UNIVERSIDAD NACIONAL AUTÓNOMA DE MÉXICO

FACULTAD DE INGENIERÍA

"ANÁLISIS, DISEÑO Y DESARROLLO DE UN DATA WAREHOUSE AL PERSONAL ACADÉMICO DEL SISTEMA DE INFORMES ACADÉMICOS DE HUMANIDADES PARA EL IISUE"

T E S I S

QUE PARA OBTENER EL TÍTULO DE

INGENIERO EN COMPUTACIÓN

P R E S E N T A

JULIO ADRIAN RESCALVO PÉREZ

MARCOS JAVIER ROMERO VALENCIA

Directora:

Ing. Gabriela Betzabé Lizárraga Ramírez

Ciudad Universitaria

México D.F. 2015

Agradecimientos

A Dios

Por permitirme culminar esta etapa en mi vida y haberme acompañado a lo largo de toda mi formación, reconfortándome en los momentos de desasosiego e incertidumbre y por haberme dado una vida llena de aprendizajes, experiencias y felicidad.

A mis padres Francisco Rescalvo y María del Carmen Pérez

Les doy gracias por su apoyo incondicional en todo momento, por la paciencia y confianza que me han brindado a lo largo de toda mi vida, y sobre todo por haberme dado la oportunidad de tener una excelente educación. Los amo, muchas gracias por todo.

A mi hermano Arturo

Por ser parte importante de mi vida, también por la confianza y cariño que me ha brindado su familia (Yanet, Ivan y Vanessa) y en especial el por representar un ejemplo de desarrollo profesional a seguir.

A Veronica

Gracias por haberme apoyado en todo momento y sobre todo por tu amor incondicional.

A mi directora de tesis Ing. Gabriela Betzabé Lizárraga Ramírez

Por todo el tiempo, paciencia y dedicación que nos otorgó para el desarrollo de esta tesis, y que nos permitió alcanzar este objetivo profesional. Muchas gracias.

A Javier

Te agradezco el haberme acompañado y haber compartido esta experiencia conmigo, tantos momentos agradables y difíciles en fin, pero este es el fruto de nuestro trabajo. Muchas gracias.

A la UNAM, al IISUE y Sinodales

¡Gracias! Por todas las enseñanzas y aportaciones que han dado a mi vida profesional, porque gracias a ustedes logre finalizar este proceso, y no me cansare de retribuirles con mi esfuerzo, conocimiento y dedicación lo mucho que me han otorgado.

Julio Adrian

Agradecimientos

A Dios

Por regalarme la dicha de lograr este proyecto y por poner en mí camino a las personas adecuadas para conseguirlo

A mis padres Ernesto Romero y Santa Valencia

Les agradezco por hacerme una persona de bien, por darme la fuerza necesaria para lograr con éxito esta meta , también por proporcionarme el apoyo y lo necesario para ser un profesional comprometido que es capaz de enfrentar los retos que se le presenten. ¡Los amo!

A mi hermano Ernesto Gabriel

Por la valiosa ayuda y apoyo que me brindaste en todo momento.

A Leonor

Por su apoyo incondicional en toda mi carrera y seguir motivándome cuando más los necesitaba.

A Maribel

Por sus consejos y por su apoyo recibido para la realización de esta tesis.

A mi directora de tesis y Profesora Gabriela Betzabé Lizárraga Ramírez

Por brindarnos su apoyo, tiempo, consejos, paciencia y amistad para la elaboración de este trabajo. ¡Gracias!

A los Sinodales

Gracias por aceptar este proyecto y por las valiosas aportaciones para su mejora.

A Julio

Gracias por haberme apoyado y por realizar este trabajo, fue enriquecedor, te deseo buena suerte para lo que venga en tu vida profesional.

A la UNAM, al IISUE y personal del ISSUE que colaboraron en este trabajo académico

Agradezco su valiosa colaboración al aportarnos gran parte de los elementos para sustentar y definir nuestra tesis.

Marcos Javier

Índice

Introducción

En la actualidad el Data Warehouse es una tecnología que toma cada día mayor importancia, a medida que las organizaciones o instituciones pasan de esquemas de sólo recolección de datos a esquemas de análisis de información, debido a los vertiginosos cambios que sufren en sus entornos externos o internos. Comienza a surgir la necesidad de hacer uso de este tipo de tecnología, ya que su función esencial es el ser la base de un sistema de información gerencial, que deberá cumplir el rol de integrador de información y posteriormente brindar una visión analítica y detallada de tal información con la finalidad de dar soporte a la toma de decisiones. Esta información cuantitativa permitirá obtener ventajas competitivas y mejorar su posición en los entornos o mercados en los que se desarrollan tales organizaciones o instituciones que hagan uso de este tipo de tecnología.

El proceso de Data Warehouse o mejor conocido como Data Warehousing es el encargado de extraer, transformar, consolidar, integrar y centralizar los datos que una organización genera en todos los ámbitos de su actividad diaria (ventas, producción, servicios, etc.) además de información externa relacionada. Permitiendo de esta manera el acceso y la exploración de la información requerida, a través de una amplia gama de posibilidades de análisis multidimensional, con el objetivo final de brindar soporte al proceso de toma de decisiones estratégico, con ayuda de reportes ad-hoc.

El proyecto que se presenta en esta tesis comienza con la elaboración de un análisis, continúa con un diseño y concluye con la creación de un Data Warehouse, cuyo objetivo primordial es el de promover una solución a la falta de sistemas de información gerenciales, que den como resultado información de calidad, cuantitativa y más detallada, brindando un mayor soporte en el proceso de toma de decisiones para la planeación y administración de recursos o simplemente para la exploración de información estadística.

CAPÍTULO 1. Caso de Estudio

En la Universidad Nacional Autónoma de México se encuentra fundado el Instituto de Investigaciones sobre la Universidad y la Educación, cuya función principal es el desarrollo de investigación sobre la educación en los distintos ámbitos académicos, esto mediante el aporte de una serie de actividades e investigaciones sociales e históricas realizadas por un conjunto de personas. Este mismo personal es el principal motor de la institución, ya que realiza y da a conocer nuevos documentos o materiales que son almacenados en una colección de registros dentro de un sistema web informático.

Dicho sistema web tiene por nombre SIAH[1], el cual alberga los datos relacionados con los Académicos que tiene la propia UNAM. Sin embargo, el Instituto de Investigaciones sobre la Universidad y la Educación únicamente tiene acceso a los registros de los Académicos (Investigadores, Profesores y Técnicos Académicos) que tiene a su cargo.

Mediante este conjunto de datos el instituto tiene la tarea de generar reportes que den a conocer gran parte de las actividades e investigaciones elaboradas por los Académicos. En ello esta tesis propone la idea de utilizar la tecnología de Data Warehouse que facilite y mejore la exploración y el análisis de esta información mediante la generación de reportes y al mismo tiempo brinde un soporte para la toma de decisiones.

