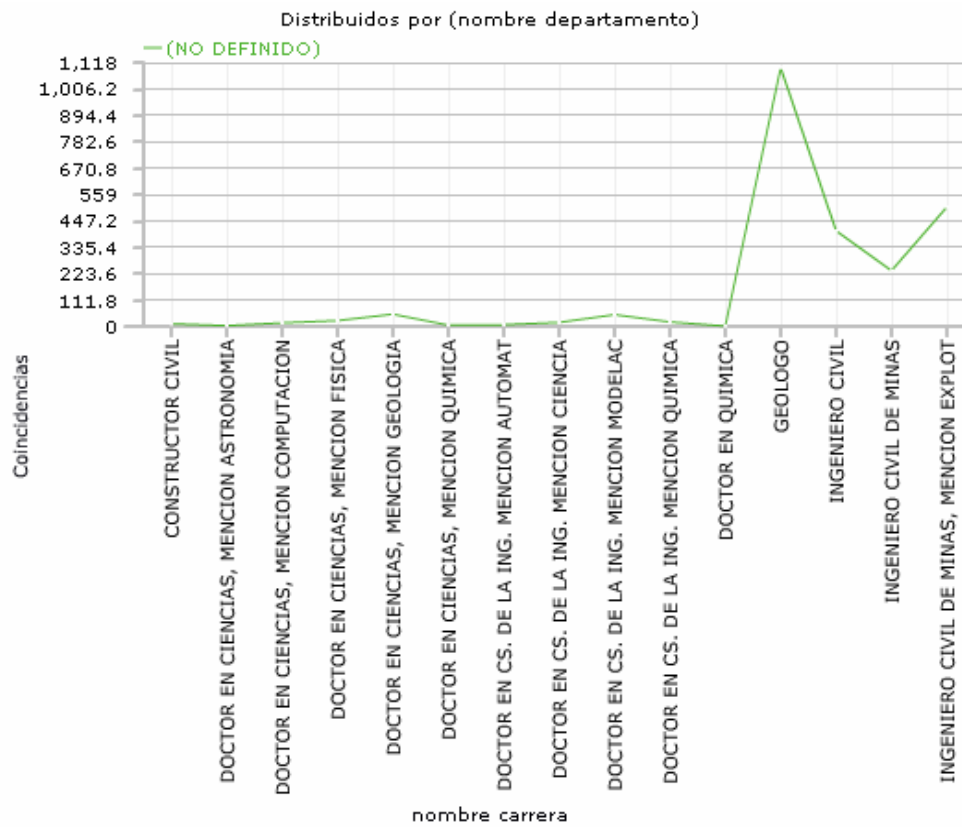


Fig 7.22 Gráfico de líneas para el número de profesores de cada departamento por carrera

Los gráficos de barras generados serían los siguientes:

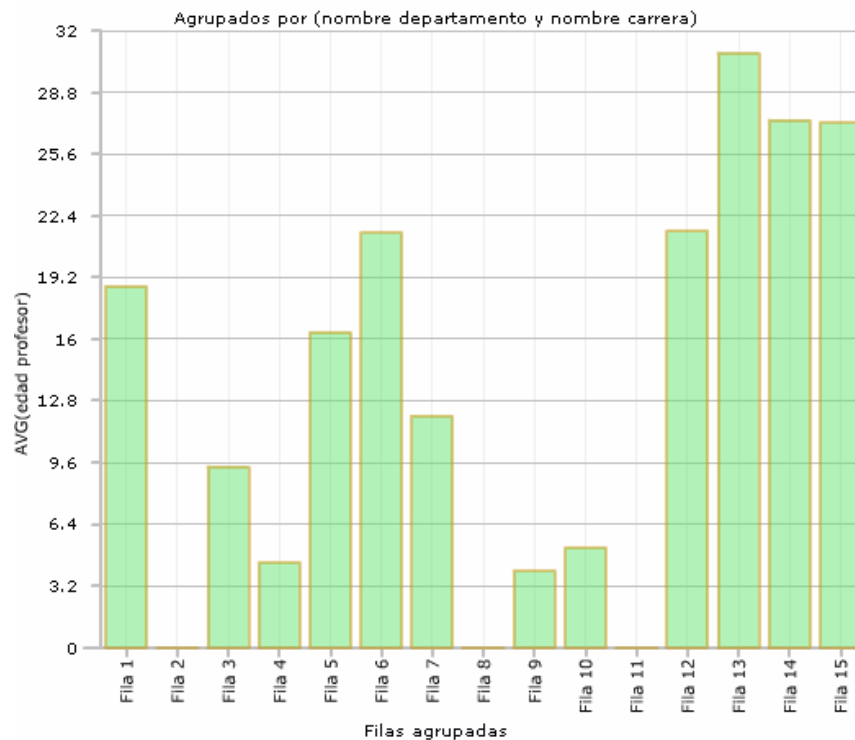


Fig 7.23 Gráfico de barras de edad promedio de profesores agrupados por departamento y carrera

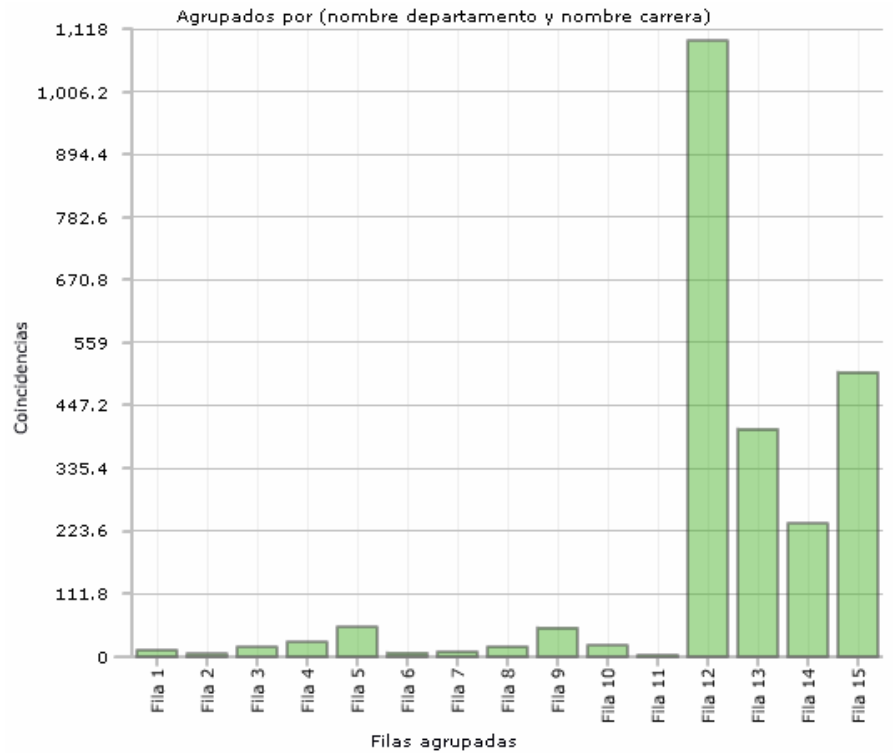


Fig 7.24 Gráfico de barras para el número de profesores agrupados por departamento y carrera

En los gráficos anteriores se han graficado el número de coincidencias para cada una de las filas de los resultados, cada fila es la agrupación de los diferentes valores que pueden tomar las columnas.

Al realizar una agrupación similar a las anteriores pero calculando los porcentajes del número de coincidencias sobre el total de resultados se generará un gráfico de torta como el que puede verse en la siguiente imagen:

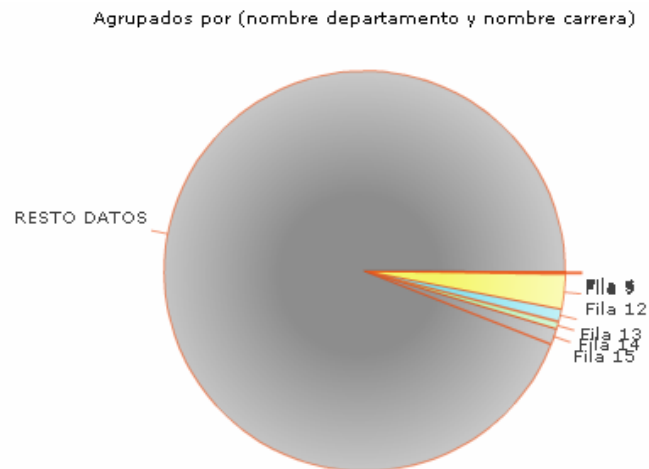


Fig 7.25 Gráfico de torta al agrupar por departamento y carrera

Cada una de las filas de los resultados corresponde a un porcentaje de la torta, cuando la suma total de los porcentajes de las filas no suman cien por ciento se completa con un nuevo

campo 'RESTO DATOS' la diferencia. Las filas con porcentajes cero no son incluidos en ningún gráfico de torta.

Por otra parte, al agrupar por el cargo del profesor en la comisión de una tesis, usando como título (campo transpuesto) el mes del año, se generará un gráfico de barras en la que cada una de las barras corresponderá a una columna transpuesta.



Fig 7.26 Gráficos de barras para el número de profesores según el cargo en la comisión de las tesis distribuidos en los meses del año

En la imagen anterior se puede ver como se ha generado un gráfico de barra por cada combinación de campos agrupados. Las etiquetas del eje X corresponden a las columnas transpuestas. Además se generará un gráfico de barras en el cual se incluyen los datos de los dos gráficos anteriores:

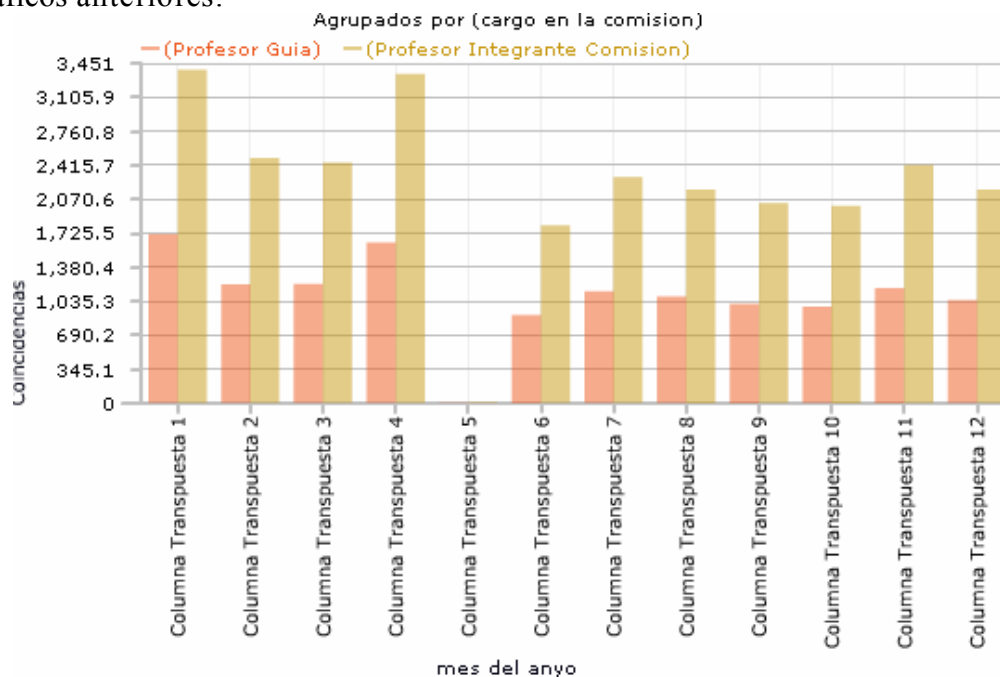


Fig 7.27 Gráfico de barras que grafica los campos agrupados y transpuestos

En el gráfico anterior las etiquetas del eje X corresponde a las columnas transpuestas y cada barra al valor que puede tomar el campo agrupado. También se generará un gráfico de líneas como el que se muestra a continuación:

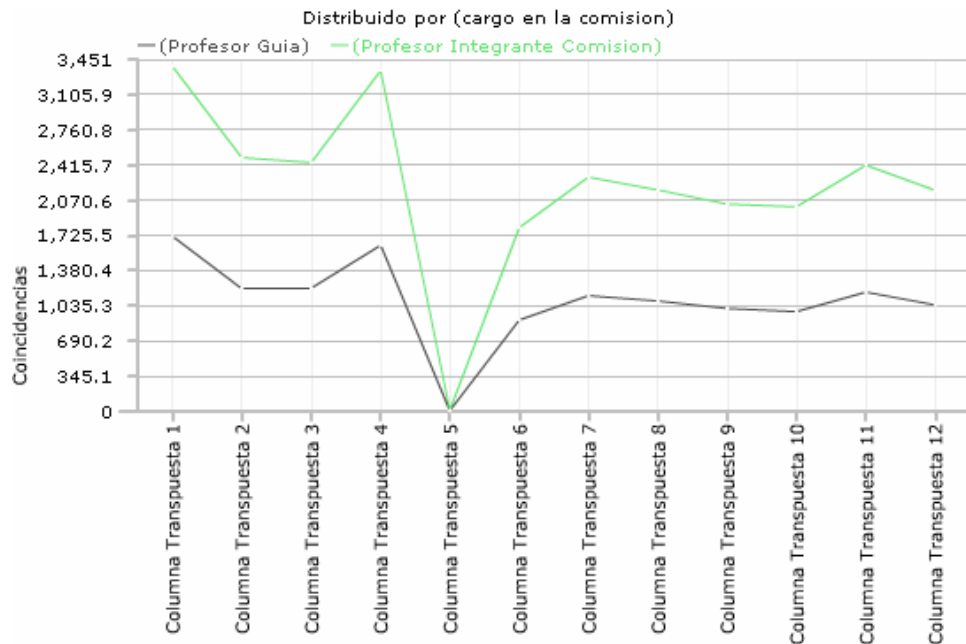


Fig 7.28 Gráfico de líneas que grafica los campos agrupados y transpuestos

En el gráfico anterior, las etiquetas del eje X corresponde a las columnas transpuestas y cada línea corresponde al valor del campo agrupado. Si se calculan porcentajes solo se generarán gráficos de tortas como los siguientes:

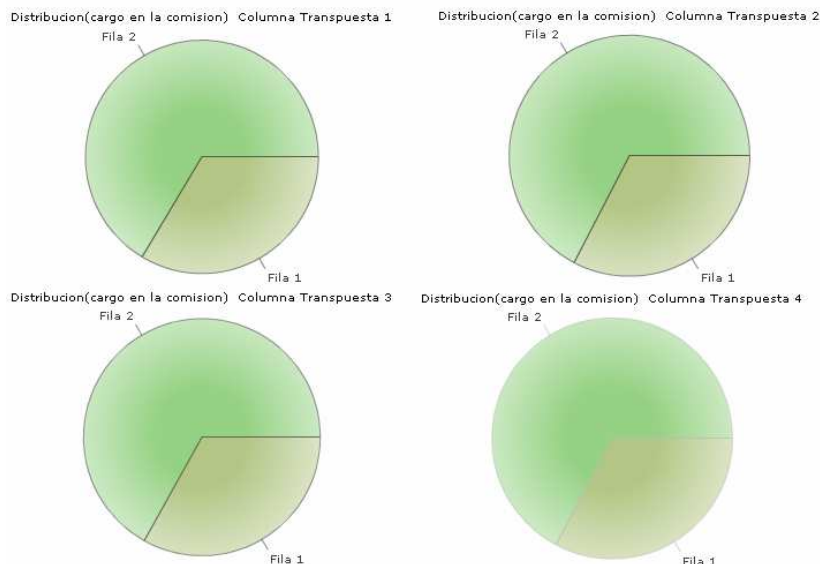


Fig 7.29 Gráficos de torta para las cuatro primeras columnas transpuestas

7.2.10 Exportar informes finales

Al exportar una consulta a un archivo Excel este contendrá tanto la tabla de resultados como los gráficos. En la imagen siguiente se puede ver un gráfico de barras para el número de profesores y para el promedio de edad de ellos:

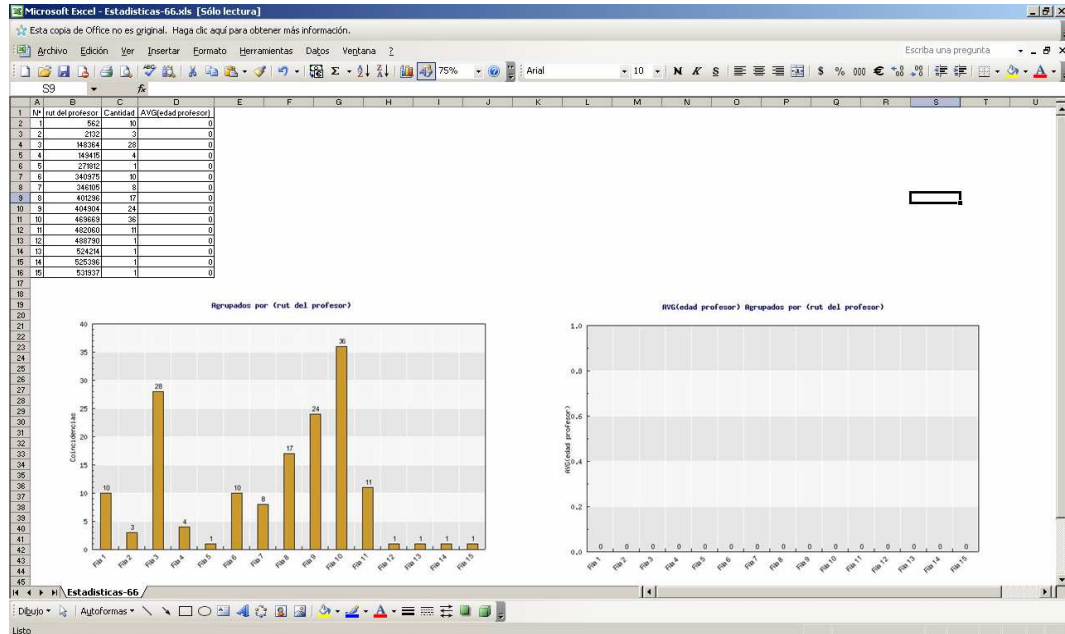


Fig 7.30 Reporte final exportado a un archivo Excel

Además es posible exportar los resultados en un archivo Excel pero sin gráficos. Si se utiliza el ejemplo anterior el resultado o archivo exportado se vería como se muestra en la siguiente imagen:

A	B	C	D	E	F	G	H	I	J	K	L	M	N
1	Nº	rut del profesor	Cantidad	AVG(edad profesor)									
2	1	862	30	0									
3	2	2132	3	0									
4	3	148364	28	0									
5	4	148415	4	0									
6	5	271812	1	0									
7	6	340975	10	0									
8	7	348105	8	0									
9	8	401296	17	0									
10	9	404904	24	0									
11	10	469669	36	0									
12	11	482060	11	0									
13	12	488790	1	0									
14	13	524214	1	0									
15	14	525396	1	0									
16	15	531937	1	0									

Fig 7.31 Reporte final exportado a un archivo Excel sin gráficos

Si los resultados de la consulta son exportados a un archivo PDF entonces este archivo contará con una tabla y gráficos como los que se pueden ver en las imágenes que se presentarán a continuación:

The screenshot shows the Adobe Acrobat Professional interface with a PDF document titled 'Estadísticas.pdf'. The main content area displays a table with the following data:

N°	rut del profesor	Cantidad	Porcentaje profesor
1	562	10	0.0000
2	2132	3	0.0000
3	348364	28	0.0000
4	345415	4	0.0000
5	273912	1	0.0000
6	240975	10	0.0000
7	246105	8	0.0000
8	401296	17	0.0000
9	404994	24	0.0000
10	469669	36	0.0000
11	482809	11	0.0000
12	488790	1	0.0000
13	524214	1	0.0000
14	528396	1	0.0000
15	533337	1	0.0000

Fig 7.32 Tabla del reporte exportado a un archivo PDF

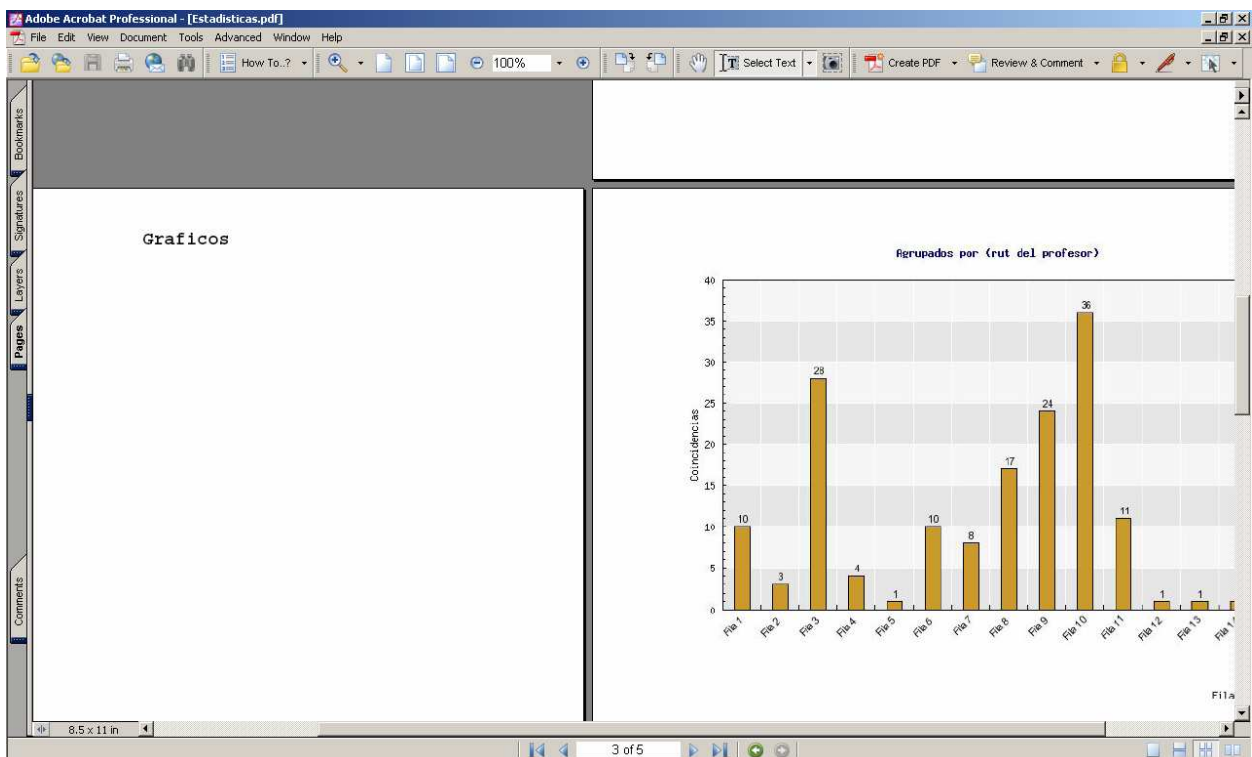


Fig 7.33 Gráficos de un reporte exportado a un archivo PDF

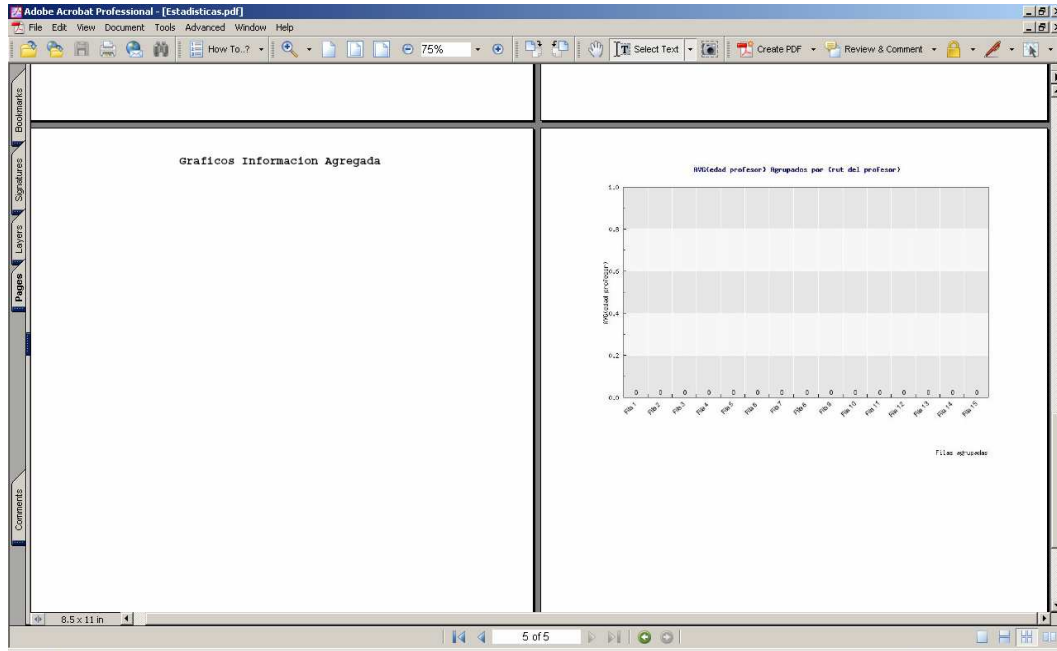


Fig 7.34 Gráficos de funciones agregadas de un reporte exportado a un archivo PDF

Las consultas estadísticas exportadas como archivos WORD o RTF contarán con la misma estructura y gráficos que en el caso de los archivos PDF, es decir, con una tabla de resultados y gráficos y funciones agregadas si corresponde.

7.3 Módulo de creación de un reporte

Este módulo permite crear la estructura ROLAP estrella de un reporte a partir de los campos definidos por el diseñador del reporte en la función asociada. En la imagen que se muestra a continuación puede verse la interfaz con los campos definidos por el diseñador previamente:

Nombre Campo	Nombre Dimension	Titulo	Tipo?	¿Se puede Agrupar?	¿Filtro Seleccionar sus valores?	Tipo tabla?	Descripcion
ANO			INT	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	Dimension	
MES			INT	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	Dimension	
DIA			INT	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	Dimension	
RUT			INT	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	Dimension	
DV			INT	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	Dimension	
NOMBRE			INT	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	Dimension	
TITULO			INT	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	Dimension	
RUT_PROFESOR			INT	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	Dimension	
DV_PROFESOR			INT	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	Dimension	
NOMBRES_PROFESC			INT	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	Dimension	
APELLIDOS_PROFESC			INT	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	Dimension	
TIPO_PROFESOR			INT	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	Dimension	
TEMA			INT	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	Dimension	
EDAD_PROFESOR			INT	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	Dimension	
DEPARTAMENTO			INT	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	Dimension	
JERARQUIA			INT	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	Dimension	

Fig 7.35 Módulo de creación de un nuevo reporte

En la imagen anterior se puede ver los campos necesarios para crear un reporte. Primero se debe definir el nombre del reporte y el tipo de actualización sobre la información de este. Luego es necesario definir el nombre de la dimensión donde será almacenado cada uno de los campos, el título con el cual el usuario leerá el campo, el tipo de dato y una descripción del campo. También se debe marcar si se podrá realizar filtros por los valores que tomará el campo en la interfaz de consulta (esto debido al gran número de valores distintos que puede tomar un campo) y si estará permitido agrupar por él. Finalmente, es necesario definir si el campo es parte de una dimensión o parte de la tabla de hechos.

Si el reporte ya fue creado anteriormente al intentar acceder a la interfaz de creación del reporte el usuario será redirigido al módulo de inicio, en el cual se le desplegará un mensaje como el que puede verse en la siguiente imagen:

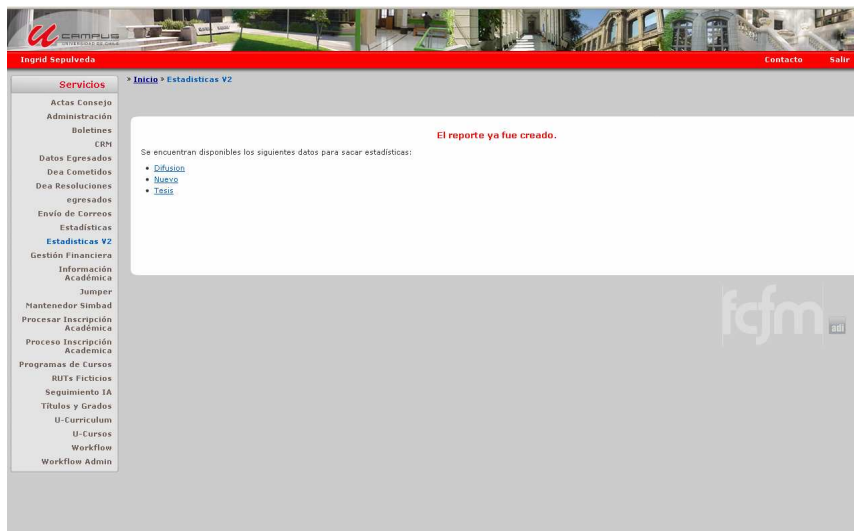


Fig 7.36 Mensaje desplegado al intentar crear un reporte que ya existe

En este módulo durante la creación de un reporte se realizan validaciones y se alerta al usuario con mensajes emergentes sobre las acciones inválidas o corregidas, estas alertas son las siguientes:

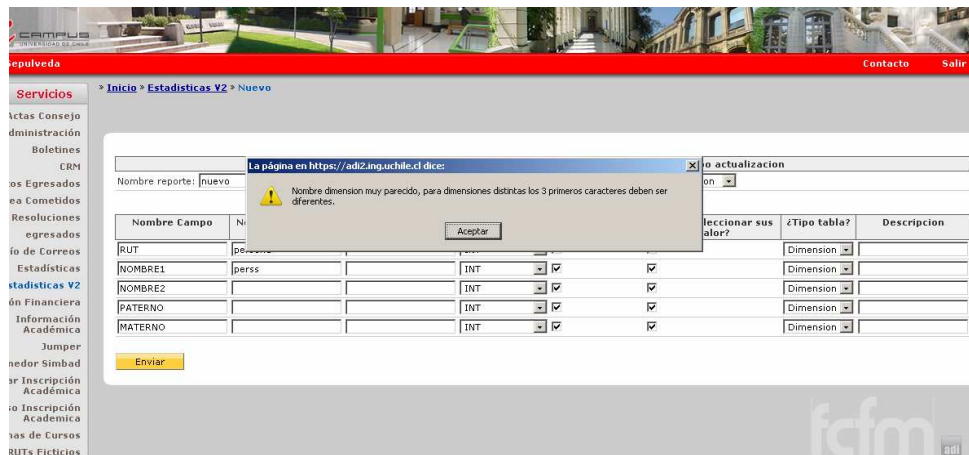


Fig 7.37 Nombres de dimensiones similares

No está permitido definir nombres de dimensiones distintas cuyas primeras tres letras sean iguales, esto debido a que las primeras tres y cuatro letras son utilizadas para definir el nombre de las columnas de las tablas. Tampoco es posible definir un mismo nombre de campo en una misma dimensión, en este caso, el último nombre de dimensión escrito por el usuario es cambiado a vacío.

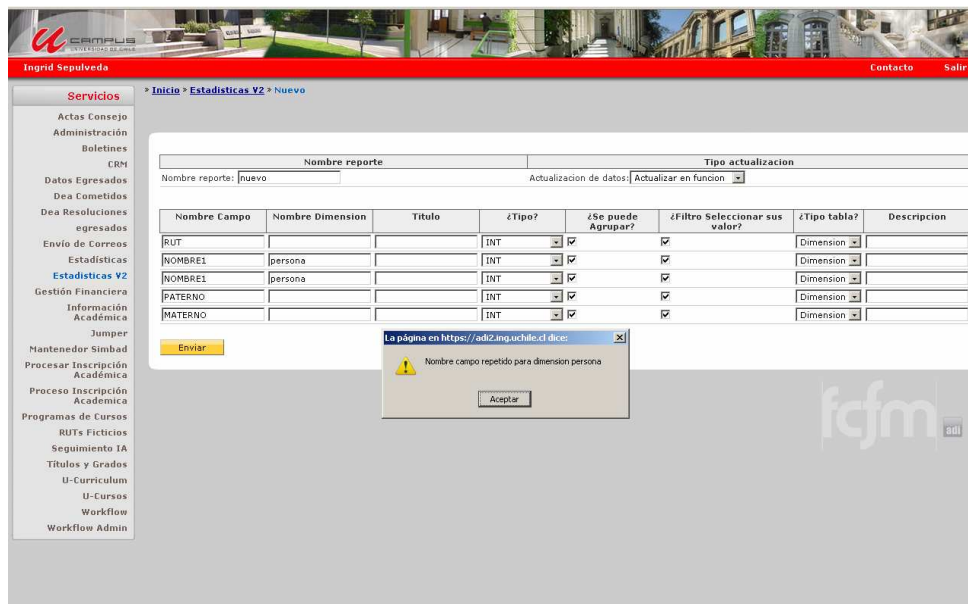


Fig 7.38 Definición de un nombre de campo repetido para una misma dimensión

Otra restricción es definir el nombre de una dimensión con el mismo nombre que el reporte, con esto se asume que el usuario está definiendo el campo en la tabla de hechos y no en una dimensión. Al ocurrir esto el valor del nombre de la dimensión que produjo el problema se cambia a vacío.

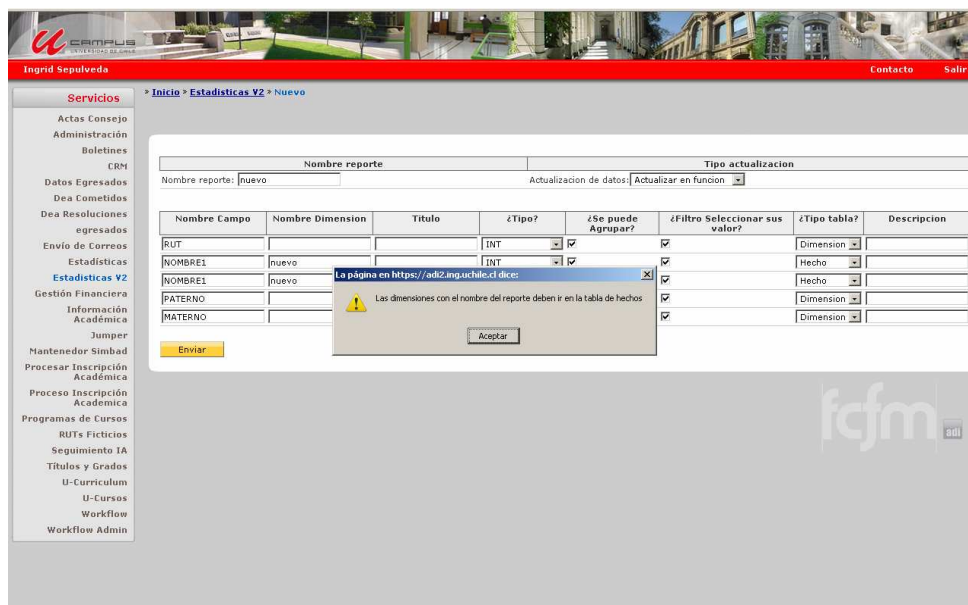


Fig 7.39 Definición de una dimensión con el nombre del reporte

Al momento de enviar los datos para crear un reporte se efectuarán otras verificaciones, como definir al menos una dimensión en el modelo ROLAP estrella, en esos casos, un mensaje de alerta se le desplegará al usuario.