[1] Sistema de Informes Académicos de Humanidades elaborado por la Coordinación de Humanidades de la UNAM

1.1. Instituto de Investigaciones sobre la Universidad y la Educación (IISUE)

El Instituto de Investigaciones sobre la Universidad y la Educación (IISUE) es una entidad académica que tiene a su cargo un Personal Académico (investigadores, profesores y técnicos académicos) con base a un proyecto principal de investigación pero éste a su vez se subdivide en distintas áreas, las cuales se presentan a continuación:

- **Currículum, formación y vinculación.** Analiza, desde un punto de vista multidisciplinario, el currículum, la formación de sujetos sociales en el espacio educativo institucional, la relación entre la formación y los requerimientos socioeconómicos que se expresan en el mundo del trabajo. [1]

- **Diversidad sociocultural en la educación.** Se ocupa de la interrelación entre sociedad, cultura y educación. Su referente central es la construcción de los sujetos en espacios socialmente complejos y culturalmente diversos en nuestra sociedad. Estudia las tecnologías como ejes transversales del desarrollo social, económico y cultural que contribuyen a la creación de conocimientos, saberes, de nuevas formas de enseñar y aprender. [1]

- **Historia de la educación y la cultura.** Ésta área se enfoca en el análisis histórico de las instituciones, toma en cuenta las manifestaciones sociales, políticas y culturales que conforman el fenómeno educativo e inciden en él. Reúne a especialistas que atienden al estudio de la Hispanoamérica colonial, independiente y contemporánea. [1]

- **Políticas de la educación.** Las investigaciones que realiza examinan las políticas educativas y científico tecnológicas, en los ámbitos institucional, nacional, regional y mundial, a fin de responder a los retos del conocimiento mismo, así como a las demandas que plantea la puesta en práctica de tales políticas. [1]

- **Teoría y pensamiento educativo.** Indaga la construcción conceptual del campo educativo, sus líneas de pensamiento, autores clásicos y contemporáneos, corrientes, enfoques metodológicos y categorías, reflexiona sobre ámbitos como el de la educación ambiental, el de la historia de la educación y el de la formación artística. [1]

1.1.1. Historia del IISUE

La instauración del Instituto de Investigaciones sobre la Universidad y la Educación (IISUE) se hizo en septiembre de 2006, se funda en el reconocimiento de la fortaleza que a lo largo de 30 años fue consolidando el Centro de Estudios sobre la Universidad (CESU[2]) en los temas de su competencia, con el concurso de sus investigadores, técnicos académicos y personal de apoyo. [1]

La estructura y actividades actuales del Instituto son el resultado de un proceso en el que se pueden señalar cuatro etapas descritas en la siguiente tabla:

FECHAS	EVENTOS
1976 Primera etapa	El CESU comenzó centrándose en la custodia del AHUNAM[3] y en la elaboración de algunas cronologías y estudios de prospectiva universitaria. Surgieron sus primeras publicaciones, conformadas básicamente por textos de académicos externos al CESU.

[2] Es una institución que realiza investigación sobre educación, en particular de la educación superior (historia de la universidad, su situación presente y su prospectiva).
[3] Archivo Histórico de la UNAM es el depósito final de la documentación generada por sus distintas entidades

1982	Posteriormente se avanzó en dos direcciones: ➢ Investigación: Investigadores que abordarían el estudio pasado y presente de la universidad y el estudio histórico sobre esta institución. ➢ AHUNAM: Se incrementaron los fondos del AHUNAM, buscando que cada dependencia universitaria le hiciera llegar sus documentos con valor permanente. Además se fomentaron las donaciones de particulares para enriquecer los archivos incorporados, se procedió a la organización de los fondos, a su conservación y a la elaboración de instrumentos de consulta.
1983 Segunda Etapa	Las publicaciones consistieron principalmente en instrumentos descriptivos del Archivo. Se publicaron memorias sobre la historia de la universidad y sobre temas de la UNAM en el presente. El CESU ya contaba con 11 investigadores.
1985 Tercera Etapa	Se efectuaron estudios sobre la universidad con perspectivas históricas y sociológicas. Se amplió la investigación más allá del estricto estudio de la UNAM y de la universidad en general, con estudios sobre la educación nacional desde el punto de vista de diversas disciplinas. El CESU llega a tener 32 investigadores.
1988	Se da el Acuerdo para la protección, uso y conservación del patrimonio histórico de la UNAM.

1996	Se ampliaron las labores del AHUNAM al crearse, para una mejor atención de los materiales bajo su custodia, la Sección del Acervo Gráfico, además del desarrollo y ampliación del Laboratorio de Conservación y Restauración. Se fortaleció el vínculo entre investigación y docencia, mismo que se ha incrementado notablemente hasta llegar a 45 investigadores.
1997 Cuarta Etapa	Se abrió una nueva etapa al expedirse el Acuerdo de Reestructuración de la Estructura Administrativa de la UNAM. Se incorporaron al CESU 21 académicos provenientes del Centro de Investigaciones y Servicios Educativos (CISE); así se elevó a 66 el número de investigadores.
2006	CESU cambió su denominación por la de Instituto de Investigaciones sobre la Universidad y la Educación. Se contaba para entonces con una plantilla académica de 61 investigadores y 39 técnicos académicos de tiempo completo.

Tabla 1.1.1. Las cuatro etapas de evolución del IISUE

Se puede afirmar que el trabajo académico de investigación del IISUE se encuentra hoy en proceso de consolidación; sus líneas de investigación cubren un importante espacio en el terreno del conocimiento y análisis de la educación, en especial de la universidad y de la educación superior. De igual manera, el AHUNAM (Archivo Histórico de la UNAM) profesionaliza su labor y desempeña un papel fundamental en las tareas del resguardo archivístico, tanto dentro de la UNAM como fuera de ella. [1]

1.1.2. Objetivos del IISUE

El IISUE (Instituto de Investigaciones sobre la Universidad y la Educación) tiene definido actualmente dos objetivos principales:

- El desarrollar investigación sobre la propia Universidad y la educación.
- El conservar, organizar y difundir la memoria documental de la UNAM.

1.1.3. Misión del IISUE

La misión es generar conocimiento sobre la Universidad y la educación, en todas sus vertientes, épocas, tendencias, manteniendo un profundo compromiso con los intereses de la sociedad y del país, dentro de un marco de libertad de pensamiento y enseñanza. [2]

1.1.4. Visión del IISUE

Ha sido y deberá ser el IISUE la institución que genere nuevas aportaciones, investigaciones, en todas sus épocas y tendencias, además de fomentar el desarrollo con sentido humanístico, social, económico y tecnológico, con la ayuda de sus investigadores. [2]

1.2. Planteamiento del problema

Dentro del Instituto de Investigaciones sobre la Universidad y la Educación (IISUE) siempre se ha tenido la tarea de elaborar reportes que muestren una visión detallada sobre el desarrollo de actividades e investigaciones elaboradas por el Personal Académico. Dichos reportes tienen que contener gráficas, que de igual forma muestren y detallen esa información a lo largo de periodos anuales, para luego realizar posibles análisis sirviendo de base en la mejora de decisiones con respecto a la administración y planeación de recursos del IISUE.