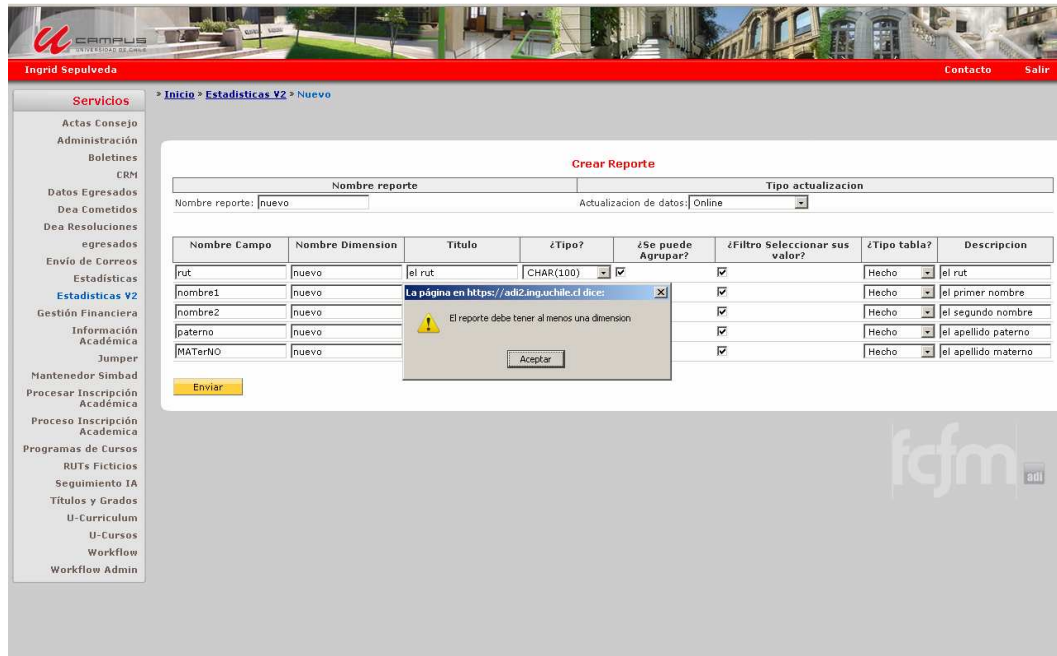


Fig 7.40 Cada reporte ROLAP estrella debe definir al menos una dimensión

Además de la verificación del número de dimensiones se comprobará que los campos título, descripción y nombre de la dimensión sean definidos con un valor distinto a vacío. Esto puede verse en la siguiente imagen:

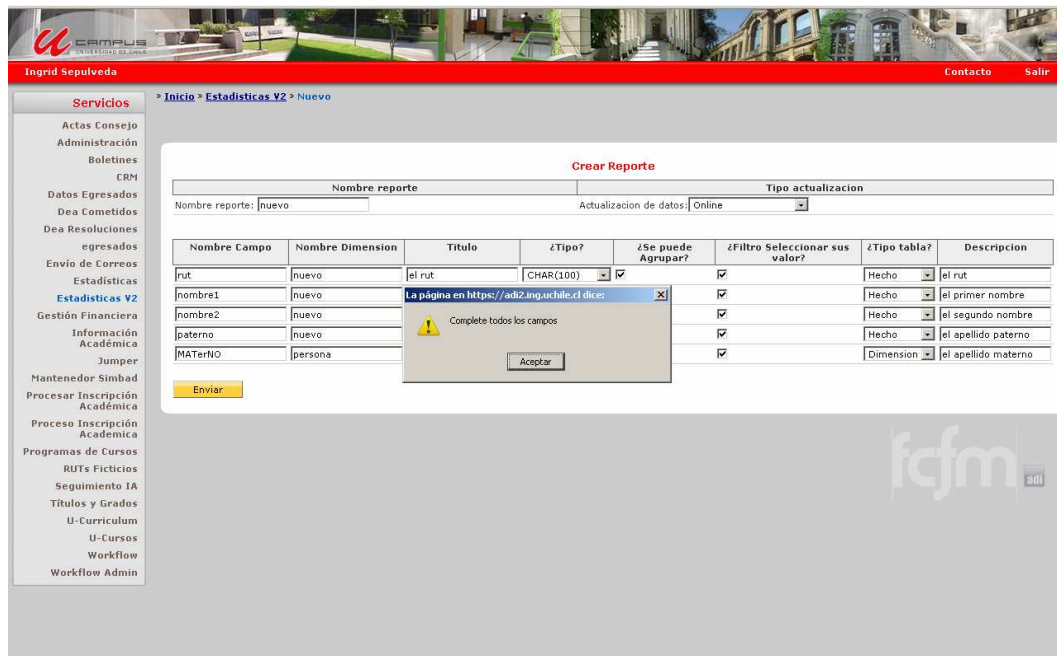


Fig 7.41 Validación de campos completados

Cuando finalmente el modelo de datos ROLAP estrella del reporte es creado con éxito el usuario diseñador del modelo será redireccionado al módulo de inicio del sistema con un mensaje como el que se aprecia en la siguiente imagen:

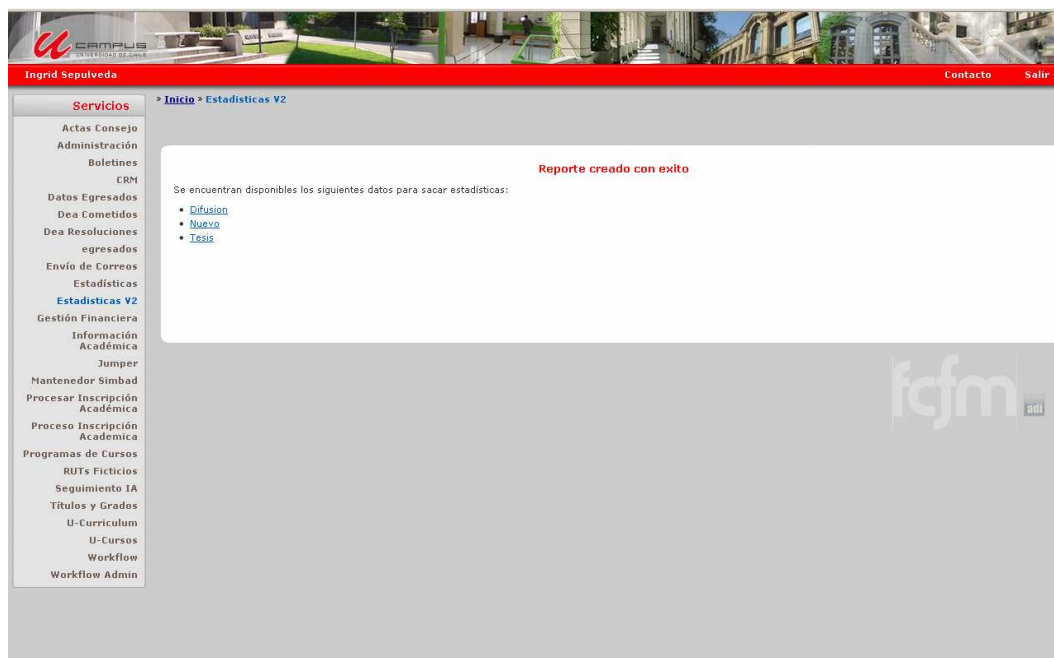


Fig 7.42 Reporte creado exitosamente

7.4 Módulo de campos visibles

El módulo de modificar los campos visibles solo experimentó pequeños cambios en la interfaz al centrar los títulos de cada columna. En esta interfaz al seleccionar la casilla de verificación ubicada al lado derecho del título “Visible”, se marcan automáticamente todas las casillas de cada campo. En caso de no seleccionar ningún campo todos ellos serán visibles en el módulo de consultas.



Fig 7.43 Módulo de selección de campos visibles

8 Pruebas del sistema e implantación

Se realizaron pruebas unitarias al sistema, de manera de comprobar que los resultados cumplen con los requisitos y que son correctos. Primero se realizaron pruebas en el ambiente de desarrollo, posteriormente en el ambiente de ADI, para así luego de corregir los problemas y errores detectados instalar y probar finalmente el sistema en pre-producción.

En las pruebas realizadas se distinguen los usuarios con permiso de AGREGAR (asociado al usuario Administrador) y los que no lo tienen (los Clientes); también existirá el usuario que creó un reporte (el Dueño) y el que creó y guardó una consulta (el Propietario). Las funcionalidades disponibles variarán de acuerdo a si se tiene o no este permiso. Además, un usuario podrá pertenecer a uno o más grupos de reportes, por lo cual podrá tener acceso a reportes distintos de acuerdo al grupo.

8.1 Pruebas del sistema

A continuación se detallarán cada una de las pruebas realizadas al sistema y los resultados esperados para cada una de ellas, separadas por módulos. El sistema cumple con todas pruebas satisfactoriamente.

8.1.1 Módulo: Inicio

En este módulo se despliegan los reportes disponibles, es decir, aquellos que tienen definida una función asociada. Los reportes se definen por grupos, es decir, habrá diferentes reportes de acuerdo al grupo que se encuentra visitando el usuario.

Función	Cargo	Prueba	Resultado esperado
Desplegar reportes disponibles	Todos	Desplegar los reportes existentes.	Se imprimirá en pantalla una lista ordenada de los nombre de los reportes disponibles para el grupo en el cual se encuentra.

8.1.2 Módulo: Crear un reporte

El módulo de creación de reportes se debe iniciar una vez que se ha definido la función asociada al reporte, es por esto que no es valido crear uno nuevo sin una función asociada definida.

Función	Cargo	Prueba	Resultado esperado
Validar permisos	Todos	Validar los permisos de acceso a la creación de un reporte.	Solo pueden acceder a esta página los usuarios con permiso AGREGAR. Los usuarios sin este permiso deben ser redireccionados al módulo inicio con un mensaje "Debe tener los permisos necesarios para esta acción.". Si el usuario es valido se le

			desplegará la interfaz de creación de un reporte.
Crear un reporte sin función asociada	Administrador	Verificar que al intentar acceder al módulo la función asociada este definida	No está permitido acceder al módulo de creación de la estructura de un reporte si la función asociada no existe. En este caso el usuario es redirigido al módulo de inicio con el mensaje “No ha definido la función asociada”
Crear un reporte	Administrador	Acceso a la interfaz de creación de un reporte.	Si un reporte asociado a una función ya existe, entonces el usuario es redirigido al módulo de inicio con el mensaje “Ya existe un reporte con ese nombre”. Si todo es valido, se despliega la interfaz con todas las opciones para crear un reporte. Los campos del reporte están definidos en los cuadros de texto de solo lectura “Nombre campo” y el nombre del reporte en el campo “Nombre reporte”. Además es necesario definir la “Actualización de datos”, el “Nombre Dimensión”, el “titulo”, el “Tipo” de dato del campo, definir si “Se puede Agrupar”, si es posible hacer un “Filtro Seleccionar sus valores”, el “¿Tipo Tabla?” y
Verificación/ayuda de campos en crear un reporte	Administrador	En la interfaz de creación de reportes se realizan una serie de cambios y alertes.	<p>Una descripción de campo.</p> <ul style="list-style-type: none"> • Dos nombres de dimensiones distintas con sus 3 primeras letras iguales. • Definir dos nombres de campos iguales en una misma dimensión. • Campos definidos en la tabla de hechos tengan un nombre de dimensión diferente al del reporte. Esto se

			<p>realiza completamente transparente para el usuario.</p> <ul style="list-style-type: none"> • No puede definirse una dimensión con el mismo nombre que el reporte. • Los caracteres NO alfanuméricos se eliminan de los nombres de las dimensiones de forma transparente para el usuario. <p>La interfaz ayuda al usuario con lo siguiente:</p> <ul style="list-style-type: none"> • Al seleccionar un campo como Hecho, el nombre de la dimensión se auto-completa con el nombre del reporte. • Al definir en el nombre de dimensión un nombre igual al nombre del reporte, automáticamente se marca como dimensión • Al seleccionar un campo como dimensión, el nombre de la dimensión se hace vacío, esto se hace de manera transparente para el usuario.
Verificación de campos antes de crear un reporte.	Administrador	Se realizan una serie de validaciones antes de enviar la solicitud de creación de un reporte.	<p>Se verifica lo siguiente antes de enviar la solicitud:</p> <ul style="list-style-type: none"> • Que se defina al menos una dimensión. • Que se completen los siguientes campos: Nombre de la dimensión, título y descripción de todas las
Crear un reporte.	Administrador	Se envía la solicitud de creación de un reporte.	<p>Si el reporte es creado con éxito, el usuario es redirigido al módulo de inicio con el mensaje "Reporte creado con éxito"</p>

			En caso de problemas o errores se desplegará en pantalla la consulta SQL que generó el error.
--	--	--	---

8.1.3 Módulo: Consultas

Este es el módulo donde se llevan a cabo las consultas a los reportes. Pese a ser un único módulo se dividirá en secciones de acuerdo a cada funcionalidad.

8.1.3.1 Actualización de información

Función	Cargo	Prueba	Resultado esperado
Seguir un reporte	Todos	Acceder a un reporte	Si el tipo de actualización del reporte es en línea, entonces los datos de la función asociada son actualizados en la BDEE. Si la actualización ocurre con éxito entonces el usuario accede al módulo de consultas y se despliega el mensaje “Reporte poblado con éxito.”. Al contrario se despliega el mensaje “Se intento poblar el reporte, pero no hay datos desde la función asociada.” redirigiéndose al usuario al
Bloquear un reporte	Cliente	Actualizar un reporte	Cuando se inicia actualizando un reporte, este no está disponible para los usuarios y se despliega el mensaje “El reporte no esta accesible por el momento.”
Actualización en curso	Todos	Existe un proceso recuperando la información desde la fuente externa.	Cuando existe un proceso rescatando la información desde la fuente externa pero aun no comienza la actualización del reporte, los usuarios podrán acceder normal y completamente transparente al módulo de consultas.

8.1.3.2 Campos en los resultados

Los campos desplegados en los resultados de una consulta deben corresponder a los marcados por el usuario, ya sea en una agrupación, una transposición o una selección

simple. Dependiendo del caso podrá haber campos especiales como lo son los porcentajes, cantidad, totales o alguno definido por una función agregada.

8.1.3.3 Pagar los resultados

Función	Cargo	Prueba	Resultado esperado
Mostrar hojas	Todos	Mostrar las hojas de cada reporte.	Se muestran las hojas en las que se puede ir viendo los resultados de los reportes.
Número de resultados	Todos	Número de resultados.	Se despliega el número de resultados totales y el número de resultados que se están viendo en la hoja.
Resultados por hoja	Todos	Los resultados de la consulta de la hoja	Todas las hojas despliegan la información en igual formato, solo varían los datos mostrados.

8.1.3.4 Exportar a DOC, RTF, PDF y Excel

Función	Cargo	Prueba	Resultado esperado
Desplegar opciones de exportar	Todos	Mostrar los botones de cada acción si corresponde	Son cinco botones: Bajar Excel, Bajar a Excel sin gráficos, Bajar como PDF, Bajar como DOC y Bajar como RTF. Estos botones deben mostrarse siempre cuando el conjunto de resultados de la consulta sea el botón.
Exportar como PDF, RTF o DOC	Todos	Validar el formato de salida de los archivos.	Al pulsar en los botones se podrá descargar un archivo de acuerdo al formato seleccionado. El formato de cada archivo es el siguiente: <ul style="list-style-type: none"> • La primera hoja es la tabla con los datos y un título “Estadísticas”. • Luego una hoja con el título “Gráficos” solo si hay gráficos que mostrar. Luego en las hojas posteriores se muestran 2 gráficos por hoja. • En otra hoja se muestra el título “Gráficos Información Agregada” solo si se ha seleccionado alguna función agregada. • Finalmente, se muestran los gráficos agregados, en cada hoja se insertan 2 gráficos al igual que en el caso anterior.
Exportar a	Todos	Exportar a	La tabla está alineada en el ángulo

Excel		Excel	superior izquierda. Los gráficos están dispuestos en dos columnas y listados hacia abajo
Gráficos embebidos	Todos	Ver los gráficos	Los gráficos están embebidos en el archivo de cada formato, por lo cual es posible verlos sin conexión a Internet. En el caso del archivo Excel sin gráficos, no se verá ninguna imagen.
Mismo contenido en los archivos exportados.	Todos	Mismo contenido en todos los formatos y consulta.	En todos los archivos exportados debe verse la misma información y gráficos. Esta información es la misma de la hoja en la cual se estaban viendo los resultados del módulo de consultas.

8.1.3.5 Cargar una consulta desde el sistema

Función	Cargo	Prueba	Resultado esperado
Desplegar consultas guardadas.	Propietario	Mostrar un menú de las consultas	Si hay consultas disponibles se muestran en un menú. Estas consultas están disponibles para un grupo y un propietario específico.
Cargar una consulta	Propietario	Cargar una consulta.	Si se selecciona una consulta desde el menú se carga en el módulo de consultas. Si se carga con éxito se muestra el mensaje “La consulta fue cargada exitosamente.” en el módulo de consultas. Si hay algún problema se muestra el mensaje “No se pudo cargar la consulta.” en el módulo de consultas.
Seleccionar la consulta blanca	Propietario	Seleccionar la consulta blanca	Al seleccionar la opción blanca del menú de consultas se quita la consulta cargada.

8.1.3.6 Cargar una consulta desde disco

Función	Cargo	Prueba	Resultado esperado
Seleccionar un archivo para cargar	Propietario	Abrir un archivo	Al presionar el botón examinar se puede seleccionar el archivo de estadísticas a cargar. Si se presiona el botón Cargar se carga en el módulo de consultas el archivo
Cargar una consulta	Propietario	Cargar una consulta	Si se selecciona bien se despliega el

consulta		consulta desde un archivo en disco	<p>mensaje “La consulta fue cargada exitosamente” en el módulo de consultas.</p> <p>Si los datos no se pueden cargar entonces se muestra el mensaje “No se pudo cargar la consulta.” en el módulo de consultas.</p>
----------	--	------------------------------------	---

8.1.3.7 Borrar una consulta

Función	Cargo	Prueba	Resultado esperado
Mostrar opción de borrar consulta	Propietario	Desplegar en pantalla la opción de borrar una consulta	Si el propietario cuenta con los permisos de AGREGAR podrá ver la opción en pantalla.
Borrar una consulta	Propietario	Borrar una consulta	<p>Una vez cargada una consulta podrá borrarse presionándose el botón Borrar. Si la consulta se borra con éxito se muestra el mensaje “La consulta fue borrada correctamente” en el módulo de consultas.</p> <p>Si hay algún error mientras se intenta borrar la consulta, se muestra el mensaje “Se ha producido un error al momento de borrar la consulta” en el módulo de consultas</p> <p>Si la consulta no es propiedad del usuario se le muestra el mensaje “Ud no posee permiso para borrar esta consulta” en el módulo de consultas.</p>

8.1.3.8 Actualizar una consulta

Función	Cargo	Prueba	Resultado esperado
Mostrar opción de actualizar	Propietario	Desplegar en pantalla la opción de actualizar una consulta	Si el propietario cuenta con los permisos de AGREGAR podrá ver la opción en pantalla.
Actualizar una consulta	Propietario	Actualizar una consulta	Una vez cargada una consulta podrá actualizarse presionándose el botón Actualizar. Si se actualiza exitosamente se muestra el mensaje “La consulta ha sido actualizada”

			<p>en el módulo de consultas.</p> <p>Si ocurre un error mientras se intenta actualizar, se muestra el mensaje “Se ha producido un error al momento de actualizar la consulta” en el módulo de consultas</p> <p>Si no se ha seleccionado ninguna consulta se muestra el mensaje “No ha seleccionado ninguna consulta” en el módulo de consultas</p> <p>Si la consulta no es propiedad del usuario se muestra el mensaje “Ud no posee permiso para actualizar esta consulta” en el módulo de consultas.</p> <p>Si el usuario no cuenta con permisos de AGREGAR se muestra el mensaje “Ud no posee permiso para actualizar esta consulta” en el módulo de consultas.</p>
--	--	--	---

8.1.3.9 Guardar una consulta en el sistema

Función	Cargo	Prueba	Resultado esperado
Mostrar opción guardar	Propietario	Desplegar en pantalla la opción de guardar	Si el propietario cuenta con los permisos de AGREGAR podrá ver la opción en pantalla.
Guardar una consulta en el sistema	Propietario	Guardar una consulta en el sistema	<p>Al presionar el botón Guardar se guarda en el sistema.</p> <p>Si la consulta se guarda exitosamente se despliega el mensaje “La consulta se ha guardado correctamente” en el módulo de consultas.</p> <p>Si ocurre un error mientras se intentaba guardar una consulta se despliega el mensaje “Se ha producido un error al momento de guardar la consulta” en el módulo de consultas.</p> <p>Si el nombre de la consulta es blanca se despliega el mensaje</p>

			<p>“Debe ingresar un nombre para guardar la consulta” en el módulo de consultas.</p> <p>Si el usuario no cuenta con permisos de AGREGAR se muestra el mensaje “Ud no posee permiso para guardar esta consulta” en el módulo de consultas.</p>
--	--	--	---

8.1.3.10 Guardar una consulta en disco

Función	Cargo	Prueba	Resultado esperado
Mostrar opción guardar	Propietario	Desplegar en pantalla la opción de guardar	Si el propietario cuenta con los permisos de AGREGAR podrá ver la opción en pantalla.
Guardar una consulta en disco	Propietario	Guardar una consulta en disco	<p>Al presionar el botón Guardar en disco se generará un archivo para descargar el archivo.</p> <p>Si ocurre un error mientras se intentaba guardar una consulta se despliega el mensaje “Se ha producido un error al momento de guardar la consulta” en el módulo de consultas.</p> <p>Si el nombre de la consulta es blanca se despliega el mensaje “Debe ingresar un nombre para guardar la consulta” en el módulo de consultas.</p> <p>Si el usuario no cuenta con permisos de AGREGAR se muestra el mensaje “Ud no posee permiso para guardar esta consulta” en el módulo de consultas.</p>

8.1.3.11 Ordenar resultado por columnas

Función	Cargo	Prueba	Resultado esperado
Seleccionar ordenar los campos resultados	Todos	Seleccionar como ordenar un campo en la salida.	<p>Solo se podrá elegir la forma de ordenación en los campos agrupados.</p> <p>Si se transpone un campo, este campo no puede ser ordenado.</p>

			Si no hay ningún campo agrupado se podrá elegir el orden por todos ellos.
--	--	--	---

8.1.3.12 Filtrar por algún campo

Función	Cargo	Prueba	Resultado esperado
Filtrar por un campo	Todos	Seleccionar un criterio de selección	<p>Todos los campos aceptan un criterio de filtro que puede ser:</p> <ul style="list-style-type: none"> • Seleccionar uno: Solo si se definió el campo como seleccionable al crear el reporte. Esta opción genera un menú con todos los valores posibles. • Seleccionar varios: Genera un menú con las opciones que permite seleccionar diferentes valores presionando el botón control y marcando la opción en Mostrar. • No está entre: Mostrarán dos campos para escribir valores. • Mayor que: Escribir un valor • Menor que: Escribir un valor • Mayor o Igual que: Escribir un valor • Menor o Igual que: Escribir un valor • Que calce la Expresión Regular: Escribir un valor • Que NO calce la Expresión Regular: Escribir un valor.

8.1.3.13 Agrupar

Función	Cargo	Prueba	Resultado esperado
Campos agrupables	Todos	Campos agrupables	Solo se podrá agrupar por campos agrupable definidos en la creación

		validos	del reporte.
Seleccionar un campo para agrupar	Todos	Seleccionar un campo para agrupar	<p>Al presionar un campo para agruparlo se ocultan los checkbox de mostrar.</p> <p>Al seleccionar más de 1 campo para agrupar se da la opción de transponer uno y solo uno de ellos. Los campos agrupados y no transpuestos podrán ser ordenados.</p> <p>Al seleccionar un campo para agrupar aparecen más opciones (Campos especiales): Solo contar, totales, porcentajes, funciones agrupadas y ordenar las cantidades.</p> <p>La opción de porcentajes tiene prioridad sobre el total y las cantidades, y son mutuamente excluyentes.</p> <p>Las funciones agregadas solo se aplican sobre los campos definidos como números. Las opciones de funciones agregadas son las siguientes:</p> <ul style="list-style-type: none"> • Promedio • Máximo • Mínimo • Desviación estándar • Varianza
Ejecutar una consulta de agrupar	Todos	Ejecutar una consulta de agrupar	Al agrupar se despliegan los resultados en una tabla y gráficos.

8.1.3.14 Transponer

Función	Cargo	Prueba	Resultado esperado
Transponer	Todos	Transponer	<p>Al seleccionar más de un campo para agrupar, se podrá definir uno de ellos para transponerlo. El campo a transponer no puede ser ordenado, el resto de los campos agrupables sí.</p> <p>Al transponer no se pueden hacer consultas agregadas.</p>

			El campo transpuesto se mostrará horizontalmente
Ejecutar una consulta de transponer	Todos	Ejecutar consulta de transponer	Al agrupar se despliegan los resultados en una tabla y también gráficos.

8.1.3.15 Tabla de resultados

Función	Cargo	Prueba	Resultado esperado
Porcentajes y transponer	Todos	Porcentajes y transponer	<p>Para poder calcular porcentajes es necesario seleccionar al menos un campo para agrupar.</p> <p>Los porcentajes de las columnas transpuestas corresponden a las cantidades de cada celda dividida por la suma total de la columna.</p> <p>Las columnas transpuestas se ordenan horizontalmente, con una columna por cada valor del campo transpuesto. Las otras columnas aparecen a la izquierda de estas.</p>
Porcentajes y agrupar	Todos	Porcentajes y agrupar	<p>Para poder calcular porcentajes es necesario seleccionar al menos un campo para agrupar. Se muestran las columnas agrupadas y los porcentajes de cada fila.</p> <p>Los porcentajes corresponden al total de cada fila dividido el total de la suma de todas las filas del reporte que satisfacen la condición (incluyendo las que no aparecen en la página)</p>
Sin opción de porcentajes y transponer	Todos	Sin opción de porcentajes y transponer	<p>Totales calcula la suma total de cada columna y del total de la tabla del reporte.</p> <p>Las columnas transpuestas se ordenan horizontalmente, las no transpuestas y agrupadas se muestran como columnas a la izquierda</p>
Sin opción de porcentajes y agrupar	Todos	Sin opción de porcentajes y agrupar	<p>La opción total muestra la suma de las columnas cantidades. La opción de “Solo contar” calcula las cantidades por fila.</p>

			Ordenar por cantidades ordena las filas por los valores de sus cantidades ascendente o descendente.
No transponer ni agrupar	Todos	No transponer ni agrupar	Se despliegan los campos chequeados de la columna mostrar.

8.1.3.16 Graficar

Para mostrar gráficos debe agruparse por al menos un campo.

Función	Cargo	Prueba	Resultado esperado
Porcentajes y Transponer	Todos	Porcentajes y Transponer	Se genera un gráfico de torta por cada columna transpuesta.
Porcentajes y agrupar	Todos	Porcentajes y agrupar	Un gráfico de torta con los porcentajes de cada fila, correspondiente a la columna porcentajes.
Sin porcentajes y Transponer	Todos	Sin porcentajes y Transponer	Si hay 2 o más campos agrupados, considerando el campo transpuesto, entonces se generará un gráfico de líneas. Para cada una de las líneas se generará un gráfico de barras independiente. También se generará un gráfico de barras
Sin porcentajes y agrupar.	Todos	Sin porcentajes y agrupar.	Se generará un gráfico de barra y otro de línea si el número de campos agrupados es mayor que
Gráfico de funciones agregadas	Todos	Gráfico de funciones agregadas	Cada función agregada genera un gráfico de líneas si el número de campos agrupados es mayor que 1. También se generará un gráfico de barras para cada función agregada.

8.1.3.17 Modificar la visibilidad de un reporte internamente (En la base de datos)

Función	Cargo	Prueba	Resultado esperado
Modificar la visibilidad	Administrador de base de datos	Al cambiar el campo COL_VISIBL E en la tabla de columnas de los reportes, el cambio debe reflejarse en los campos que el usuario	Al cambiar el valor de la columna COL_VISIBLE a 0 el campo no debe aparecer en ninguna interfaz del sistema: Campos visible, consultas ni en los resultados de consultas.

		puede seleccionar para ver.	
--	--	-----------------------------	--

8.1.4 Módulo: Campos visibles

Función	Cargo	Prueba	Resultado esperado
Verificación nombre reporte	Todos	Verificar el nombre del reporte.	Si el reporte no existe se redirige al usuario al módulo inicio con el mensaje “Reporte no existe.” Si todo es correcto se despliega la interfaz de campos visibles.
Acceder a interfaz campos visibles.	Todos	Acceder exitosamente a la interfaz.	Permite seleccionar entre los campos disponibles en el reporte. Deben encontrarse marcados todos los campos que se pueden ver en el módulo de consultas, cada uno de ellos con su título y descripción. Al marcar un campo, este se podrá ver en el módulo de consultas. Al marcar el cuadro al lado derecho del título “Visible“ se marcan automáticamente o se desmarcan todos los campos según
Guardar preferencias	Todos	Guardar los cambios de campos visibles	Si responde a la consulta exitosamente el usuario es redirigido al módulo de consultas donde verá solo los campos seleccionados. Si ningún campo es seleccionado, entonces en el módulo de consultas se desplegarán todos los campos.

8.2 Implantación en producción

El paso de un servicio desarrollado en el ADI a un ambiente de producción debe cumplir con las siguientes etapas:

1. Pasar el módulo desde la cuenta del desarrollador a un servidor de prueba. Cada archivo traspasado debe estar versionado utilizando SVN, herramienta que también permite detectar los archivos que no han sido traspasados.
2. Se procede a revisar el código fuente del módulo para verificar que cumpla los estándares de programación definidos por el ADI.
3. Se pasa el módulo a un servidor de pre-producción en el cual se realizan pruebas de usuario.

4. Luego de esta revisión y de haberse corregidos los errores y bugs detectados, el módulo puede pasar a producción.

Las tareas a realizar para activar el servicio son las siguientes:

1. Bajar desde SVN el módulo en su ubicación definitiva.
2. Configurar los parámetros del módulo en las llaves de InfoNúcleo.
3. Importar la base de datos del sistema.
4. Asignar a los usuarios los permisos en el sistema.

9 Conclusiones

El diseño de un sistema de apoyo a la generación de reportes para la Facultad es una tarea compleja, puesto que la información está distribuida en diferentes fuentes de información y los volúmenes pueden volver ineficientes a las aplicaciones que no están preparadas para manejarlos. Estas son algunas de las deficiencias con las cuales contaba SIESTA, que fueron mejoradas en la nueva implementación del sistema SIAGER.

Durante la etapa de análisis de los requerimientos de SIAGER se realizaron entrevistas a los usuarios frecuentes, lo que permitió conocer de primera fuente las necesidades y los reportes frecuentemente consultados y generados por estos. Para estos reportes se plantearon nuevos modelos de datos que pueden ser implementados fácilmente en el sistema y así responder satisfactoriamente a las necesidades de los usuarios. Estos modelos además brindan a la Facultad una plataforma que resume, compila y centraliza la información, evitando la consulta de variadas fuentes.

Para resolver las deficiencias de SIESTA en el nuevo sistema, primero se analizaron a fondo sus problemas para mejorarlos o realizar una implementación distinta. Una de las mejoras implementadas en SIAGER fue la utilización del modelo de datos ROLAP estrella, el cual redujo significativamente los tiempos de respuesta de las consultas y en un 58% el espacio en disco utilizado, de acuerdo al ejemplo de prueba analizado en este trabajo. Al modificar la lógica utilizada para realizar las consultas, se redujo la utilización de la memoria en alrededor de 11 veces. También se modificó la lógica de actualización de los reportes, lo cual se espera mejore la experiencia del usuario, puesto que no habrá retardos durante la actualización.

Finalmente, al orientar la aplicación hacia la generación de reportes finales se logró resolver una de las necesidades más importantes para los usuarios del sistema, los cuales a partir de ahora podrán contar con los reportes en un formato adecuado para su lectura y distribución.

9.1 Contribuciones

Al conocer a fondo la situación actual se detectaron las ventajas y desventajas de la aplicación SIESTA, las cuales fueron aprovechadas, corregidas, mejoradas o solucionadas. Además se plantearon cinco modelos ROLAP estrella que responden a las consultas frecuentes de información de parte de los usuarios, lo cual agregará un considerable valor a la Facultad en cuanto a gestión, administración y análisis.

También se mejoró la utilización de los recursos disponibles en el servidor Web y de bases de datos al reducir la utilización de memoria, disco y los tiempos de respuesta. Estas reducciones le permitirán a la Facultad bajar sus costos de mantención, además de permitir un crecimiento y evolución controlada del sistema.

Desde ahora los usuarios podrán acceder a un único servicio que responde de la mejor manera a sus necesidades. Se espera que la información se almacene resumida y constituya el canal único de consulta por parte de los usuarios, que dependerán en una menor medida de administradores o expertos.

Al estar integrado a una plataforma ya conocida por los usuarios, se espera una disminución en el costo de aprendizaje y en su aceptación. El sistema se integra a la plataforma U-Campos, cuyas funcionalidades y resultados son conocidos por los usuarios. Las interfaces y modo de uso también resultan familiares para ellos.

9.2 Limitaciones

Como hemos señalado anteriormente el sistema utiliza un modelo ROLAP estrella para almacenar los datos de cada reporte. Este modelo es potente y se ha difundido a lo largo de los años, llegando a consolidarse. Si bien este modelo señala una serie de recomendaciones a seguir para asegurar un buen diseño, en la implementación del sistema SIAGER se han relajado algunas de estas recomendaciones.

Además de las limitaciones anteriores se consideran las funcionalidades que no fueron implementadas y los trabajos futuros que pueden ser desarrollados para mejorar o complementar el sistema actual. A continuación se analizan las limitaciones y el trabajo futuro de la aplicación.

9.2.1 Tablas de dimensiones compartidas

Dado que no se cuenta con vistas en la versión del motor de bases de datos utilizado, se deben realizar copias de tablas para obtener los beneficios de las dimensiones conformadas y compartidas. Al copiar las tablas dimensiones entre los diferentes modelos ROLAP estrella se asegura su independencia en el proceso de actualización de la información. Si bien sería posible que una dimensión estuviera relacionada con más de una tabla de hechos, la implementación se volvería compleja de entender y violaría los estándares de definición de modelos de datos del ADI, puesto que una columna de una tabla específica tendría nombres distintos dependiendo de la tabla de hechos con la cual está relacionada.

Los índices definidos sobre cada dimensión (en base a las consultas que sobre ella se realizan) constituyen otra buena razón para la creación de tablas distintas e independientes. Las dimensiones son el punto de entrada a la tabla de hechos, por lo cual los filtros se realizan sobre estas y no sobre los hechos, esto puede dar lugar a una definición de índices distintos sobre dimensiones con estructuras similares.

9.2.2 Unión de dos tablas de hechos

La combinación o unión de dos tablas de hechos se considera una acción no permitida y como regla general se debe evitar su ocurrencia en el motor de base de datos. Una alternativa es realizar la combinación de las consultas fuera del servidor de bases de datos, uniendo consultas ejecutadas sobre una única tabla de hechos y de forma independiente. Otro problema que puede ocurrir al combinar o unir tablas de hechos es la posible mezcla de granularidades, lo cual debería ser verificado.

9.2.3 Operador OR

El sistema SIAGER solo permite realizar filtros utilizando el operador AND para unir las diferentes condiciones. No cuenta con la posibilidad de utilizar el operador OR para unir las condiciones ni para una condición específica de forma anidada.

9.2.4 Función radio o porcentaje de funciones definidas

El sistema no implementa la funcionalidad para el cálculo de radio de funciones definidas por el usuario. Sí cuenta con la posibilidad de calcular porcentajes de forma general tanto para el caso de agrupación simple como transposición.

9.3 Trabajos futuros

En este punto se mencionan los aspectos que pueden ser mejorados y los siguientes pasos a ser implementados. El punto a mejorar más importante corresponde al mejoramiento de la interfaz de usuario de SIAGER. Por otra parte, el próximo paso es poblar los modelos ROLAP estrella que contienen la información de las estadísticas de los reportes.

9.3.1 Mejoramiento de la interfaz

Un requisito no mejorado en este trabajo, y considerada como una deficiencia relevante, es la interfaz de usuario. Este trabajo se concentró en rediseñar la capa de datos y de lógica, ofreciendo un desempeño eficiente de las consultas y actualización de la información, junto con la posibilidad de generar reportes finales, postergando este importante requerimiento para futuros trabajos.

9.3.2 Poblar los modelos ROLAP estrellas propuestos

El siguiente paso es obtener la información necesaria para realizar el poblado de los modelos propuestos en la sección 5.4 Modelos ROLAP estrella propuestos para reportes frecuentes. De esta manera se contará con los modelos de datos que cubrirán los casos frecuentes sobre consultas estadísticas.

10 Referencias

10.1 Definiciones:

- [1] **Cubo OLAP**
http://es.wikipedia.org/wiki/Cubo_OLAP

- [2] **OLAP**
<http://es.wikipedia.org/wiki/OLAP>

- [3] **ROLAP**
<http://es.wikipedia.org/wiki/ROLAP>

- [4] **MOLAP**
<http://es.wikipedia.org/wiki/MOLAP>

- [5] **Inteligencia empresarial**
[http://es.wikipedia.org/wiki/BI_\(inform%C3%A1tica\)](http://es.wikipedia.org/wiki/BI_(inform%C3%A1tica))

- [6] **OLTP**
<http://es.wikipedia.org/wiki/OLTP>

- [7] **ETL**
<http://es.wikipedia.org/wiki/ETL>

- [8] **Desarrollo en espiral**
http://es.wikipedia.org/wiki/Desarrollo_en_espiral

- [9] **Online analytical processing**
http://en.wikipedia.org/wiki/Olap#cite_note-Codd1993-0

- [10] **Esquema de estrella**
http://es.wikipedia.org/wiki/Esquema_en_estrella

- [11] **Esquema de copo de nieve**
http://es.wikipedia.org/wiki/Esquema_en_copo_de_nieve

- [12] **Tabla de dimensión**
http://es.wikipedia.org/wiki/Tabla_de_dimensi3n

- [13] **Tabla de hechos**
http://es.wikipedia.org/wiki/Tabla_de_hechos

- [14] **Modelo vista controlador**
http://es.wikipedia.org/wiki/Modelo_Vista_Controlador

- [15] **Mantenibilidad**
<http://es.wikipedia.org/wiki/Mantenibilidad>

[16] **Usabilidad**
<http://es.wikipedia.org/wiki/Usabilidad>

[17] **Minimalista**
<http://es.wikipedia.org/wiki/Minimalista>

[18] **Automatización**
<http://es.wiktionary.org/wiki/automatización>

[19] **Rendimiento**
<http://es.wikipedia.org/wiki/Rendimiento>

[20] **Snapshot**
[http://en.wikipedia.org/wiki/Snapshot_\(computer_storage\)](http://en.wikipedia.org/wiki/Snapshot_(computer_storage))

[21] **Eficiencia**
<http://es.wikipedia.org/wiki/Eficiencia>

[22] **Eficacia**
<http://es.wikipedia.org/wiki/Eficacia>

[23] **Simultáneo**
<http://es.wiktionary.org/wiki/simultáneo>

[24] **Essbase**
<HTTP://en.wikipedia.org/wiki/Essbase>

[25] **Memcached**
<http://en.wikipedia.org/wiki/Memcached>

[26] **Explain Plan**
http://en.wikipedia.org/wiki/Explain_Plan

10.2 Documentos:

[1] **Office Open XML File Formats**, ECMA, December 2006

10.3 Librerías:

[1] **PHPExcel**
<http://www.codeplex.com/PHPExcel/>

[2] **Cezpdf**
<http://manual.phpdoc.org/HTMLSmartyConverter/HandS/Cpdf/Cezpdf.html>

10.4 Libros:

[1] **The Data Warehouse Toolkit** “The Complete Guide to Dimensional Modeling” second edition, Ralph Kimball, Margy Ross.

[2] **The Data Warehouse Toolkit** “Practical Techniques for Building Dimensional Data Warehouses”, Ralph Kimball, W.H. Inmon

10.5 Memorias:

[1] Introducción al trabajo de título-CC69E:“**Diseño y construcción de un sistema que facilite el trabajo grupal integrado a la plataforma U-cursos**”, Pedro Antonio Vargas Naser, Noviembre 2007.

[2] Memoria para optar al título de ingeniero civil en computación: “**Automatización de solicitudes de alumnos en la Facultad de Ciencias Físicas y Matemáticas de la Universidad de Chile**”, Oscar Miguel De La Iglesia Caro, Diciembre 2007.