Además se cuenta con el problema de no poder manipular la Base de Datos (BD) creada por el sistema informático SIAH, por consiguiente tampoco se puede obtener información útil y relevante de esa fuente de información, tal problema se complica más debido a que en el IISUE no se cuenta con personal debidamente capacitado para cumplir esas tareas y a que no se cuenta con suficientes sistemas informáticos que puedan dar solución a esas distintas tareas. Así la Coordinación de Humanidades[4], quien es la encargada de almacenar la información relacionada con las actividades, productos e investigaciones desarrolladas por parte de los Académicos, realiza el trabajo mediante una aplicación web, que se denomina SIAH (Sistema de Informes Académicos de Humanidades), este sistema alimenta una BD, la cual es filtrada y enviada anualmente en un archivo de Access al IISUE. Sin embargo, el poder obtener información estratégica y relevante de esa misma BD, y si fuera posible de otras fuentes de información, implicaría tomar mejores decisiones para el Personal Administrativo, ya que este mismo personal es quien se encarga de planear y administrar los recursos que son destinados principalmente para el desarrollo de la investigación y demás tareas que terminan siendo el objetivo primordial del IISUE y con ello también contar con información analítica y detallada de fácil acceso.

[4] La Coordinación de Humanidades es una institución fundada en 1945 dentro de la UNAM cuyo principal objetivo es el fomentar el desarrollo académico y en particular la investigación.

1.2.1. Aspectos generales

Entonces es así como el IISUE tiene la tarea de elaborar reportes anuales y que estos a su vez sirvan como base en la toma de decisiones con respecto a la planeación y administración de recursos. Tomando en cuenta esta parte, hemos propuesto la idea de analizar, diseñar e implementar un Data Warehouse, que brinde una solución en este tipo de tareas.

Con ello se puede acceder a tener información detallada y relevante sobre los aspectos que tienen que ver con el Personal Académico. También se podrá contar con reportes ad-hoc (reportes personalizados en tiempo real) que servirán para dar una mejor y personalizada exploración de los datos.

También a medida que pase el tiempo puede ser posible añadir nuevas fuentes de información que logren y permitan perseguir nuevos objetivos, ya que un Data Warehouse tiende a consolidar información de distintas fuentes de información a través del tiempo.

Por último se tendría un Depósito de Datos útil y confiable, sobre el cual estar explorando y analizando dicha información según se requiera, brindando con ello una solución a toda esa serie de tareas que son ejecutadas anualmente.

1.3. Objetivos

1.3.1. Generales

Analizar, diseñar e implementar la tecnología Data Warehouse en el IISUE, con la principal intención de que cuenten con un sistema informático gerencial que resuelva sus necesidades de tipo analítico y estadístico, cubriendo las principales actividades relacionadas con el Personal Académico (investigadores, profesores y técnico académicos).

1.3.2. Específicos

Los objetivos que a continuación se muestran son en base al desarrollo del Data Warehouse que se realizará en el presente trabajo:

- Explicar las principales metodologías de desarrollo de un Data Warehouse.
- Aplicar una metodología pertinente para nuestro caso de estudio.
- Realizar el diseño multidimensional para el área del Personal Académico.
- Realizar la extracción, transformación y carga de los datos pertenecientes al área del Personal Académico.
- Utilizar una herramienta para la exploración y generación de reportes ad-hoc, creando ejemplos de consultas.
- Determinar si es factible implantar actualmente la tecnología de Data Warehouse en el IISUE.

1.4. Alcances y limitaciones

Consideramos que los alcances de esta tesis son los siguientes:

- El desarrollo del Data Warehouse únicamente pretende cubrir el área del Personal Académico, esto por considerar que de ellos proviene la información más importante, ya que es en donde se realizan todas las actividades relacionadas a la investigación.

- Los cubos están destinados al personal de nivel directivo, es decir, está reservado para ser usado por usuarios que cuentan con la facultad de tomar decisiones dentro del Instituto.

- El trabajo realizado llegará hasta la creación de tres Data Marts y respecto a las tablas de hechos, estas se cargarán con datos ficticios de 3 años a partir de 2011.

- Probar y configurar una herramienta (Pentaho BI Server 4.8 Community Edition) que permita la exploración y la realización de consultas ad-hoc de los datos en el Data Warehouse.

Mientras que las limitantes están definidas a continuación:

- La integración de varias fuentes de información no pueden ser posibles dado que el instituto actualmente no cuenta con varias Bases de Datos formalmente diseñadas.

- No se podrá poner en producción el Data Warehouse debido a que actualmente no se cuenta con los requerimientos técnicos de hardware necesarios para su correcto funcionamiento.

CAPÍTULO 2. Data Warehouse y Data Marts

Actualmente toda empresa, organización o institución necesita de herramientas y procesos que le permitan tomar decisiones correctas sobre los negocios u operaciones que realiza, es por ello que surgen nuevos procesos que les ayudan a conocer mejor el entorno o negocio en el que se desarrollan. Por tal motivo se necesita principalmente de datos, los cuales serán la materia prima para estos procesos, posteriormente se integrará esta información mediante el uso de tecnologías nuevas e innovadoras que arrojarán información relevante, comprensible y detallada, que será crucial para lograr y sostener ventajas competitivas.

Ahora bien, el poseer conocimientos correctos significa tener respuestas correctas y esto a su vez realizar decisiones estratégicas en beneficio de la empresa. Pero las tareas que nos permiten lograrlo, como es el recolectar, procesar, depurar y transformar tal información necesaria, implica una tarea poco sencilla, más si consideramos que una empresa tienen distintas áreas, que a veces no suelen estar integradas en un mismo sistema. Sin embargo el Componente de Business Intelligence[5] (Inteligencia de Negocios) que resuelve este caos en el manejo y análisis de la información o datos, es el Data Warehouse.

[5] Business Intelligence es toda una estrategia de negocios y herramientas enfocadas a la administración y creación de conocimientos en una empresa.

2.1. Definición de un Data Warehouse

Existe una variedad de definiciones con respecto al Data Warehouse (DWH), pero todas ellas coinciden en un mismo término: colección de datos, a continuación presentamos algunas definiciones hechas por expertos en el desarrollo de Data Warehouse.

Un Data Warehouse (DWH) es una colección de datos orientada a un determinado ámbito (empresa, organización, etc.), integrado, no volátil y variable en el tiempo, que ayuda a la toma de decisiones en la entidad en la que se utiliza. [3]

Un DWH es una copia de los datos transaccionales estructurados específicamente para consultas y análisis. [4]

Un DWH o Depósito de Datos es una base de datos corporativa en la que se integra información depurada de las diversas fuentes que hay en la organización. [5]

Un DWH es la integración de datos consolidados, almacenados en un dispositivo de memoria no volátil, proveniente de múltiples y posiblemente diferentes fuentes de datos. Con el propósito del análisis y a partir de éste, tomar decisiones en función de mejorar la gestión del negocio. Contiene un conjunto de cubos de datos que permiten a través de técnicas de OLAP consolidar, ver y resumir los datos acorde a diferentes dimensiones de estos. [6]

Ahora bien tomando en cuenta las definiciones anteriores, creemos que la más conocida y más cercana a lo que es un DWH es la propuesta por William H. Inmon quien es considerado como el Padre del Data Warehouse. Un DWH es una colección de datos orientada a un determinado ámbito (empresa, organización, etc.), integrado, no volátil y variable en el tiempo, que ayuda a la toma de decisiones en la entidad en la que se utiliza.