[3] Memoria para optar al título de ingeniero civil en computación: “**Sistema de titulación de la Facultad de Ciencias Físicas y Matemáticas**”, José Patricio Castro Hernández, Abril 2005.

[4] Memoria para optar al título de ingeniero civil en computación: “**Diseño y construcción de una herramienta de generación de formularios para la plataforma workflow de la Facultad de Ciencias Físicas y Matemáticas**”, Javier Rolando Villanueva González, Agosto 2005.

[5] Memoria para optar al título de ingeniero civil en computación: “**Rediseño de la interfaz cliente/colaborador de la plataforma workflow de la Facultad de Ciencias Físicas y Matemáticas de la Universidad de Chile**”, Juan Pablo Rozas Muñoz, Otoño 2007.

[6] Memoria para optar al título de ingeniero civil en computación: “**Diseño y construcción de un sistema de acta de examen electrónico para la Facultad de Ciencias Físicas y Matemáticas de la Universidad de Chile**”, Cristian Leonardo Céspedes Viñuela, Agosto 2002.

10.6 MySQL:

[1] **Especificación de tipos numéricos**
<http://dev.mysql.com/doc/refman/4.1/en/numeric-types.html>

[2] **Especificación de tipos de datos de caracteres**
<http://dev.mysql.com/doc/refman/4.1/en/char.html>

[3] **Especificación del tipo de datos YEAR**
<http://dev.mysql.com/doc/refman/4.1/en/year.html>

[4] **Comparación entre tipos de datos MySQL y Oracle**
http://download.oracle.com/docs/cd/E12151_01/doc.150/e12155/oracle_mysql_compared.htm

- [5] **Enterprise Data Warehousing with MySQL**
http://www.mysql.com/why-mysql/white-papers/mysql_wp_for_data_warehousing.pdf
- [6] **MySQL Partners**
<http://solutions.mysql.com/solutions/partners/all>
- [7] **Triggers**
<http://dev.mysql.com/doc/refman/5.0/en/create-trigger.html>
- [8] **Vistas**
<http://dev.mysql.com/doc/refman/5.0/en/create-view.html>
- [9] **Procedimientos almacenados y Funciones definidas por el usuario**
<http://dev.mysql.com/doc/refman/5.0/en/create-procedure.html>
- [10] **myisampack**
<http://dev.mysql.com/doc/refman/4.1/en/myisampack.html>
- [11] **Problemas asociados a MyISAM**
<http://www.mysqlperformanceblog.com/2006/06/17/using-myisam-in-production/>
- [12] **SQL no estándar**
<http://mysql.conclase.net/curso/index.php?sen=SELECT>
- [13] **Tipo de tabla Memory**
<http://dev.mysql.com/doc/refman/4.1/en/memory-storage-engine.html>
- [14] **Tablas estáticas**
<http://dev.mysql.com/doc/refman/4.1/en/static-format.html>
- [15] **Tipo de tabla MyISAM**
<http://dev.mysql.com/doc/refman/4.1/en/myisam-storage-engine.html>
- [16] **Tipo de tabla MyISAM**
<http://es.wikipedia.org/wiki/MyISAM>
- [17] **Tipo de tabla innnoDB**
<http://dev.mysql.com/doc/refman/4.1/en/innodb-overview.html>
- [18] **MyISAMCHK**
<http://dev.mysql.com/doc/refman/4.1/en/myisamchk.html>
- [19] **Tablas comprimidas**
<http://dev.mysql.com/doc/refman/4.1/en/compressed-format.html>
- [20] **Tablas dinámicas**
<http://dev.mysql.com/doc/refman/4.1/en/dynamic-format.html>

- [21] **Requerimientos de espacio por tipo de dato**
<http://dev.mysql.com/doc/refman/4.1/en/storage-requirements.html>
- [22] **To pack or not to pack - MyISAM Key compression**
<http://www.mysqlperformanceblog.com/2006/05/13/to-pack-or-not-to-pack-myisam-key-compression/>
- [23] **Space Needed for Keys**
<http://dev.mysql.com/doc/refman/4.1/en/key-space.html>
- [24] **Make your Data as Small as Possible**
<http://dev.mysql.com/doc/refman/4.1/en/data-size.html>
- [25] **Mysql V/S PostgreSQL**
http://www.wikivs.com/wiki/MySQL_vs_PostgreSQL
- [26] **PostgreSQL**
<http://en.wikipedia.org/wiki/PostgreSQL>
- [27] **SHOW TABLE**
<http://dev.mysql.com/doc/refman/4.1/en/show-table-status.html>
- [28] **SHOW COLUMNS**
<http://dev.mysql.com/doc/refman/4.1/en/show-columns.html>
- [29] **Claves subrogada**
<http://www.dbmsmag.com/9805d05.html>
- [30] **INSERT**
<http://dev.mysql.com/doc/refman/4.1/en/insert.html>
- [31] **REPLACE**
<http://dev.mysql.com/doc/refman/4.1/en/replace.html>
- [32] **INSERT ON DUPLICATE**
<http://dev.mysql.com/doc/refman/4.1/en/insert-on-duplicate.html>
- [33] **LIMIT**
<http://dev.mysql.com/doc/refman/4.1/en/limit-optimization.html>
- [34] **LOCK TABLE**
<http://dev.mysql.com/doc/refman/4.1/en/lock-tables.html>

10.7 Paper's:

- [1] "Towards On Line Analytical Mining in Large Databases", Jiawei Han (<http://www.sigmod.org/sigmod/record/issues/9803/han.ps.gz?searchterm=olap>).

[2] **“A Functional Model for Data Analysis”**, Nicolas Spyrtos (<http://olapmapper.sourceforge.net/pdf/paper.pdf>).

[3] **“Providing OLAP to User-Analysts: An TI Mandate”**, E.F. Codd, S.B. Codd y C.T. Salley (<http://www.google.com/url?sa=t&ct=res&cd=2&url=http%3A%2F%2Fesgh.pl%2Fniezbednik%2Fplik.php%3Fid%3D4882%26pid%3D460&ei=QIZESM77GJSeYsa7b8E&usg=AFQjCNF-huzed49oe5ThUzBx9608UXkoVA&sig2=gk0RciOI6jZ38okPX4t0kw>).

[4] **“Microsoft SQL Server 2008 Reporting Services in SQL Server 2008”**, Ann Weber, Graeme Malcolm (<http://download.microsoft.com/download/a/c/d/acd8e043-d69b-4f09-bc9e-4168b65aaa71/RSinSQL2008.doc>).

[5] **“Microsoft SQL Server 2005 Analysis Services – Improve your OLAP Environment With Microsoft and Teradata”**, Rupal Shah y Richard Tkachuk (<http://www.microsoft.com/downloadS/thankyou.aspx?familyId=8f38fafb-a7c3-4cc1-a059-7a7a732058a2&displayLang=en>).

[6] **“Microsoft SQL Server 2005 Planning for Scalability and Performance with Reporting Services”**, John Miller, Anne Janzer, Brian Welcker (http://download.microsoft.com/download/4/7/a/47a548b9-249e-484c-abd7-29f31282b04d/RS_Scalability.doc).

[7] **“Oracle Business Intelligence Suite Enterprise Edition Plus – Technical Overview”**, Noviembre 2007, ORACLE.

[8] **Multidimensional Index Structures in Relational Databases**, CHRISTIAN BÖHM, STEFAN BERCHTOLD, HANS-PETER KRIEGEL, URS MICHEL

[9] **A Model Driven Modernization Approach for Automatically Deriving Multidimensional Models in Data Warehouses**, Jose-Norberto Mazon, Juan Trujillo

[10] **Automating Data Warehouse Conceptual schema Design and Evaluation**, Cassandra Phipps, Karen C. Davis

[11] **Towards the Automation of Data Warehouse Logical Design: a Rule-Based Approach**, Verónica Peralta, Alvaro Illarze, Raúl Ruggia

[12] **On the Applicability of Rules to Automate Data Warehouse Logical Design**, Verónica Peralta, Alvaro Illarze, Raúl Ruggia

10.8 Sitios Web:

[1] **Fact constellation schema**
<http://www.datawarehouse4u.info/Data-warehouse-schema-architecture-fact-constellation-schema.html>

[2] **Test Ji-Cuadrado**
<http://www.fisterra.com/mbe/investiga/chi/chi.asp>

- [3] **Minería de datos**
<http://74.125.45.104/search?q=cache:xpO-fgPdPesJ:www.cs.buap.mx/~bbeltran/NotasMD.doc+heurísticas+rolap&hl=es&ct=clnk&cd=1&client=opera>
- [4] **Api's**
<http://www.gotapi.com/mysql>
- [5] **Gráficos flash**
<http://teethgrinder.co.uk/open-flash-chart/>
- [6] **Imagen como link**
https://developer.mozilla.org/en/The_data_URL_scheme
- [7] **RFC 2397, The "data" URL scheme**
<http://tools.ietf.org/html/rfc2397>
- [8] **Ejemplo SpreadsheetML**
http://www.brainbell.com/tutorials/ms-office/excel/Create_Spreadsheets_Using_SpreadsheetML.htm
- [9] **Definición de SpreadsheetML**
<http://www.spreadsheetml.com/>
- [10] **Using HTML in E-mail**
<http://people.dsv.su.se/~jpalme/ietf/mhtml.html>
- [11] **Definición del estándar MHTML**
<http://es.wikipedia.org/wiki/MHTML>
- [12] **MIME Encapsulation of Aggregate Documents, such as HTML (MHTML)**
<http://tools.ietf.org/html/rfc2557>
- [13] **Open XML Deep Dive**
<http://blogs.msdn.com/dmahugh>
- [14] **Open XML**
<http://www.ecma-international.org/publications/standards/Ecma-376.htm>
- [15] **Microsoft Office XML formats**
http://en.wikipedia.org/wiki/Microsoft_Office_XML_formats
- [16] **WordML**
<http://www.informit.com/guides/content.aspx?g=xml&seqNum=176>
- [17] **base64 Decode a Picture in XML**
<http://blogs.msdn.com/kaevans/archive/2005/03/25/402426.aspx>

[18] **OpenXML_White_Paper**

[http://www.ecma-](http://www.ecma-international.org/news/TC45_current_work/OpenXML_White_Paper_Spanish.pdf)

[international.org/news/TC45_current_work/OpenXML_White_Paper_Spanish.pdf](http://www.ecma-international.org/news/TC45_current_work/OpenXML_White_Paper_Spanish.pdf)

[19] **Ecma Office Open XML File Formats Standard**

http://www.ecma-international.org/news/TC45_current_work/TC45_available_docs.htm

[20] **“Apuntes de Ficheros y Bases de Datos”**, María Mercedes Marqués Andrés
(<http://www3.uji.es/~mmarques/f47/apun/apun.html>).

11 Anexos

11.1 Entrevistas

11.1.1 Entrevista realizada a Jefe Área de Info-tecnologías (ADI)

A continuación se presentan las respuestas de la entrevista realizada al Jefe del Área de Info-tecnologías (ADI):

1. ¿Dónde es posible conseguir más información, indicadores o estadísticos?

R: U-cursos, SIMBAD, U-Curriculum, Casa central (SISPER, GUIA CURRICULUM).

2. Comente cuál es su opinión con respecto a los siguientes aspectos de un sistema de apoyo a la generación de indicadores y estadísticos: Operacionales, utilización de recursos, tiempos de respuestas validos, seguridad y correcta operación.

R: Tiempos de respuestas validos para una persona, control de permisos, buena utilización de los recursos.

3. ¿Cuáles son las estadísticas o indicadores frecuentes?

R: Listas de personas con filtros, como por ejemplo mechones, alumnos por carrera, etc.

4. ¿Cuáles son las estadísticas o indicadores más costosos?

R: Las que involucran más datos, por ejemplo, los que involucran los cursos y notas.

5. ¿A qué cree que se deben sus costos?

R: Su costo se debe al volumen de datos

6. Mencione todo los actores o entidades que usted conozca que hayan solicitado en alguna oportunidad indicadores o estadísticos de cualquier índole.

R: Directores de departamentos y de la Escuela (Patricio Poblete), Alfredo Lucas, Subdirectores (Julio Salas, Rocío Duque), etc.

7. ¿Quiénes son los que frecuentemente lo hacen?

R: Director de la Escuela (Patricio Poblete)

8. ¿Qué entidad o actor conoce que haya generado estadísticos o indicadores?

R: ADI, Susana Cerro, Área informática, CEC.

9. ¿Cuáles son las dificultades a las cuales se enfrentan para conseguir las estadísticas e indicadores necesarios?

R: Solicitados a última hora, solicitar vistas cuando son bases de datos de diferentes fuentes.

10. ¿Qué se ha hecho para solucionar estos problemas?

R: SIESTA

11. ¿Cómo cree que se puede solucionar?

R: SIESTA, mejorando la burocracia del STI de casa central.

12. ¿Cómo evaluaría la situación actual?

R: Suficiente.

13. ¿Cuál debería ser el problema a solucionar?

R: Eliminar los cuellos de botella y reducir el número de funcionarios involucrados. Sería bueno contar con un canal formal de petición de reportes y estadísticas.

14. ¿Cree que un sistema de generación de reportes y estadísticos automatizado podría eliminar las dificultades actuales?

R: Sí.

15. ¿A quién debería orientarse un sistema como este?

R: Usuario final.

16. ¿Cuáles son sus características? Comente con respecto al sistema operativo, motores de bases de datos, tamaños de bases de datos, relevancia de la información contenida en ellos, actores y entidades que las consultan, nombres de las bases de datos, fecha de inicio de su utilización, limitaciones, fortalezas y debilidades.

R: SIMBAD utiliza ORACLE sobre Solaris. La información contenida es ella es crítica y es incuestionable. Responde a las necesidades actuales. Sería bueno integrarlo con Postgrado.

17. ¿Cómo se relacionan todos estos sistemas tecnológicamente? Comente.

R: Se realizan consultas para rescatar información y con PHP se procesa y maneja dicha información.

18. ¿Qué pasa cuando un indicador no es posible de generar con la información contenida en una sola BD?

R: Nunca ha pasado, se armaría un proyecto para responder a la petición o bien no se hace.

19. ¿Cree que es posible desarrollar un sistema como este?

R: Sí, pero debe estar bien acotado.

20. ¿Cree que se justifica desarrollarlo?

R: Sí, eliminar los reportes y formularios de ORACLE.

21. ¿Conoce el nombre de alguna herramienta comercial u open source?

R: Sí, las herramientas de data mining (ASP, SQL Server)

22. ¿Ha utilizado alguna(s) herramienta(s) como esta(s)? ¿Cuál(es) es/son?

R: No. Las herramientas OLAP.

23. ¿Hay intentos por resolver esta problemática? Comente la experiencia.

R: Sí, Julio Villane intentó hacer un sistema de estadísticas de logs. Carolina (No recuerda el nombre) en el año 2001 hizo un módulo de estadística para u-cursos.

24. ¿Cómo calificaría los resultados?

R: Inconclusos como para calificarlos.

25. ¿Cuáles son las fortalezas de SIESTA (Sistema de estadística)?

R: Es un sistema Web.

26. ¿Cuáles son las debilidades de SIESTA?

R: Algunas consultas son lentas a veces, algunas consultas deben revisarse por los JOIN que hacen y el volumen de información involucrada.

27. ¿Cómo definiría el volumen de datos a tratar?

R: SIMBAD: Cuenta con información de los alumnos, aproximadamente 600 personas/año durante 40 años, GUIA CURRICULAR: La información de toda la Universidad de Chile, de mediano tamaño, aproximadamente 20.000 personas entre docentes y alumnos, U-CURRICULUM: De pequeño tamaño, cuenta con información de docentes.

11.1.2 Entrevista realizada a Vice-decano Facultad Ciencias Físicas y Matemáticas

A continuación se presentan las respuestas de la entrevista realizada al Vice – decano de la Facultad de Ciencias Físicas y Matemáticas, un usuario frecuente de información estadística e indicadores:

1. ¿Dónde es posible conseguir más información, indicadores o estadísticos?

R: De los datos de matrícula desde la casa central de la Universidad de Chile y de Bienestar.

2. ¿Cuáles son las estadísticas o indicadores frecuentes?

R: Estadísticas de cohorte, BIA, Prioridades de U-Cursos, las que vinculan el rendimiento dentro de la universidad con el origen, el colegio y notas.

3. Mencione todo los actores o entidades que usted conozca, que hayan solicitado en alguna oportunidad indicadores o estadísticos de cualquier índole.

R: Jefes docentes de cada departamento, directores de departamento, Director de Escuela de Ingeniería y Ciencias, Decano, Vice-decano.

4. ¿Qué entidad o actor conoce que haya generado estadísticos o indicadores?

R: Susana Cerro por petición del Director de la Escuela de Ingeniería y Ciencias.

5. ¿Quiénes son actualmente los encargados de generarlos?

R: El responsable es el Director de la Escuela de Ingeniería y Ciencias (Patricio Poblete)

6. ¿Cuáles son las dificultades a las cuales se enfrentan para conseguir las estadísticas e indicadores necesarios?

R: Todo es hecho ad hoc, y se termina haciendo el trabajo varias veces.

7. ¿Qué se ha hecho para solucionar esto?

R: Solo esto.

8. ¿Cree que se puede solucionar?

R: Sí, con un sistema amigable, rápido, bien pensado, usable y sin parches. Debe cubrir los casos frecuentes, el resto debe hacerlo una persona, como Susana Cerro.

9. ¿Cómo evaluaría la situación actual?

R: Insuficiente.

10. ¿Cómo explicaría las dificultades?

R: Falta una mirada integral y pensar bien una solución.

11. ¿Cuál debería ser el problema a solucionar?

R: Reducir los tiempos. Los procesos deben modernizarse acorde con una universidad moderna.

12. ¿Cree que un sistema de generación de reportes y estadísticos automatizado podría eliminar las dificultades actuales?

R: Sí, eso espero.

13. ¿A quién debería orientarse un sistema como este?

R: Al usuario final y que cuente con un módulo de administración. Podría ser interesante un módulo para alumnos.

14. ¿Cree que es posible desarrollar un sistema como este?

R: Sí

15. ¿Ha utilizado alguna(s) herramienta(s) como esta(s)? ¿Cuál(es) es/son?

R: El ADI ha desarrollado un sistema de estadística

16. ¿Hay intentos por resolver esta problemática? Comente la experiencia.

R: No para pregrado ni postgrado.

17. ¿Cuáles son las limitaciones de los sistemas actuales? En cuanto a lo tecnológico, humano, etc.

R: No se ha cubierto el 100% de los casos y se depende mucho de los administradores.

18. ¿Existe algún indicador o estadístico imposible de generar o generado aproximadamente a la fecha por falta de recursos? ¿De qué índole son?

R: No.

19. ¿Cómo definiría el volumen de datos a tratar?

R: Muchos datos

20. ¿Cuál sería el impacto de un sistema como este?

R: Sería tremendamente útil.

21. ¿Qué esperarías de un sistema como este?