También se debe mencionar que dicha colección de datos debe ser homogénea y fiable, además su almacenamiento debe ser de tal forma que permita su análisis desde muy diversas perspectivas y que a su vez, dé unos tiempos de respuesta óptimos en las consultas.

Para ello la información se encuentra altamente des normalizada y modelada de una forma bastante diferente a los sistemas operacionales[6].

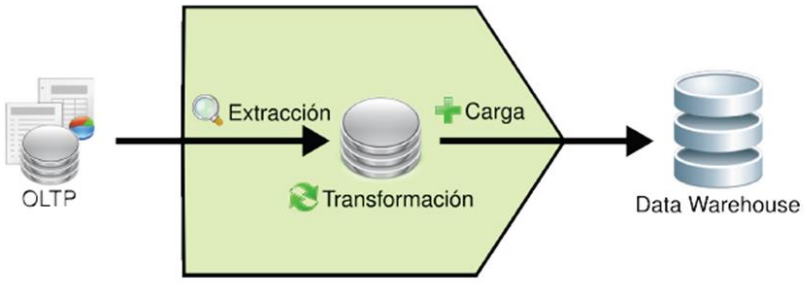


FIGURA 2.1. Data Warehouse (DWH)

[6] Se llama sistemas operacionales a las diferentes bases de datos de una empresa u organización, asociadas a sistemas que permiten el ingreso confiable de información y el procesamiento eficiente de transacciones. A dichos sistemas se les llama OLTP: On Line Transaction Processing

2.1.1. Características de un DWH

Las características fundamentales de un DWH se describen a continuación:

- **Orientado a temas**: la primera característica del DWH nos explica que la información se clasifica en base a los aspectos que son de interés para la organización. Esta clasificación afecta el diseño y la implementación de los datos encontrados en el almacén de datos, debido a que la estructura del mismo difiere considerablemente a la de los clásicos procesos operacionales orientados a las aplicaciones [9], es decir, en un DWH el diseño está orientado a la información estratégica, mientras que un sistema operacional está orientado a la información operativa.

- **Integrado:** un DWH debe contener datos integrados provenientes de las diversas fuentes de información mediante una estructura consistente, debiendo eliminarse las inconsistencias existentes entre los diversos sistemas operacionales. La información se estructura en diversos niveles de detalle para adecuarse a las necesidades de consulta de los usuarios [9]. Algunas de las inconsistencias más comunes que nos solemos encontrar son: en nomenclatura, en unidades de medida, en formatos de fechas, múltiples tablas con información similar (por ejemplo, hay varias aplicaciones con tablas de clientes), entre otras.

- **Histórico (variante en el tiempo):** los datos en el DWH, que pueden ir variando a lo largo del tiempo, deben quedar reflejados de forma que al ser consultados reflejen estos cambios y no se altere la realidad que había en el momento en que se almacenaron, evitando así la problemática que ocurre en los sistemas operacionales, que reflejan solamente el estado de la actividad de negocio presente. Un DWH debe almacenar los diferentes valores que toma una variable a lo largo del tiempo. [9]

- **No volátil:** la información de un DWH es útil para el análisis y la toma de decisiones solo cuando es estable [9], una vez introducida, debe ser de sólo lectura, nunca se modifica ni se elimina, y ha de ser permanente y mantenerse para futuras consultas.

Este esquema puede ejemplificarlo de manera más concreta.

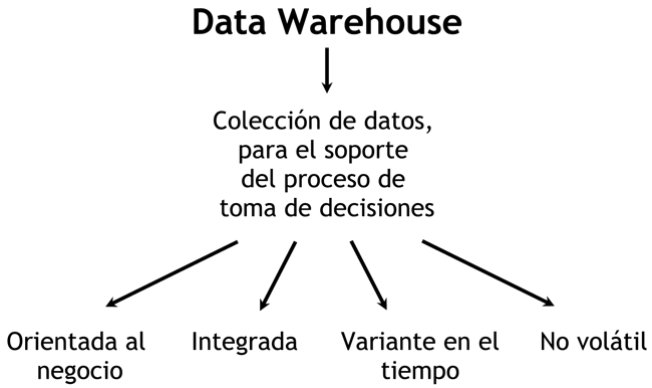


FIGURA 2.1.1. Esquema de Data Warehouse

2.1.2. Arquitectura de un DWH

La arquitectura de un Data Warehouse suele ser la forma de representar los procesos, la comunicación, y la presentación final al usuario de los datos que alberga un DWH [9], es decir, está definida en base a los componentes que en el DWH interactúan, esto se puede apreciar mejor en la FIGURA 2.1.2.

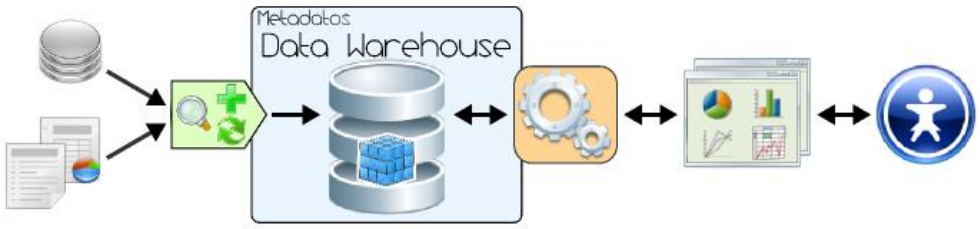


FIGURA 2.1.2. Arquitectura de un DWH

Tal y como se puede apreciar, el ambiente está formado por diversos componentes que interactúan entre sí y que cumplen una función específica dentro del sistema. Básicamente la forma de operar del esquema anterior se resume de la siguiente manera:

- Los datos operacionales son extraídos de las aplicaciones, ficheros, bases de datos o de fuentes externas (datos geográficos, demográficos, económicos, etc.). Esta información generalmente reside en diferentes tipos de sistemas, arquitecturas y plataformas que hacen que los formatos sean muy variados.

- Los datos son integrados, transformados y depurados, para luego ser cargados en el DWH.

- Posteriormente la información del DWH se estructura y se almacena en cubos multidimensionales, ya que estos preparan la información para responder a consultas dinámicas con un buen rendimiento. Pero también existe la posibilidad de poder utilizar otros tipos de estructuras de datos para representar la información del DWH.

- Se utilizan herramientas de acceso para explorar el DWH.

- Finalmente los usuarios acceden a los cubos multidimensionales del DWH utilizando las herramientas de acceso con el fin de realizar las consultas, el análisis y la visualización de datos en forma gráfica que sean de sumo interés para ellos.

2.1.3. Estructura de un DWH

El DWH estructura los datos de manera muy particular y existen diferentes niveles de esquematización y detalle que los delimitan. [9]

En la FIGURA 2.1.3. se puede observar su respectiva estructura:

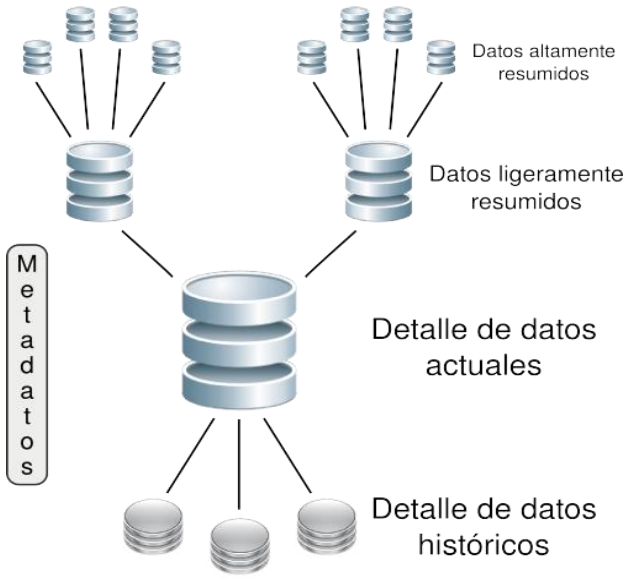


FIGURA 2.1.3. Estructura de un Data Warehouse

Como se puede observar, los almacenes de datos están compuestos por diversos tipos de datos, que se organizan y dividen de acuerdo al nivel de detalle o granularidad que posean. [9]

A continuación se explicarán cada uno de los tipos de datos:

- Detalle de datos actuales: son aquellos que reflejan las ocurrencias más recientes. Generalmente se almacenan en disco, aunque su administración sea costosa y compleja, con el fin de conseguir que el acceso a la información sea sencillo y veloz, ya que son bastante voluminosos. Su gran tamaño se debe a que los datos residentes poseen el más bajo nivel de granularidad, o sea, se almacenan a nivel de detalle. [9]

- Detalle de datos históricos: representan aquellos datos antiguos, que no son frecuentemente consultados. También se almacenan a nivel de detalle, normalmente sobre alguna forma de almacenamiento externa, ya que son muy pesados y en adición a esto, no son requeridos con mucha periodicidad. Este tipo de datos son consistentes con los de Detalle de datos actuales. [9]

- Datos ligeramente resumidos: son los que provienen desde un bajo nivel de detalle y sumarizan o agrupan los datos bajo algún criterio o condición de análisis. Habitualmente son almacenados en disco. Por ejemplo, en este caso se almacenaría la sumarización del detalle de las ventas realizadas en cada mes. [9]

- Datos altamente resumidos: son aquellos que compactan aún más a los datos ligeramente resumidos. Se guardan en disco y son muy fáciles de acceder. Por ejemplo, aquí se encontraría la sumarización de las ventas realizadas en cada año. [9]

- Metadatos: representan la información acerca de los datos. De muchas maneras se sitúa en una dimensión diferente al de otros datos del DWH, ya que su contenido no es tomado directamente desde el ambiente operacional. [9]

2.1.3.1. Metadatos

Los metadatos son datos que describen o dan información de otros datos, que en este caso, existen en la arquitectura del Data Warehouse. Brindan información de localización, estructura y significado de los datos, básicamente mapean los mismos. [9]

El concepto de metadatos es análogo al uso de índices para localizar objetos en lugar de datos. [9]

Es importante aclarar que existen metadatos también en las bases de datos transaccionales, pero los mismos son transparentes a los usuarios. La gran ventaja que traen sobre el DWH en relación con los metadatos es que los usuarios pueden gestionarlos, exportarlos, importarlos, realizarles mantenimiento e interactuar con ellos, ya sea manual o automáticamente. [9]

Las funciones que cumplen los metadatos en el ambiente Data Warehouse son muy importantes y significativas, algunas de ellas son:

- Facilitan el flujo de trabajo, convirtiendo datos automáticamente de un formato a otro. [9]

- Contienen un directorio para facilitar la búsqueda y descripción de los contenidos del DWH, tales como: bases de datos, tablas, nombres de atributos, indicadores, acumulaciones, reglas de negocios, estructuras y modelos de datos, relaciones de integridad, jerarquías, etc. [9]

- Poseen un guía para el mapeo, de cómo se transforman e integran los datos de las fuentes operacionales y externos al ambiente del depósito de datos. [9]

- Almacenan las referencias de los algoritmos utilizados para la esquematización entre el detalle de datos actuales, con los datos ligeramente resumidos y éstos con los datos altamente resumidos, etc. [9]

- Contienen las definiciones del sistema de registro desde el cual se construye el DWH. [9]

Se pueden distinguir tres diferentes tipos de Metadatos:

- Los metadatos de los procesos ETL, referidos a las diversas fuentes utilizadas, reglas de extracción, transformación, limpieza, depuración y carga de los datos al depósito. [9]

- Los metadatos operacionales, que son los que básicamente almacenan todos los contenidos del DWH, para que este pueda desempeñar sus tareas. [9]

- Los metadatos de consulta, que contienen las reglas para analizar y explotar la información del almacén, tales como drill-up y drill-down. Son estos metadatos los que las herramientas de análisis y consulta emplearán para realizar documentaciones y para navegar por los datos. [9]

2.1.4. Ventajas y desventajas de un DWH

Hasta este punto hemos descrito el concepto y las partes que integran a un DWH, sin embargo, no hemos tratado sus principales ventajas y desventajas al usar dicha tecnología. A continuación mostramos las ventajas más significativas del uso de los Data Warehouse.

- Puede ejecutar y procesar una gran cantidad de información de toda la organización.

- Es rápido y flexible al momento de consultar la información.

- Permite mejorar la toma de decisiones y el aumento de la productividad del negocio, puesto que permite conocer los resultados de la empresa, sean estos positivos o negativos.

- Es fiable la comunicación entre los diferentes departamentos de la empresa.

- Cualquier departamento puede consultar la información.

- Permite realizar los planes de la empresa en forma más efectiva, lo que ayuda a mejorar las relaciones entre el cliente y el proveedor.

Mientras que las desventajas generalmente suelen repercutir en los altos costos que genera el desarrollar este tipo de tecnología, las principales desventajas son las siguientes:

- El usar la tecnología de Data Warehouse implica realizar la reestructuración de los sistemas operacionales y esto da como resultado altos costos de desarrollo.

- Se deben hacer revisiones continuas de los modelos de datos, objetos y de las transacciones, lo cual termina provocando que el diseño sea mucho más complejo.

- Es necesario tener centros de almacenamiento específicos que ayuden a la misma seguridad de los datos.

2.2. Definición de un Data Mart

Un derivado del propio Data Warehouse suele ser el concepto de Data Mart (DM), ya que un Data Mart es un subconjunto lógico del Data Warehouse, sin embargo para aclarar mejor el significado de este término a continuación mostramos algunas definiciones.