R: Algo como U-cursos

11.1.3 Entrevista realizada a Jefe de Área Desarrollo de Informática de la Escuela de Ingeniería y Ciencias

A continuación se presentan las respuestas de la entrevista realizada al Jefe de Área Desarrollo de Informática, uno de los funcionarios de la Escuela de Ingeniería y Ciencias encargada de generar un número importante de reportes estadísticos e indicadores:

1. ¿Dónde es posible conseguir más información, indicadores o estadísticos?

R: Se puede solicitar las consultas SQL.

2. Comente cuál es su opinión con respecto a los siguientes aspectos de un sistema de apoyo a la generación de indicadores y estadísticos: Operacionales, utilización de recursos, tiempos de respuestas validos, seguridad y correcta operación.

R: Hay consultas pesadas por el volumen de datos sobre los que operan, debe contar con control de permisos.

3. ¿Tiene o sabe dónde conseguir información que permitan conocer cuáles son los tipos de reportes que son necesarios generar?

R: Reportes de ORACLE.

4. ¿Cuáles son las estadísticas o indicadores frecuentes?

R: En cuanto a los Mechones: Promedios, seguimientos de los retirados, filtro de mechones por región o por sexo, mechones que aprueban todas las asignaturas, mechones repitentes, mechones eliminados. En cuanto a los Titulados: Temas de memoria, el mejor titulado, sobre los exámenes de grado.

5. ¿Cuáles son las estadísticas o indicadores más costosos?

R: Los relacionados con los planes de estudio.

6. ¿Cuáles son esos costos?

R: Tiempo y los planes de estudio.

7. ¿A qué cree que se debe sus costos?

R: Mal diseño de traspaso y de la base de datos.

8. Mencione todos los actores o entidades que usted conozca que hayan solicitado en alguna oportunidad indicadores o estadísticos de cualquier índole.

R: Decano, Directores de Estudio, Vice-decano, profesores, Directores, comisión de primer año, etc.

9. ¿Quiénes son los que frecuentemente lo hacen?

R: Secretarios de Estudio.

10. ¿Qué entidad o actor conoce que haya generado estadísticas o indicadores?

R: El ADI.

11. ¿Cuáles son las dificultades a las cuales se enfrentan para conseguir las estadísticas e indicadores necesarios?

R: Urgencia, sobrecarga de trabajo, reglas poco claras para generar las consultas, planes de estudio.

12. ¿Qué se ha hecho para solucionar estos problemas?

R: Nada.

13. ¿Cómo cree que se puede solucionar?

R: Separar magíster de pregrado, planes de estudio.

14. ¿Cómo evaluaría la situación actual?

R: Satisfactoria

15. ¿Cual debería ser el problema a solucionar?

R: Eliminar los cuellos de botellas

16. ¿Cree que un sistema de generación de reportes y estadísticas automatizado podría eliminar las dificultades actuales?

R: Facilitar algunas, pero no todas porque son muy complicadas.

17. ¿A quién debería orientarse un sistema como este?

R: Al usuario final

18. ¿Ha utilizado alguna(s) herramienta(s) como esta(s)? ¿Cuál(es) es/son?

R: ORACLE

19. ¿Hay intentos por resolver este problema? Comente la experiencia

R: Si, algunos memoristas.

20. ¿Existe algún indicador o estadístico imposible de generar o generado aproximadamente a la fecha por falta de recursos? ¿De qué índole son?

R: Todo lo pedido se ha podido generar.

21. ¿Cómo definiría el volumen de datos a tratar?

R: Son aproximadamente 1.5 millones de cursos inscritos, 13 carreras, 40.000 personas y se dictan 500 cursos por semestre.

22. ¿Cuál sería el impacto de un sistema como este?

R: Bueno.

23. ¿Qué esperaría de un sistema como este?

R: Que funcionará para los casos más generales.

12.1.4 Entrevista realizada a Gerente de Proyectos del Área de Info-tecnologías (ADI)

A continuación se presentan las respuestas de la entrevista realizada al Gerente de Proyectos del Área de Info-tecnologías (ADI):

1. ¿Dónde es posible conseguir más información, indicadores o estadísticos?

R: En casa central: GUIA, SISPER y AUGE.

2. Comente cuál es su opinión con respecto a los siguientes aspectos de un sistema de apoyo a la generación de indicadores y estadísticos: Operacionales, utilización de recursos, tiempos de respuestas validos, seguridad y correcta operación.

R: Tiempos de respuesta de aproximadamente 5 segundos o podría ocasionarle desesperación al usuario, debe haber control y permisos de acceso, sería bueno controlar los tiempos y la información que se entrega.

3. ¿Tiene o sabe dónde conseguir información que permitan conocer cuáles son los tipos de reportes que son necesarios generar?

R: Sistema de estadística (SIESTA) y sistema de mechones de ADI

4. ¿Cuáles son las estadísticas o indicadores frecuentes?

R: Filtros sobre mechones

5. ¿Cuáles son las estadísticas o indicadores más costosos?

R: Recuento de UD'S.

6. ¿Cuáles son esos costos?

R: Horas hombre.

7. ¿A qué cree que se debe sus costos?

R: Falta de información y desconocimiento.

8. Menciones todos los actores o entidades que usted conozca que hayan solicitado en alguna oportunidad indicadores o estadísticos de cualquier índole.

R: Autoridades, Decano, Vice-decano, autoridades de los departamentos.

9. ¿Quiénes son los que frecuentemente lo hacen?

R: Subdirector de asuntos docentes y Vice-decano.

10. ¿Qué entidad o actor conoce que haya generado estadísticas o indicadores?

R: ADI y Susana Cerro.

11. ¿Cuáles son las dificultades a las cuales se enfrentan para conseguir las estadísticas e indicadores necesarios?

R: Urgencia, poco tiempo y burocracia.

12. ¿Qué se ha hecho para solucionar estos problemas?

R: En el ADI contamos con una wiki para compartir la información, especialmente las consultas.

13. ¿Cómo cree que se puede solucionar?

R: Con algún sistema existente o por desarrollar y mejorar la comunicación.

14. ¿Cómo evaluaría la situación actual?

R: Caótica

15. ¿Cual debería ser el problema a solucionar?

R: Eliminar los cuellos de botellas

16. ¿Cree que un sistema de generación de reportes y estadísticas automatizado podría eliminar las dificultades actuales?

R: No el proceso, si la información.

17. ¿A quién debería orientarse un sistema como este?

R: Al usuario final

18. ¿Ha utilizado alguna(s) herramientas(s) como esta(s)? ¿Cuál(es) es/son?

R: SPSS, Ebios; pero web no hay o no conozco.

19. ¿Hay intentos por resolver este problema? Comente la experiencia

R: SIESTA y sistema de mechones desarrollados por el ADI.

20. ¿Existe algún indicador o estadístico imposible de generar o generado aproximadamente a la fecha por falta de recursos? ¿De qué índole son?

R: No conozco ninguna imposible.

21. ¿Cómo definiría el volumen de datos a tratar?

R: Muchos datos personales. Los cursos inscritos desde el año 1968 aproximadamente.

22. ¿Cuál sería el impacto de un sistema como este?

R: Reducción de horas hombre y tiempos.

23. ¿Qué esperaría de un sistema como este?

R: Que resuelva los problemas actuales.

11.1.5 Entrevista realizada a Jefe de Bienestar Estudiantil

A continuación se presentan las respuestas de la entrevista realizada al Jefe de Bienestar Estudiantil de la Escuela de Ingeniería y Ciencias:

1. ¿Dónde es posible conseguir más información, indicadores o estadísticos?

R: ADI, Susana Cerro, Mineduc, Programas UChile.

2. ¿Cuales son las estadísticas o indicadores frecuentes?

R: Avance y situación académica.

3. ¿Cuales son los fines u objetivos de estos indicadores?

R: Mantenimiento de beneficios o investigación sobre alumnos

4. ¿Qué entidad o actor conoce que haya generado estadísticas o indicadores?

R: ADI y Susana Cerro.

5. ¿Quiénes son actualmente los encargados de generarlos?

R: Ingrid Sepúlveda y Susana Cerro.

6. ¿Cuáles son las dificultades a las cuales se enfrentan para conseguir las estadísticas e indicadores?

R: Sistemas aislados entre los del ADI y Susana Cerro

7. ¿Qué se ha hecho para solucionar estos problemas?

R: Reuniones para intentar solucionar el problema.

8. ¿Cómo cree que se puede solucionar?

R: Con voluntad

9. ¿Cómo explicaría las dificultades?

R: No ha habido comunicación entre los departamentos

10. ¿Cuál debería ser el problema a resolver?

R: Prestar más atención, falta gente que pueda responder a los problemas.

11. ¿Cree que un sistema de generación de reportes y estadísticos automatizado podría eliminar las dificultades actuales?

R: Sí

12. ¿A quien debería orientarse un sistema como este?

R: Al usuario final y a un administrador.

13. ¿Qué pasa cuando un indicador no es posible de generar con la información contenida en una sola BD?

R: Se hace a mano, se tiene confianza en los resultados.

14. ¿Ha utilizado alguna(s) herramienta(s) como esta(s)? ¿Cuál(es) es/son?

R: No

15. ¿Hay intentos por resolver esta problemática? Comente la experiencia

R: Sí, al parecer en seminario de diseño.

16. ¿Cómo calificaría los resultados?

R: El ADI ha sido exitoso. Nunca supe el resultado del trabajo de los alumnos.

17. ¿Cuáles son las limitaciones de los sistemas actuales? En cuanto a lo tecnológico, humano, etc.

R: Falta información sobre el ingreso de los datos, se borran los datos cuando hay errores, filtrar columnas. Me gustaría hacer nuevos requerimientos sobre historial de la información.

18. ¿Cómo definiría el volumen de datos a tratar?

R: Gran cantidad de información. Se cuenta con los datos personales de los profesores y alumnos desde el año 1986 aproximadamente.

19. ¿Cuál sería el impacto de un sistema como este?

R: Reducir los tiempos y facilitar la gestión

11.2 Librerías

11.2.1 Open Flash Chart

La función programada para utilizar la librería y generar los gráficos es la siguiente:

```
1 <<?php
2 function graficar($datos, $tipo = 'barras', $titulo='', $label_y='', $label_x='', $width = '50%', $height = 450 )
3 {
4     $colores = array( '80a033', '50b432', 'ed561b', 'edef00', '24cbe5', '64e572', '505050', '404040', 'd01f3c', 'C79810', '356aa0', 'c0c0c0' );
5     $g = new graph();
6
7     if($tipo=='torta'){
8         $color=$colores[ round( rand(0,count($colores)-1) ) ];
9         $g->pie(60,$color,'(font-size: 10px; color: #404040:');
10        $x_labels=array_keys( $datos );
11        $g->set_tool_tip( '#x_label#: #val#&' );
12        $g->pie_values( array_values($datos), $x_labels );
13        $g->pie_slice_colours( $colores );
14        $g->set_x_label_style( 10, '#000', 2);
15    }
16    else
17    {
18        if($tipo=='barras'){
19
20            $x_labels = array_keys( $datos );
21            $color=$colores[ round( rand(0,count($colores)-1) ) ];
22            $color2=$colores[ round( rand(0,count($colores)-1) ) ];
23            $bar = new bar_outline( 50, $color, $color2 );
24            $bar->data=array_values($datos);
25            $g->data_sets[] = $bar;
26            $g->set_data( $datos );
27        }
28        else if($tipo=='lineas_juntas')
29        {
30            foreach ( $datos as $serie =>$valores )
31            {
32                $color=$colores[ round( rand(0,count($colores)-1) ) ];
33                $x_labels = array_keys( $valores );
34                $valores_r=array_values($valores);
35                $g->set_data($valores_r);
36                $g->line_dot( 1,1, $color, $serie, 10 );
37            }
38        }
39        else if ($tipo=='barras_juntas')
40        {
41            $x_labels = array_keys( $datos['titulo'] );
42            foreach ( $datos['valor'] as $serie =>$valores )
43            {
```



```

44     $color=$colores[ round( rand(0,count($colores)-1) ) ];
45     $bar = new bar( 50, $color );
46     $bar->key( $serie, 10 );
47     $bar->data = $valores;
48     $g->data_sets[] = $bar;
49     }
50     $datos=$datos['valor'];
51     }
52
53     $max=maximo($datos);
54     $g->set_x_label_style( 10, '#000', 1,-1); //VERTICAL , PASO=-1
55     $g->x_axis_colour( '#00000', '#f0f0f0' );
56     $g->y_axis_colour( '#00000', '#00000' );
57     $g->y_label_steps( 10 );
58     $g->set_tool_tip( '#x_label#: #val#' );
59     $g->set_y_legend( $label_y,10,round( rand(0,count($colores)-1) ));
60     $g->set_x_legend( $label_x,10,round( rand(0,count($colores)-1) ));
61     $g->set_y_max( ceil( $max*1.02 ) );
62
63     }
64     $g->set_x_labels($x_labels);
65     $g->title( $titulo, '{font-size: 10px;}' );
66     $g->bg_colour = '#ffffff';
67     $g->set_inner_background( '#ffffff' );
68
69     $id=rand();
70     $chart = InfoNucleo::getParametro( 'common', 'ruta_graphviz', FALSE ).'/'.$id.'.txt';
71     file_put_contents( $chart, $g->render() );
72     chmod( $chart, 0777 );
73
74     $fb = '<div style="text-align:center">';
75     $fb .= '<object width="'. $width. '" height="'. $height. '" id="chart_'.$id.'" />';
76     $fb .= '<param name="movie" value="?chart='.$id.'&accion=svf&data='.urlencode( '?chart='.$id.'&accion=datos&rand='.rand() ).'" />';
77     $fb .= '<embed src="?chart='.$id.'&accion=svf&data='.urlencode( '?chart='.$id.'&accion=datos&rand='.rand() ).'" width="'. $width. '" height="'. $height. '" /></embed>';
78     $fb .= '</object>';
79     $fb .= '</div>';
80
81     return $fb;
82 }
83 >>

```

11.2.2 Librería ExcelMHTML

La librería es la siguiente:

```

1 <?php
2 class excelMHTML
3 {
4     var $codigo;
5     var $imagenes;
6     var $codigo_tabla;
7
8     function excelMHTML()
9     {
10         $this->imagenesEmbebidas=true;
11         $this->imagenes=array();
12         $this->codigo_tabla=array();
13         $this->codigo[]="From: <Guardado por Microsoft Internet Explorer 7>";
14         $this->codigo[]="Subject: ";
15         $this->codigo[]="Date: Thu, 23 Oct 2008 18:14:41 -0300";
16         $this->codigo[]="MIME-Version: 1.0";
17     }
18     function addTable($tabla)
19     {
20         $this->codigo_tabla[]="<TABLE BORDER=1>";
21         $this->codigo_tabla[]="<TBODY>";
22         foreach ($tabla as $n_file => $columnas)
23         {
24             $this->codigo_tabla[]="<TR>";
25             foreach ($columnas as $n_col => $valor)
26                 $this->codigo_tabla[]="<TD>".$valor."</TD>";
27             $this->codigo_tabla[]="</TR>";
28         }
29         $this->codigo_tabla[]="</TBODY>";
30         $this->codigo_tabla[]="</TABLE>";
31     }
32     function addImage($path)
33     {
34         if(strtolower(substr($path,0,4))=='http') die('solo archivos locales, no URL');
35         $this->imagenes['codigo'][]=base64_encode(file_get_contents($path));
36         $this->imagenes['path'][]=$path;
37         $this->imagenes['datos_imagen'][]=getimagesize($path);
38     }
39     function output($nombre='Estadisticas.xls')
40     {
41
42         if(!empty($this->imagenes))
43     {

```

```

44     $this->codigo[]='Content-Type: multipart/related;';
45     $this->codigo[]='    type="text/html";';
46     $this->codigo[]='    boundary="-----_NextPart_000_0000_01C9353B.383B6910"';
47     $this->codigo[]='X-MimeOLE: Produced By Microsoft MimeOLE V6.00.2900.3350';
48     $this->codigo[]='';
49     $this->codigo[]='This is a multi-part message in MIME format.';
50     $this->codigo[]='';
51     $this->codigo[]='-----_NextPart_000_0000_01C9353B.383B6910';
52 }
53
54 $this->codigo[]='Content-Type: text/html;';
55 $this->codigo[]='    charset="Windows-1252";';
56 if(empty($this->imagenes))
57 {
58     $this->codigo[]='Content-Transfer-Encoding: 7-bit';
59     $this->codigo[]='Content-Location: mhtml:file://ALGO_QUE_NO_EXISTE.html';
60     $this->codigo[]='X-MimeOLE: Produced By Microsoft MimeOLE V6.00.2900.3350';
61     $this->codigo[]='';
62     $this->codigo[]='<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.0 Transitional//EN">';
63     $this->codigo[]='<HTML><HEAD>';
64     $this->codigo[]='<META http-equiv=Content-Type content="text/html; charset=windows-1252">';
65     $this->codigo[]='<META content="MSHTML 6.00.6000.16735" name=GENERATOR></HEAD>';
66     $this->codigo[]='<BODY>';
67 }
68 else
69 {
70     $this->codigo[]='Content-Transfer-Encoding: quoted-printable';
71     $this->codigo[]='Content-Location: file://ALGO_QUE_NO_EXISTE.html';
72     $this->codigo[]='';
73     $this->codigo[]='<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.0 Transitional//EN" ';
74     $this->codigo[]='<"http://www.w3c.org/TR/1999/REC-html401-19991224/loose.dtd">';
75     $this->codigo[]='<HTML><HEAD>';
76     $this->codigo[]='<META http-equiv=3DContent-Type content=3D"text/html; =';
77     $this->codigo[]='< charset=3Dwindows-1252">';
78     $this->codigo[]='<META content=3D"MSHTML 6.00.6000.16735" name=3DGENERATOR></HEAD>';
79     $this->codigo[]='<BODY>';
80 }
81
82
83 $this->codigo=array_merge($this->codigo,$this->codigo_tabla);
84 if(!empty($this->imagenes))
85 {
86     $this->codigo[]='<TABLE>';
87     $this->codigo[]='<TBODY>';
88     $this->codigo[]='<TR><TD></TD></TR>';
89     $this->codigo[]='<TR><TD></TD></TR>';
90     foreach ($this->imagenes['path'] as $i => $path)
91     {
92         if($i%2==0)
93             $this->codigo[]='<TR>';
94
95             $path='file:///'.(substr($path,0,1)!='/'?substr($path,1):$path);
96             $this->codigo[]='<TD><IMG SRC=3D"'. $path. '"></TD>';
97
98         if($i%2==0)
99         {
100             for($i=0;$i<15;$i++)
101                 $this->codigo[]='<TD></TD>';
102         }
103         else
104         {
105             $this->codigo[]='</TR>';
106             for($i=0;$i<31;$i++)
107                 $this->codigo[]='<TR><TD></TD></TR>';
108         }
109     }
110     $this->codigo[]='</TBODY>';
111     $this->codigo[]='</TABLE>';
112 }
113 $this->codigo[count($this->codigo)-1].="\\n</BODY></HTML>";
114
115 if(!empty($this->imagenes))
116 {
117     foreach ($this->imagenes['path'] as $i => $path)
118     {
119         $datos=$this->imagenes['datos_imagen'][$i];
120         $this->codigo[]='';
121         $this->codigo[]='-----_NextPart_000_0000_01C9353B.383B6910';
122         $this->codigo[]='Content-Type: '.$datos['mime'];
123         $this->codigo[]='Content-Transfer-Encoding: base64';
124         $this->codigo[]='Content-Location: file:///'.(substr($path,0,1)!='/'?substr($path,1):$path);
125         $this->codigo[]='';
126         $this->codigo[]=$this->imagenes['codigo'][$i];
127     }
128     if(!empty($this->imagenes))
129     {
130         $this->codigo[]='';
131         $this->codigo[]='-----_NextPart_000_0000_01C9353B.383B6910--';
132     }
133 }
134 $this->codigo[]='';
135 header('Content-Type: application/vnd.ms-excel; charset=ISO-8859-1 ');
136 header('Content-Disposition: attachment; filename="'.$nombre.'" ');
137 print implode( "\\n", $this->codigo );
138 exit;
139 }
140 }
141 >>