Un DM es la implementación de un DWH con alcance restringido a un área funcional, problema en particular, departamento, tema o grupo de necesidades. [9]

Un DM es una estructura departamental que es alimentada desde el DWH, donde los datos están basados en las necesidades de la información, de los departamentos o áreas del negocio. [3]

La diferencia de un Data Mart (DM) con respecto a un Data Warehouse (DWH) es solamente en cuanto al alcance. Mientras que un Data Warehouse es un sistema centralizado con datos globales de la empresa y de todos sus procesos operacionales, un Data Mart es un subconjunto temático de datos, orientado a un proceso o área de negocio específica. Debe tener una estructura óptima desde todas las perspectivas que afecten a los procesos de dicha área. Según Ralph Kimball, cada Data Mart debe estar orientado a un proceso determinado dentro de la organización, por ejemplo: a pedidos de clientes, compras, inventarios de almacén, envío de materiales, etc. Para Kimball el conjunto de Data Marts forma el Data Warehouse. [5]

Dadas las definiciones anteriores el objetivo primordial de un DM es minimizar los riesgos y optimizar los tiempos de entrega que conlleva la creación de un DWH, por eso era necesario considerar sus características.

2.2.1. Características de un DM

Al igual que el Data Warehouse, los Data Mart tienen las mismas características de orientado a temas donde se clasifica tomando en cuenta los aspectos de interés para la organización, otra característica es la de integración de datos provenientes de diversas fuentes de información, así como otros dos rasgos el de no volatilidad y variante en el tiempo ya tratados en el subtema de características de un DWH.

2.2.2. Tipos de DM

Existen dos tipos de Data Marts los dependientes y los independientes, claramente definidos por las operaciones que se deseen o requieran desarrollar, además hay que mencionar que son alimentados en distinta forma.

- **Dependientes**

Los DM Dependientes son aquellos en donde primero está ya definido el DWH y luego los DM son desarrollados, construidos y cargados a partir del DWH. En el siguiente esquema se ilustra los DM Dependientes. [9]

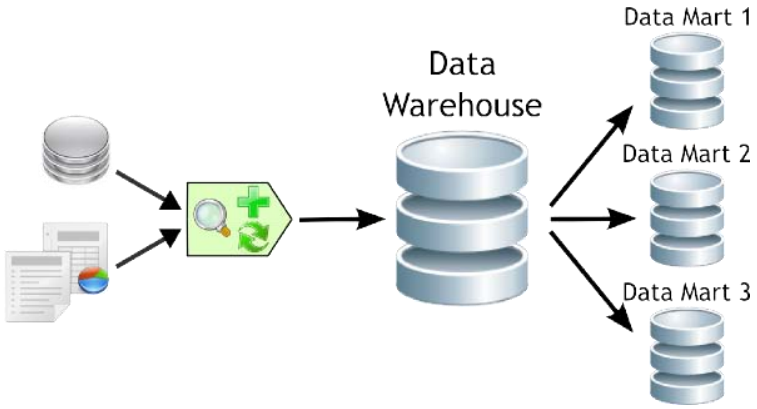


FIGURA 2.2.2.a. Data Marts Dependientes

Como se puede apreciar, el DWH es cargado a través de procesos ETL y luego este alimenta a los diferentes DM, cada uno de los cuales recibirá los datos que correspondan al tema o departamento que traten. Esta forma de implementación cuenta con la ventaja de no tener que incurrir en complicadas sincronizaciones de hechos, pero requiere una gran inversión y un periodo largo de construcción.

- **Independientes**

Los DM Independientes son aquellos en donde se define previamente los DM y estos son alimentados directamente por los sistemas operacionales y/o fuentes externas.

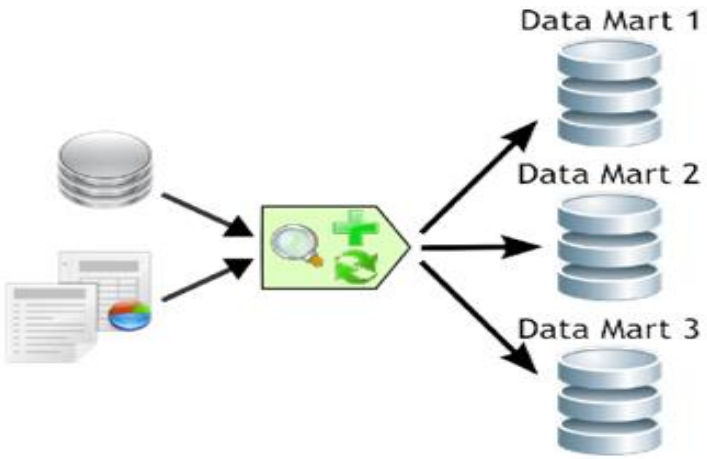


FIGURA 2.2.2.b. Data Marts Independientes

Los DM se cargan a través de procesos ETL, los cuales suministrarán la información adecuada a cada uno de ellos. En muchas ocasiones, los DM son implementados sin que exista el DWH, ya que tienen las mismas características pero con la particularidad de que están enfocados en un tema específico. Luego de que hayan sido creados y cargados todos los DM, se procederá a su integración con el depósito. La ventaja que trae aparejada este modelo es que cada DM se crea y pone en funcionamiento en un corto lapso de tiempo, también se puede tener una pequeña solución a un costo no tan elevado. Luego que todos los DM estén puestos en marcha, se puede decidir si construir el DWH o no. El mayor inconveniente está dado en tener que sincronizar los hechos al momento de la consolidación en el depósito. [9]

2.2.3. Ventajas y desventajas de un DM

A continuación se desglosarán las principales ventajas que implica desarrollar Data Marts.

- Dado a que un Data Mart soporta menos usuarios que un Data Warehouse se puede este optimizar para dar mejores rendimientos de acceso.

- Las peticiones o requerimientos pueden acotarse al área o departamento en el que se está desarrollando el Data Mart, sin afectar a los demás usuarios.

- Los costos que implica la construcción de un Data Mart son menores comparados con la implementación de todo un Data Warehouse.

Mientras que la principal desventaja es que un solo Data Mart no puede contener varios temas de análisis, pues se debe recurrir a un conjunto de Data Marts para poder cubrir dicha necesidad y también requerir un conjunto bastante grande de Data Marts, lo que implicaría un alto costo de procesamiento o complejidad en el diseño.

2.3. Componentes que integran a un DWH

2.3.1. OLTP y Fuentes Externas

Los sistemas OLTP (Procesamiento de Transacciones En Línea) están diseñados para gestionar una gran cantidad de transacciones concurrentes sobre sus bases de datos, y que los usuarios o clientes puedan insertar, modificar, borrar y consultar dichos datos.

Están enfocados a que cada transacción trabaje con pequeñas cantidades de filas, y a que ofrezcan una respuesta rápida. Habitualmente utilizan sistemas de bases de datos relacionales para gestionar los datos, y suelen estar altamente normalizados [9]. En ellos es muy importante la integridad de los datos y deben cumplir las propiedades ACID (Atomicity, Consistency, Isolation, Durability):

- **Atomicidad:** es una operación, que se realiza por completo o no se realiza, nunca debe quedar a medias.

- **Consistencia:** sólo se ejecutan las operaciones que cumplen las reglas de integridad de la base de datos.

- **Aislamiento (Isolation):** una operación no puede afectar a otras dos transacciones sobre los mismos datos, son independientes y no generan errores entre sí.

- **Durabilidad:** una vez realizada una operación, ésta es persistente y no se puede deshacer

En general es cualquier aplicación con la que el usuario interactúa para introducir datos al sistema, como ejemplos de este tipo de sistemas, podemos citar las aplicaciones departamentales, aplicaciones de ventas, de comercio electrónico, entre muchas otras.

Por otra parte las fuentes externas son aquellas en donde la información es ajena a las BD de la misma organización, es decir, son BD que albergan datos demográficos, de competitividad en el mercado o cualquier otra fuente que sea útil para tener un mejor análisis de los datos que son integrados en el DWH.

2.3.2. ETL (Extracción, Transformación y Carga)

Un DWH o un DM, se va cargando periódicamente y en él se unifica información procedente de múltiples fuentes de datos, creando una base de datos que cumple con una lista de características ya descritas anteriormente. Lo cual implica que deben existir una serie de procesos que leen los datos de las diferentes fuentes de datos, los transforman, adaptan al modelo que se haya definido, los depuran, limpian y los introducen en esta base de datos de destino. Esto es lo que se conoce como procesos ETL procesos de Extracción, Transformación y Carga. [9]

La integración de datos agrupa una serie de técnicas y subprocesos que se encargan de llevar a cabo todas las tareas relacionadas con la extracción, manipulación, control, integración, depuración de datos, carga y actualización del DWH, etc. Así es como la serie de técnicas y subprocesos que se realizan desde que se toman los datos de origen (OLTP y/o Fuentes Externas) hasta que se cargan en el DWH, son tareas imprescindibles que nos darán datos exactos e íntegros para dar confiabilidad a nuestro DWH.

FIGURA 2.3.2. Proceso ETL

A continuación, se detallará cada una de estas etapas, se expondrá cuáles son los procesos que llevan a cabo los ETL y se enumerarán sus principales tareas.

2.3.2.1. Extracción

Es aquí, en donde, basándose en las necesidades y requisitos de los usuarios, se exploran las diversas fuentes OLTP que se tengan a disposición y se extrae la información que se considere relevante al caso.

Si los datos operacionales residen en un Sistema de Gestión de Bases de Datos (SGBD) Relacional, el proceso de extracción se puede reducir por ejemplo, a consultas en SQL o procedimientos almacenados. En cambio, si se encuentran en un sistema no convencional o fuentes externas, ya sean textuales, hipertextuales, hojas de cálculos, etc., la obtención de los mismos puede ser un tanto más dificultoso, debido a que se tendrán que realizar cambios de formato y/o volcado de información a partir de alguna herramienta específica. [9]

Una vez que los datos son seleccionados y extraídos, se guardan en un almacenamiento intermedio, lo cual permite, entre otras ventajas:

- Manipular los datos sin interrumpir, ni paralizar los OLTP, ni tampoco el DWH.
- No depender de la disponibilidad de los OLTP.
- Almacenar y gestionar los metadatos que se generarán en los procesos ETL.
- Facilitar la integración de las diversas fuentes, internas y externas.

El almacenamiento intermedio constituye en la mayoría de los casos una base de datos en donde la información puede ser almacenada unos ejemplos de esto son las tablas auxiliares, tablas temporales, etc. Los datos de estas tablas serán los que finalmente (luego de su correspondiente transformación) poblarán el DWH. [9]

2.3.2.2. Transformación

Esta función es la encargada de convertir aquellos datos inconsistentes en un conjunto de datos compatibles y congruentes, para que puedan ser cargados en el DWH. Estas acciones se llevan a cabo, ya que pueden existir diferentes fuentes de información y es vital conciliar un formato y forma única, definiendo estándares, para que todos los datos que ingresarán al DWH estén integrados.

Los casos más comunes en los que se deberá realizar integración, son los siguientes:

- Codificación. Una inconsistencia muy típica que se encuentra al intentar integrar varias fuentes de datos, es la de contar con más de una forma de codificar un atributo en común. Por ejemplo, en el campo “estatus”, algunos diseñadores completan su valor con “0” y “1”, otros con “Apagado” y “Encendido”, otros con “off” y “on”, etc. Lo que se debe realizar en estos casos, es seleccionar o recodificar estos atributos, para que cuando la información llegue al DWH, esté integrada de manera uniforme. En la FIGURA 2.3.2.2.a., se puede apreciar que de varias formas de codificar se escoge una, entonces cuando surge una codificación diferente a la seleccionada, se procede a su transformación. [9]

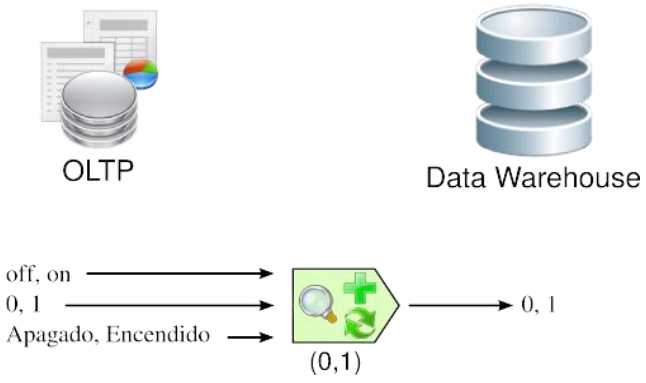


FIGURA 2.3.2.2.a. Codificación

- Medida de atributos. Los tipos de unidades de medidas utilizados para representar los atributos de una entidad, varían considerablemente entre sí, a través de los diferentes OLTP. Por ejemplo, al registrar la longitud de un producto determinado, de acuerdo a la aplicación que se emplee para tal fin, las unidades de medida pueden ser explicitadas en centímetros, metros, pulgadas, etc. En esta ocasión, se deberán estandarizar las unidades de medida de los atributos, para que todas las fuentes de datos expresen sus valores de igual manera. Los algoritmos que resuelven estas inconsistencias son generalmente los más complejos. [9]

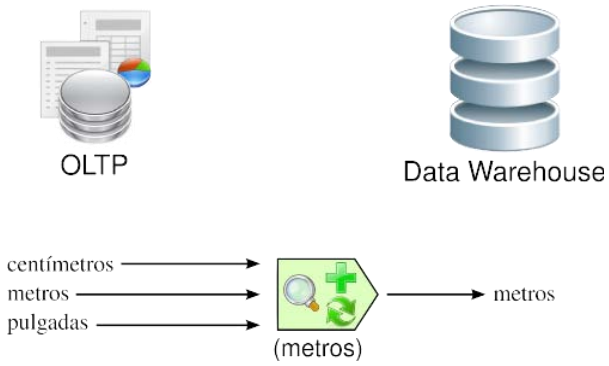


FIGURA 2.3.2.2.b. Medida de atributos

- Convenciones de nombramiento. Usualmente, un mismo atributo es nombrado de diversas maneras en los diferentes OLTP. Un ejemplo sería: al referirse al nombre del proveedor, puede hacerse como “nombre”, “razón_social”, “proveedor”, etc. Aquí, se debe utilizar la convención de nombramiento que para los usuarios sea más comprensible.