```

11.2.3 Librería WordML

La librería es la siguiente:

```
1 <?php
2 class WordML
3 {
4     var $numberImages;
5     var $source;
6     var $font;
7     var $marginTop;
8     var $marginBottom;
9     var $marginLeft;
10    var $marginRight;
11    var $pageSizeH;
12    var $pageSizeW;
13
14    function setFont($newFont)
15    {
16        $this->font=$newFont;
17    }
18    function setMarginTop($new)
19    {
20        $this->marginTop=$new;
21    }
22    function setMarginBottom($new)
23    {
24        $this->marginBottom=$new;
25    }
26    function setMarginLeft($new)
27    {
28        $this->marginLeft=$new;
29    }
30    function setMarginRight($new)
31    {
32        $this->marginRight=$new;
33    }
34    function setPageSizeH($new)
35    {
36        $this->pageSizeH=$new;
37    }
38    function setPageSizeW($new)
39    {
40        $this->pageSizeW=$new;
41    }
42    function WordML($title='', $author='', $company='')
43    {
44        $this->marginTop=1417;
45        $this->marginBottom=1417;
46        $this->marginLeft=701;
47        $this->marginRight=701;
48        $this->pageSizeH=15840;
49        $this->pageSizeW=12240;
50        $this->font="Times New Roman";
51
52        $this->source=array();
53
54        $this->source[]='<?xml version="1.0" encoding="UTF-8" standalone="yes"?'>';
55        $this->source[]='<?o:application progid="Word.Document"?'>';
56        $this->source[]='<w:wordDocument xmlns:w="http://schemas.microsoft.com/office/word/2003/wordml";
57        $this->source[]='<w:core xmlns:s="http://schemas.microsoft.com/office/word/2003/core";
58        $this->source[]='<w:core xmlns:al="http://schemas.microsoft.com/office/word/2003/core";
59        $this->source[]='<w:auxInt xmlns:st="http://schemas.microsoft.com/office/word/2003/auxInt";
60        $this->source[]='<w:office xmlns:o="urn:schemas-microsoft-com:office:office";
61        $this->source[]='<w:st xmlns:st="http://schemas.microsoft.com/office/word/2003/wordml/sp2" w:acrossPresent="no" w:acrossPresent="no" w:space="preserve";
62        $this->source[]='</w:office>';
63
64        $this->source[]='<w:ignoreElements w:val="http://schemas.microsoft.com/office/word/2003/wordml/sp2"?'>';
65        $this->source[]='</w:ignoreElements>';
66        $this->source[]='<w:documentProperties>';
67        $this->source[]='<w:title>.$title.</w:title>';
68        $this->source[]='<w:author>.$author.</w:author>';
69        $this->source[]='<w:lastAuthor>.$author.</w:lastAuthor>';
70        $this->source[]='<w:revisionId>.$revisionId.</w:revisionId>';
71        $this->source[]='<w:totalTime>.$totalTime.</w:totalTime>';
72        $this->source[]='<w:created>2008-10-21T12:21:00Z.</w:created>';
73        $this->source[]='<w:lastSaved>2008-10-21T12:22:00Z.</w:lastSaved>';
74        $this->source[]='<w:pages>2.</w:pages>';
75        $this->source[]='<w:words>999.</w:words>';
76        $this->source[]='<w:characters>.$characters.</w:characters>';
77        $this->source[]='<w:company>.$company.</w:company>';
78        $this->source[]='<w:lines>.$lines.</w:lines>';
79        $this->source[]='<w:paragraphs>.$paragraphs.</w:paragraphs>';
80        $this->source[]='<w:charactersWithSpaces>62.</w:charactersWithSpaces>';
81        $this->source[]='<w:version>11.0000.</w:version>';
82        $this->source[]='</w:documentProperties>';
83        $this->source[]='<w:font>';
84        $this->source[]='<w:defaultFont w:asci=".$this->font." w:fareast=".$this->font." w:hansi=".$this->font." w:=".$this->font."?'>';
85        $this->source[]='</w:font>';
86        $this->source[]='<w:styles>';
87        $this->source[]='<w:versionOfBuiltInStyleNames w:val="4"?'>';
88        $this->source[]='<w:latentStyles w:defaultState="off" w:latentStyleCount="156"?'>';
89        $this->source[]='<w:style w:type="paragraph" w:default="on" w:styleId="Normal"?'>';
90        $this->source[]='<w:base w:val="Normal"?'>';
91        $this->source[]='<w:font w:val=".$this->font."?'>';
92        $this->source[]='<w:lang w:val="ES" w:fareast="ES" w:bidirectional="AR-SA"?'>';
93        $this->source[]='<w:rtl?'>';
94        $this->source[]='<w:style w:type="character" w:default="on" w:styleId="Fuente de párrafo predeter."?'>';
95        $this->source[]='<w:base w:val="Default Paragraph Font"?'>';
96        $this->source[]='<w:ulName w:val="Fuente de párrafo predeter."?'>';
97        $this->source[]='<w:smallHidden?'>';
98        $this->source[]='</w:style>';
99        $this->source[]='<w:style w:type="table" w:default="on" w:styleId="Tabla normal"?'>';
100    $this->source[]='<w:base w:val="Normal Table"?'>';
101    $this->source[]='<w:ulName w:val="Tabla normal"?'>';
102    $this->source[]='<w:smallHidden?'>';
103    $this->source[]='<w:tblPr?'>';
104    $this->source[]='<w:tblInd w:val="0" w:type="dxa"?'>';
105    $this->source[]='<w:tblMar?'>';
106    $this->source[]='<w:tblTop w:val="0" w:type="dxa"?'>';
107    $this->source[]='<w:tblLeft w:val="108" w:type="dxa"?'>';
108    $this->source[]='<w:tblBottom w:val="0" w:type="dxa"?'>';
109    $this->source[]='<w:tblRight w:val="108" w:type="dxa"?'>';
110    $this->source[]='</w:tblCellMar>';
111    $this->source[]='</w:tblPr>';
112    $this->source[]='</w:tblInd>';
113    $this->source[]='</w:tblMar>';
114    $this->source[]='</w:tblTop>';
115    $this->source[]='</w:tblLeft>';
116    $this->source[]='</w:tblBottom>';
117    $this->source[]='</w:tblRight>';
118    $this->source[]='</w:tblCellMar>';
119    $this->source[]='</w:tblPr>';
120    $this->source[]='</w:tblInd>';
121    $this->source[]='</w:tblMar>';
122    $this->source[]='<w:style w:type="list" w:default="on" w:styleId="Sin lista"?'>';
123    $this->source[]='<w:base w:val="No List"?'>';
124    $this->source[]='<w:ulName w:val="Sin lista"?'>';
125    $this->source[]='<w:smallHidden?'>';
126    $this->source[]='</w:style>';
127    $this->source[]='<w:style w:type="table" w:styleId="Tabla con cuadrícula"?'>';
128    $this->source[]='<w:base w:val="Table Grid"?'>';
129    $this->source[]='<w:ulName w:val="Tabla con cuadrícula"?'>';
```

```

130 $this->source[]=<w:basedOn w:val="TablaNormal"/>;
131 $this->source[]=<w:rsid w:val="00767437"/>;
132 $this->source[]=<w:rPr>;
133 $this->source[]=<w:font w:val="" $this->font."/>;
134 $this->source[]=<w:rPr>;
135 $this->source[]=<w:tblPr>;
136 $this->source[]=<w:tblInd w:w="0" w:type="dxa"/>;
137 $this->source[]=<w:tblBorders>;
138 $this->source[]=<w:top w:val="single" w:sz="4" w:bwidwth="10" w:space="0" w:color="auto"/>;
139 $this->source[]=<w:left w:val="single" w:sz="4" w:bwidwth="10" w:space="0" w:color="auto"/>;
140 $this->source[]=<w:bottom w:val="single" w:sz="4" w:bwidwth="10" w:space="0" w:color="auto"/>;
141 $this->source[]=<w:right w:val="single" w:sz="4" w:bwidwth="10" w:space="0" w:color="auto"/>;
142 $this->source[]=<w:insideH w:val="single" w:sz="4" w:bwidwth="10" w:space="0" w:color="auto"/>;
143 $this->source[]=<w:insideV w:val="single" w:sz="4" w:bwidwth="10" w:space="0" w:color="auto"/>;
144 $this->source[]=<w:tblBorders>;
145 $this->source[]=<w:tblCellMar>;
146 $this->source[]=<w:top w:w="0" w:type="dxa"/>;
147 $this->source[]=<w:left w:w="10" w:type="dxa"/>;
148 $this->source[]=<w:bottom w:w="0" w:type="dxa"/>;
149 $this->source[]=<w:right w:w="10" w:type="dxa"/>;
150 $this->source[]=<w:tblCellMar>;
151 $this->source[]=<w:tblPr>;
152 $this->source[]=<w:styles>;
153 $this->source[]=<w:styles>;
154 $this->source[]=<w:docPr>;
155 $this->source[]=<w:view w:val="print"/>;
156 $this->source[]=<w:zoom w:percent="100"/>;
157 $this->source[]=<w:doNotEmbedSystemFonts/>;
158 $this->source[]=<w:proofState w:spelling="clean" w:grammar="clean"/>;
159 $this->source[]=<w:savedTemplate w:val=""/>;
160 $this->source[]=<w:defaultTabStop w:val="708"/>;
161 $this->source[]=<w:hyphenationZone w:val="425"/>;
162 $this->source[]=<w:characterSpacingControl w:val="DontCompress"/>;
163 $this->source[]=<w:optimizeForScreen/>;
164 $this->source[]=<w:validateAgainstSchema/>;
165 $this->source[]=<w:saveInvalidXML w:val="off"/>;
166 $this->source[]=<w:ignoreMixedContent w:val="off"/>;
167 $this->source[]=<w:alwaysShowPlaceholderText w:val="off"/>;
168 $this->source[]=<w:compact>;
169 $this->source[]=<w:appTables/>;
170 $this->source[]=<w:smartGridCell/>;
171 $this->source[]=<w:wrapTextInTable/>;
172 $this->source[]=<w:useAsianBreakRules/>;
173 $this->source[]=<w:useWord2002TableStyleRules/>;
174 $this->source[]=<w:compact>;
175 $this->source[]=<w:rsids>;
176 $this->source[]=<w:rsidRoot w:val="00705921"/>;
177 $this->source[]=<w:rsid w:val="002B1AD2"/>;
178 $this->source[]=<w:rsid w:val="002D4001"/>;
179 $this->source[]=<w:rsid w:val="00314FBB"/>;
180 $this->source[]=<w:rsid w:val="0052762C"/>;
181 $this->source[]=<w:rsid w:val="00705921"/>;
182 $this->source[]=<w:rsids>;
183 $this->source[]=<w:docPr>;
184 $this->source[]=<w:body>;
185 $this->source[]=<w:sect>;
186 }
187 function pixelsPoint($pixels)
188 {
189     $points = 0.75 * floatval($pixels);
190     return number_format($points,2);
191 }
192 function output($name='', $path='')
193 {
194     $name = $name != '' ? $name : basename($_SERVER['PHP_SELF'], '.php') . '.xml';
195
196     $this->source[]=<w:sectPr>;
197     $this->source[]=<w:pgSz w:w="" $this->pageSizeW." w:h="" $this->pageSizeH."/>;
198     $this->source[]=<w:pgMar w:top="" $this->marginTop." w:right="" $this->marginRight." w:bottom="" $this->marginBottom." w:left="" $this->marginLeft."
199     eader="708" w:footer="708" w:gutter="0"/>;
200     $this->source[]=<w:cols w:space="374"/>;
201     $this->source[]=<w:docGrid w:line-pitch="360"/>;
202     $this->source[]=<w:sectPr>;
203     $this->source[]=<w:sect>;
204     $this->source[]=<w:wordDocument>;
205
206
207
208     if($path == ''){
209         header("Content-Type: application/msword; charset=ISO-8859-1");
210         header("Content-Disposition: attachment; filename=\"$name\"");
211         echo implode("\n",$this->source);
212     }
213     else
214     {
215         if(substr($path,-1) <> DIRECTORY_SEPARATOR) $path = $path."/";
216         file_put_contents($path . $name, $this->source);
217     }
218 }
219
220 * void addImage(string $imagePath, int $width, int $height, string $title = ''){
221 * @param $table_r: Arreglo de la forma $table_r[$fila0][$columna0],$table_r[$fila0][$columna2]...
222 * /
223 function addTable($table_r,$size=12,$align='center',$negrita=false)
224 {
225     $this->source[]=<w:tbl>;
226     $this->source[]=<w:tblPr>;
227     $this->source[]=<w:tblStyle w:val="TablaConcuadracula"/>;
228     $this->source[]=<w:tblW w:w="5000" w:type="pct"/>;
229     $this->source[]=<w:jc w:val="" $align."/>;
230     $this->source[]=<w:tblLook w:val="01E0"/>;
231     $this->source[]=<w:tblPr>;
232     $this->source[]=<w:tblGrid>;
233
234     $i=0;
235     foreach ($table_r as $fila_id => $columnas)
236     {
237         if($i++=0)
238         {
239             for($j=0;$j<count($table_r);$j++)
240                 $this->source[]=<w:gridCol w:w="" . (5000/count($columnas))."/>;
241                 $this->source[]=<w:tblGrid>;
242         }
243         $this->source[]=<w:tr>;
244         foreach ($columnas as $columnas_id => $valor)
245         {
246             $this->source[]=<w:tc>;
247             $this->source[]=<w:tblPr><w:tblW w:w="2952" w:type="dxa"/><w:tblPr>;
248             if($valor='') $valor=' ';
249             $valor=$this->replace($valor);
250             $this->source[]=$this->newText($valor,$size,$align,$negrita);
251             $this->source[]=<w:tc>;
252         }
253         $this->source[]=<w:tr>;
254     }
255     $this->source[]=<w:tbl>;

```

```

256  /**
257  * void addImage(string $imagePath, int $width, int $height, string $title = '') {
258  * @param $path: Ex /tmp/image
259  * @param $scale: Ex 0.8=80%
260  */
261  function addImage($path,$align='center',$scale=0.8)
262  {
263      $this->numberImages++;
264      $this->source[]='<w:p>';
265      $this->source[]='<w:pPr>';
266      $this->source[]='<w:jc w:val="" $align."/>';
267      $this->source[]='<w:wpPr>';
268      $this->source[]='<w:d>';
269      $this->source[]='<w:pict>';
270
271      $datos_img=getimagesize($path);
272      $width=$scale*$datos_img[0];
273      $height=$scale*$datos_img[1];
274
275      if($this->numberImages==1)
276      {
277          $this->source[]='<v:shapetype id="x0000_t75" coords="21600,21600" o:spt="75" o:preferrelative="t" path="m@48514@11898118985xe filled="f" stroked="f">';
278          $this->source[]='<v:stroke jointstyle="miter"/>';
279          $this->source[]='<v:formulas>';
280          $this->source[]='<v:f eqn="if lineDrawn pixelLineWidth 0"/>';
281          $this->source[]='<v:f eqn="sum 0 1 0"/>';
282          $this->source[]='<v:f eqn="sum 0 0 01"/>';
283          $this->source[]='<v:f eqn="prod 02 1 2"/>';
284          $this->source[]='<v:f eqn="prod 03 21600 pixelWidth"/>';
285          $this->source[]='<v:f eqn="prod 03 21600 pixelHeight"/>';
286          $this->source[]='<v:f eqn="sum 0 0 1"/>';
287          $this->source[]='<v:f eqn="prod 06 1 2"/>';
288          $this->source[]='<v:f eqn="prod 07 21600 pixelWidth"/>';
289          $this->source[]='<v:f eqn="prod 08 21600 0"/>';
290          $this->source[]='<v:f eqn="prod 07 21600 pixelHeight"/>';
291          $this->source[]='<v:f eqn="sum 010 21600 0"/>';
292          $this->source[]='<v:formulas>';
293          $this->source[]='<v:path o:extrusionok="f" gradientshapeok="t" o:connecttype="rect"/>';
294          $this->source[]='<o:lock w:ext="edit" aspectratio="t"/>';
295          $this->source[]='</v:shapetype>';
296      }
297      $this->source[]='<w:binData w:name="wordml://02000001.$this->numberImages.'.jpg">';
298      $this->source[]='base64_encode(file_get_contents($path)).</w:binData>';
299      $this->source[]='<v:shape id="x0000_i102'.(4+$this->numberImages).' type="#x0000_t75" style="width:'.$this->pixelsPoint($width).'pt;height:'.$this->pixelsPoint($height).'pt">';
300      $this->source[]='<v:imagedata src="wordml://02000001.$this->numberImages.'.jpg" o:title="accessibilityIssues."/>';
301      $this->source[]='</v:shape>';
302      $this->source[]='</w:pict>';
303      $this->source[]='</w:d>';
304      $this->source[]='</w:pPr>';
305      $this->source[]='</w:p>';
306  }
307  function newText($texto,$size=28,$align='center',$negrita=false,$cursiva=false)
308  {
309      $size*=2;
310      $this->source[]='<w:p w:rsidR="00705921" w:rsidPr="0052762C" w:rsidDefault="002B1AD2">';
311      $this->source[]='<w:pPr>';
312      $this->source[]='<w:framePr w:space="141" w:wrap="around" w:anchor="text" w:anchor="text" w:x-align="center" w:y="1"/>';
313      $this->source[]='<w:suppressOverlap/>';
314      $this->source[]='<w:jc w:val="" $align."/>';
315      $this->source[]='<w:pPr>';
316      $this->source[]='<w:sz w:val="" $size."/>';
317      $this->source[]='<w:sz-cs w:val="" $size."/>';
318      $this->source[]='<w:rPr>';
319      $this->source[]='<w:pPr>';
320      $this->source[]='<w:r w:rsidRPr="0052762C">';
321      $this->source[]='<w:rPr>';
322      $this->source[]='<w:rFonts w:ascii="" $this->font.' w:h-ansi="" $this->font.' w:cs="" $this->font.'"/>';
323      $this->source[]='<w:kern w:val="12"/>';
324      $this->source[]='<w:sz w:val="" $size."/>';
325      $this->source[]='<w:sz-cs w:val="" $size."/>';
326      $this->source[]='<w:cursiva?<w:l/>';
327      $this->source[]='<w:rPr>';
328      $this->source[]='<w:t>'. $this->replace($texto). '</w:t>';
329      $this->source[]='</w:r>';
330      $this->source[]='</w:pPr>';
331      $this->source[]='</w:p>';
332  }
333  function newPage($align='center')
334  {
335      $this->source[]='<w:p>';
336      $this->source[]='<w:pPr>';
337      $this->source[]='<w:jc w:val="" $align."/>';
338      $this->source[]='<w:pPr>';
339      $this->source[]='<w:d>';
340      $this->source[]='<w:br w:type="page"/>';
341      $this->source[]='</w:d>';
342      $this->source[]='</w:pPr>';
343  }
344  function replace($valor)
345  {
346      return str_replace(array('á','é','í','ó','ú','õ','â','ë','ï','ô','ü','&quot;'),array('a','e','i','o','u',''','A','E','I','O','U',''),$valor);
347  }
348  }

```

11.2.4 Clase Gráfico

La librería es la siguiente:

```
1 <<
2 require_once('InfoNucleo.php');
3 $ruta_fontera = InfoNucleo::getParametro('comon', 'ruta_fontera');
4 define('FF_BOLD', $ruta_fontera.'/');
5 unset($GLOBALS['php_errormsg']);
6
7 require('jgraph/jgraph.php');
8 require('jgraph/jgraph_pie.php');
9 require('jgraph/jgraph_pie3d.php');
10 require('jgraph/jgraph_bar.php');
11 require('jgraph/jgraph_flag.php');
12 require('jgraph/jgraph_line.php');
13
14
15 class Grafico {
16     function graficador($tipo, $datos, $nombre, $tit, $label_x='', $label_y='', $subtit='', $tab='') {
17         switch ($tipo) {
18             case 'barras horizontales' :
19                 $ycount=count($datos['datos']);>50?9*$count($datos['datos']):40*$count($datos['datos']);
20                 $graph=Grafico::graficocartesiano(750,$y,0,mul,$datos['labels'],1);
21                 $plot=Grafico::lploteador($datos['datos']);
22                 if(count($datos['datos'])>50){
23                     $graph->xaxis->SetFont(FF_FONTO,FS_NORMAL,8);
24                 }
25                 else $plot->value->Show();
26                 $graph->Add($plot);
27                 $graph->xaxis->title->Set($label_x);
28                 $graph->yaxis->title->Set($label_y);
29                 $graph->xaxis->title->SetMargin(70);
30                 break;
31             case 'barras verticales' :
32                 //<math>x</math>count($datos['datos'])>30?12*$count($datos['datos']):800;
33                 $graph=Grafico::graficocartesiano(800,500,0,mul,$datos['labels'],1);
34                 $plot=Grafico::lploteador($datos['datos']);
35                 $graph->xaxis->SetFont(FF_ARIATL,FS_NORMAL,8);
36                 if(count($datos['datos'])<=30)
37                 {
38                     $plot->value->SetFont(FF_ARIATL,FS_NORMAL,8);
39                     $plot->value->Show();
40                 }
41                 $graph->xaxis->SetLabelAngle(45);
42                 $graph->Add($plot);
43                 $graph->xaxis->title->Set($label_x);
44                 $graph->yaxis->title->Set($label_y);
45                 $graph->xaxis->title->SetMargin(70);
46                 break;
47             case 'lineas' :
48                 $xmax=-1;
49                 $colores=array('navy','darkgreen','gold','darkorchid4','deeppink2','indianred2','deepskyblue2');
50                 if('is_array($datos['lineas'])&&$datos['lineas']=array());
51                 foreach ($datos['lineas'] as $linea){
52                     foreach ($linea['datos'] as $dato){
53                         if($dato*$max*$max=$dato;
54                     }
55                 }
56                 $i=10;
57                 while (($max/$i)>0.5) {$i*=10;}
58                 $max=floor(10*$max/$i)+(floor(10*$max/$i)%2!=0?1:1)*$i/10;
59                 $graph=Grafico::graficocartesiano(800,500,$max,mul,$datos['labels']);
60                 $c=0;
61                 foreach ($datos['lineas'] as $ldatos){
62                     $plot=Grafico::lploteador($ldatos['datos'],$ldatos['legend'],$colores[$c].'%0.5');
63                     $c++;
64                     $graph->Add($plot);
65                 }
66                 $graph->xaxis->SetFont(FF_ARIATL,FS_NORMAL,8);
67                 $graph->xaxis->SetLabelAngle(45);
68                 $graph->xaxis->title->Set($label_x);
69                 $graph->yaxis->title->Set($label_y);
70                 $graph->xaxis->title->SetMargin(70);
71                 break;
72             case 'torta' :
73                 $graph=Grafico::graficotorta();
74                 $legends=$datos['legends'];$datos['legends']=$datos['labels'];
75                 $spl=Grafico::pie3deador($datos['datos'],$legends,$datos['colores']);
76                 $graph->Add($spl);
77                 if(count($legends)>15)$graph->legend->SetFont(FF_FONTO,FS_NORMAL,8);
78                 $graph->SetAntialiasing();
79                 break;
80             default:Grafico::graficador('barras horizontales', $datos,$tit,$subtit);
81         }
82     }
83     if($tit)
84     {
85         $graph->title->SetFont(FF_FONT1,FS_BOLD,10);
86         $graph->title->Set($tit);
87     }
88     if($subtit) $graph->subtit->Set($subtit);
89     if($tab) $graph->tabtitle->Set($tab);
90     $graph->SetImFormat('jpeg');
91     if($nombre!='')
92     {
93         $graph->Stroke($nombre);
94     }
95     else
96     {
97         $graph->Stroke();
98     }
99 }
100
101 function graficocartesiano($x,$y,$xmax=0,$tab=mul,$xlabels=mul,$grizado=0) {
102     $graph = new Graph($x,$y);
103     $graph->SetBox();
104     if($tab)$graph->tabtitle->Set($tab);
105     $graph->title->SetFont(FF_FONT1,FS_BOLD,10);
106     $graph->title->SetColor('darkblue');
107     $graph->tabtitle->SetColor('navy','white','navy');
108     $graph->tabtitle->SetFont(FF_ARIATL,FS_BOLD,10);
109     $graph->SetFrame(false);
110     $graph->SetScale('textlin',0,$xmax,0,0);
111     $graph->SetMarginColor('white');
112     if ($grizado)
113         $graph->Set90AndMargin(130,20,10,30);
114     else
115         $graph->SetMargin(100,70,40,100);
116
117     $graph->ygrid->SetFill(true, '#666666', '#666666');
118     $graph->ygrid->SetLineStyle('dashed');
119     $graph->xgrid->SetColor('darkgray');
120     $graph->xgrid->Show();
121     $graph->xgrid->SetLineStyle('dashed');
122     $graph->xgrid->SetColor('darkgray');
123     $graph->xaxis->SetTitleLabels($xlabels);
124     $graph->xaxis->SetFont(FF_FONT1,FS_NORMAL,8);
125     $graph->legend->Pos(0.01,0.10);
126     $graph->legend->SetShadow('darkgray80.5');
127     $graph->legend->SetFillColor('#eeeeee');
128     $graph->img->SetAntialiasing();
129     return $graph;
130 }
```

```

128 function graficotorta(){
129     $graph = new PieGraph(800,500,"auto");
130     $graph->SetFrame(false);
131     $graph->title->SetFont(FF_FONT2,FS_BOLD,10);
132     $graph->title->SetColor("darkblue");
133     $graph->legend->Pos(0.05,0.30);
134     $graph->legend->SetFillColor("#f0f0f0");
135     return $graph;
136 }
137
138 function pie3deador($datos,$legends,$colors=null){
139     // IMPORTANTE: la clase tiene un bug que da vuelta los datos si se le entregan colores a mano
140     if($colors || count($datos)<9){$p1 = new PiePlot3d(array_reverse($datos));
141     else $p1 = new PiePlot3d($datos);}
142     $p1->SetStartAngle(45);
143     if ($colors)$p1->SetSliceColors($colors);
144     elseif(count($datos)<9)$p1->SetSliceColors(
145         array('cadetblue3','wheat','darkseagreen2','pink2','lightyellow','lightsteelblue2','lightcyan2','chistle2'));
146
147     $p1->SetCenter(0.37);
148     $p1->SetSize(0.50);
149     $p1->value->SetFont(FF_FONT1,FS_NORMAL,8);
150     if($legends)$p1->SetLegends($legends);
151     return $p1;
152 }
153
154 function lploteador($ydata,$leyend,$color='navy'){
155     if($color=='0.5')$color=array(rand(10,250), rand(10,250),rand(10,250));
156     $lplot = new LinePlot($ydata);
157     $lplot->SetCenter();
158     $lplot->SetColor($color);
159     $lplot->mark->SetType(MARK_SQUARE);
160     $lplot->mark->SetColor($color);//'blue@0.5');
161     $lplot->mark->SetFillColor($color);
162     $lplot->mark->SetSize(6);
163     $lplot->SetLegend($leyend);
164     return $lplot;
165 }
166 function bploteador($datos){
167     $bplot = new BarPlot($datos);
168     if (count($datos)>50){
169         $bplot->SetWidth(0.6);
170         $bplot->SetFillColor('skyblue@0.4');
171     }
172     else {
173         $bplot->SetFillColor('#cb9a2d');
174         $bplot->value->SetFormat('%0f');
175         $bplot->value->SetColor('black');
176     }
177     $bplot->SetColor('black');
178     return $bplot;
179 }
180
181 }
182
183
184
185 ?>

```

11.3 Análisis de opciones PACK_KEYS

En este punto se describen las diferentes pruebas realizadas para determinar y comprobar los tiempos de respuesta y ahorro de espacio en disco de diferentes consultas y estructuras de tablas similares a las utilizadas en este trabajo. Dichas pruebas se realizaron a través de la comparación de los resultados con compresión (PACK_KEYS=1) y no compresión (PACK_KEYS=0) de claves.

11.3.1 Prueba N°1 PACK_KEYS

Se realizó esta prueba sobre una tabla con 20.000 filas y dos columnas enteras, en la cual la primera solo tiene el valor 1 y la segunda los valores enteros entre 1 y 20000 en orden ascendente. La clave primaria es una llave compuesta de las dos columnas. Los resultados fueron los siguientes:

Resultado con PACK_KEYS=1 (Con compresión)

Datos	179,991	Bytes
Índice	248,832	Bytes
Total	428,823	Bytes

Resultado con PACK_KEYS=0 (Sin compresión)

Datos	179,991	Bytes
Índice	248,832	Bytes

Total 428,823 Bytes

11.3.2 Prueba N°2 PACK_KEYS

Se realizó sobre una tabla con 20.000 filas y cuatro columnas enteras, en la cual la primera y segunda columna solo tienen el valor 1, la tercera tiene valores enteros entre el 1 y el 20000 en orden ascendente, la cuarta valores entre el 2 y el 20002 y la quinta columna los valores entre el 3 y el 20003. La clave primaria es una llave compuesta de las cinco columnas. Los resultados fueron los siguientes:

Resultado con PACK_KEYS=1 (Con compresión)

Datos	419,979	Bytes
Índice	295,936	Bytes
Total	715,915	Bytes

Resultado con PACK_KEYS=0 (Sin compresión)

Datos	419,979	Bytes
Índice	508,928	Bytes
Total	928,907	Bytes

11.3.3 Prueba N°3 PACK_KEYS

Similar a la anterior pero con 40000 filas. Los resultados fueron los siguientes:

Resultado con PACK_KEYS=1 (Con compresión)

Datos	839,979	Bytes
Índice	590,848	Bytes
Total	1,397	KB

Resultado con PACK_KEYS=0 (Sin compresión)

Datos	839,979	Bytes
Índice	991	KB
Total	1,811	KB

11.3.4 Prueba N°4 PACK_KEYS

Similar a las dos pruebas anteriores pero con 60000 filas. Los resultados fueron los siguientes.

Resultado con PACK_KEYS=1 (Con compresión)

Datos	1,230	KB
Índice	884,736	Bytes
Total	2,094	KB

Resultado con PACK_KEYS=0 (Sin compresión)

Datos	1,230	KB
Índice	1,488	KB
Total	2,718	KB

11.3.5 Prueba N°5 PACK_KEYS

En esta prueba se realizaron mediciones del espacio utilizado por la tabla utilizada en las tres pruebas anteriores para los siguientes números de filas:

Nº TUPLAS	PACK_KEY=1	PACK_KEY=0	AHORRO ESPACIO
1000000	35,731,456	46,356,480	22.92025624
500000	17,866,752	23,031,808	28.90875745
250000	8,871,936	11,590,656	23.4561357
125000	4,467,712	5,795,840	22.91519435
62500	2,234,368	2,883,584	22.51420455
31250	1,112,064	1,449,984	23.30508475
15625	559,528	722,344	22.53995326
7812	279,743	363,711	23.0864615
3906	140,373	182,357	23.02297142
1953	71,712	93,216	23.06900103
976	36,859	47,099	21.74143825
488	19,443	24,563	20.8443594
244	11,247	13,295	15.40428733
122	6,637	7,661	13.36640125
61	3,308	5,356	38.23749066
30	2,657	2,657	0
15	2,342	2,342	0
7	2,174	2,174	0
3	2,090	2,090	0
1	2,069	2,069	0

Fig 11.1 Medición de espacio utilizado para la opción PACK_KEYS

Como se puede ver en la imagen anterior el ahorro de espacio es de aproximadamente 20%. Los gráficos del espacio utilizado para ambos casos son los siguientes:

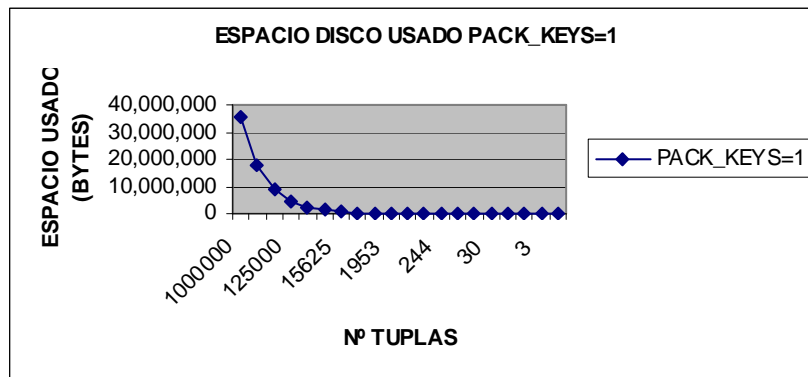


Fig 11.2 Espacio utilizado en disco según número de tuplas con la opción 1

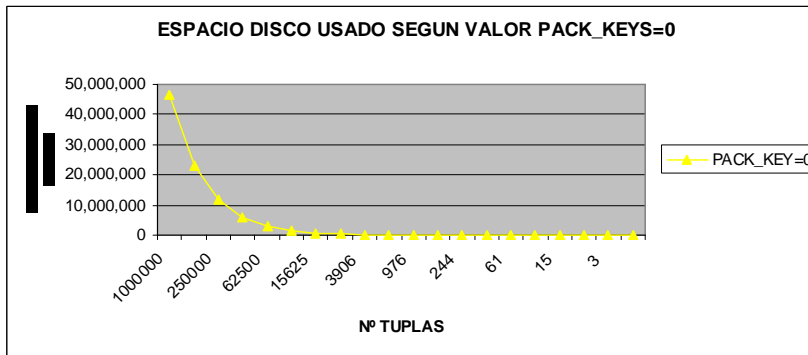


Fig 11.3 Espacio utilizado en disco según número de tuplas con la opción 0

El porcentaje de ahorro de espacio en disco se puede ver en el siguiente gráfico:

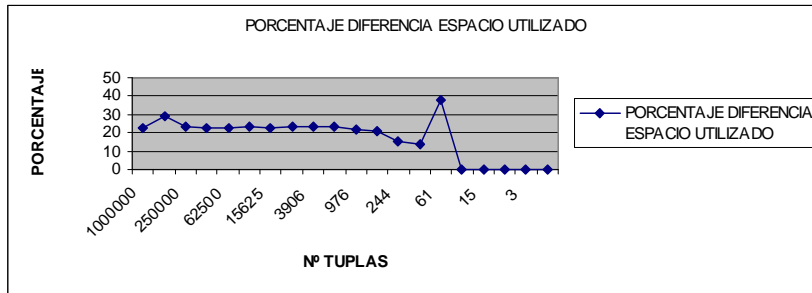


Fig 11.4 Porcentaje de utilización de espacio en disco opción 1 v/s 0

11.3.6 Prueba N°6 PACK_KEYS

Se realizaron pruebas con la tabla anterior para la siguiente consulta:

Select * from TABLA as T, TABLA as T1 Where T.COL3=T.COL4 LIMIT 0,X

Resultado con PACK_KEYS=1 (Con compresión)

LIMIT 0,X	Tiempo respuesta(SEG)
1	0.0092
2	0.0107
4	0.0138
8	0.0206
16	0.0344
32	0.0611
64	0.1165
128	0.2197
256	0.4376
512	0.8699
1024	1.7228
2048	3.4274
4096	6.8628

Resultado con PACK_KEYS=0 (Sin compresión)

LIMIT 0,X	Tiempo respuesta(SEG)
1	0.0122
2	0.0138
4	0.017
8	0.0238
16	0.0369
32	0.0642
64	0.1177
128	0.2253
256	0.4383
512	0.8685
1024	1.7299
2048	3.4387

4096	6.8732
------	---------------

11.3.7 Prueba N°7 PACK_KEYS

Se realizaron pruebas con la tabla anterior para la siguiente consulta:

Select * from TABLA as T, TABLA as T1 Where T.COL=T.COL2 LIMIT 0,X

Resultado con PACK_KEYS=1 (Con compresión)

LIMIT 0,X	Tiempo respuesta(SEG)
1	0.0004
2	0.0003
4	0.0004
8	0.0004
16	0.0004
32	0.0005
64	0.0007
128	0.001
256	0.0017
512	0.0031
1024	0.0052
2048	0.0095
4096	0.0182
1000000	4.4031

Resultado con PACK_KEYS=0 (Sin compresión)

LIMIT 0,X	Tiempo respuesta(SEG)
1	0.0247
2	0.0004
4	0.0004
8	0.0004
16	0.0004
32	0.0005
64	0.0007
128	0.001
256	0.0017
512	0.0194
1024	0.0052
2048	0.0096
4096	0.0183
1000000	4.5961

11.3.8 Prueba N°8 PACK_KEYS

Se realizaron pruebas con la tabla anterior para la siguiente consulta:

Select * from TABLA as T, TABL as T2 where T.COL=T2.COL group by T.COL,T.COL2,T.COL3,T.COL4,T.COL5

Resultado con PACK_KEYS=1 (Con compresión)

LIMIT 0,X	Tiempo respuesta(SEG)
1	0.7997
2	0.6038
4	3.199
8	6.414
16	12.8353
32	25.6684
64	51.6928
128	104.3964

Resultado con PACK_KEYS=0 (Sin compresión)

LIMIT 0,X	Tiempo respuesta(SEG)
1	0.7998
2	1.6074
4	3.2109
8	6.4155
16	12.7902
32	25.6383
64	51.3662
128	102.5957