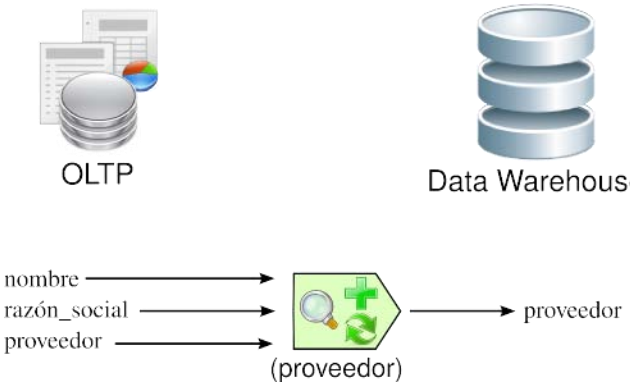


FIGURA 2.3.2.2.c. Convenciones de nombramiento

- Limpieza de datos. Su objetivo principal es el de realizar distintos tipos de acciones contra el mayor número de datos erróneos, inconsistentes e irrelevantes. [9]

    - Las acciones más típicas que se pueden llevar a cabo al encontrarse con Datos Anómalos son:

        - Ignorarlos.
        - Eliminar la columna.
        - Filtrar la columna.
        - Filtrar la fila errónea, ya que a veces su origen, se debe a casos especiales.
        - Reemplazar el valor.
        - Discretizar los valores de las columnas. Por ejemplo de 1 a 2, poner “bajo”; de 3 a 7, “óptimo”; de 8 a 10, “alto”.

    - Las acciones que suelen efectuarse contra Datos Faltantes son:

        - Ignorarlos.
        - Eliminar la columna.
        - Filtrar la columna.
        - Filtrar la fila errónea, ya que a veces su origen, se debe a casos especiales.
        - Reemplazar el valor.
        - Esperar hasta que los datos faltantes estén disponibles.

Un punto muy importante que se debe tener en cuenta al elegir alguna acción, es el de identificar el porqué de la anomalía, para luego actuar en consecuencia, con el fin de evitar que se repitan, agregándole de esta manera más confiabilidad a los datos de la organización.

2.3.2.3. Carga

La carga es un proceso que puede ser ejecutado por etapas o por lapsos de tiempo, dependiendo de cuál sea el mejor caso para alimentar nuestro DWH.

La carga inicial, se refiere precisamente a la primera carga de datos que se le realizará al DWH. Por lo general, esta tarea consume un tiempo bastante considerable, ya que se deben insertar registros que han sido procesados por primera vez y provenientes de periodos grandes de tiempo. [9]

Los mantenimientos periódicos mueven pequeños volúmenes de datos, y su frecuencia está dada en función al nivel de detalle del DWH y los requerimientos de los usuarios. El objetivo de esta tarea es añadir al DWH aquellos datos nuevos que se fueron generando desde la última actualización. [9]

Antes de realizar una nueva actualización, es necesario identificar si se han producido cambios en las fuentes originales de los datos recogidos, desde la fecha del último mantenimiento, a fin de no atentar contra la consistencia del DWH. Para efectuar esta operación, se pueden realizar las siguientes acciones:

- Recurrir a Marcas de Tiempo (Time Stamp), en los registros de los OLTP. Esto con el fin de saber cuándo fueron insertados los datos y posteriormente marcar límites de cargas entre los registros. Además de comparar los datos existentes en los dos ambientes (OLTP y DWH)

- Cotejar las diferentes fuentes de datos de los OLTP.

- Comparar los datos existentes en los dos ambientes (OLTP y DWH).

Si este control consume demasiado tiempo y esfuerzo, o simplemente no puede llevarse a cabo por algún motivo en particular, existe la posibilidad de cargar el DWH desde cero, este proceso se denomina Carga Total.

En síntesis el proceso de ETL es un proceso vital para el desarrollo del DWH ya que es el encargado de alimentar y mantener dicho sistema. Sus tres pasos se pueden describir brevemente de la siguiente forma:

- Primero se extraen los datos relevantes desde los OLTP y se depositan en un almacenamiento intermedio.
- Se integran y transforman los datos, para evitar inconsistencias.
- Se cargan los datos desde el almacenamiento intermedio hasta el DWH.

2.3.3. Modelado Multidimensional

El Diagrama de datos Entidad - Relación, es comúnmente usado en el diseño de bases de datos transaccionales. Sin embargo, el modelado de datos en un DWH no puede realizarse en un modelo Entidad - Relación, debido a que los usuarios se centran en la consulta de grandes cantidades de registros y no en el guardar, eliminar y actualizar pequeñas cantidades de registros. [9]

Una base de datos multidimensional es una base de datos en donde su información se almacena en forma multidimensional, es decir, a través de tablas de hechos y tablas de dimensiones. Además proveen una estructura de datos determinada (cubos OLAP) que permite, tener acceso flexible a los datos, para explorar y analizar sus relaciones, con sus respectivos hechos.

Las bases de datos multidimensionales se pueden modelar mediante tres posibles esquemas, los cuales permiten realizar consultas sobre los datos en un ambiente DHW:

- Esquema en Estrella (Star Schema).
- Esquema en Copo de Nieve (Snowflake Schema).
- Esquema Constelación (Constellation Schema).

Los mencionados esquemas pueden ser implementados sobre 3 distintas formas, que independientemente al tipo de arquitectura, requieren que toda la estructura de datos este desnormalizada o semi desnormalizada, para evitar desarrollar uniones (joins)

complejas para acceder a la información, con el fin de agilizar la ejecución de consultas [9]. Los diferentes tipos de implementación son los siguientes:

- Relacional — ROLAP

  Es decir, Relational On Line Analytic Processing (ROLAP) cuenta con todos los beneficios de una SGBD Relacional a los cuales se les provee extensiones y herramientas para poder utilizarlo como un Sistema Gestor de DWH.

  Este tipo de organización física se implementa sobre tecnología relacional, pero disponen de algunas facilidades para mejorar el rendimiento.

- Multidimensional — MOLAP

  El objetivo de los sistemas MOLAP (Multidimentional On Line Analytic Processing) es almacenar físicamente los datos en estructuras multidimensionales de manera que la representación externa y la interna coincidan.

  Para ello, se dispone de estructuras de almacenamiento específicas (Arrays) y técnicas de compactación de datos que favorecen el rendimiento del DWH.

- Híbrido — HOLAP

  HOLAP (Hybrid On Line Analytic Processing) constituye un sistema híbrido entre MOLAP y ROLAP, que combina estas dos implementaciones para almacenar algunos datos en un motor relacional y otros en una base de datos multidimensional.

2.3.3.1. Tipos de Modelado

Dentro del modelado de un DWH, hay que decidir cuál es el esquema más apropiado para obtener los resultados que deseamos conseguir.

- **Esquema en Estrella (Star Schema)**

Comúnmente se suele modelar el DWH utilizando el esquema en estrella (star schema). El esquema en estrella, consta de una tabla de hechos central y de varias tablas de dimensiones relacionadas a esta, a través de sus respectivas claves [9]. En la FIGURA 2.3.3.1.a. se puede apreciar un esquema en estrella estándar:

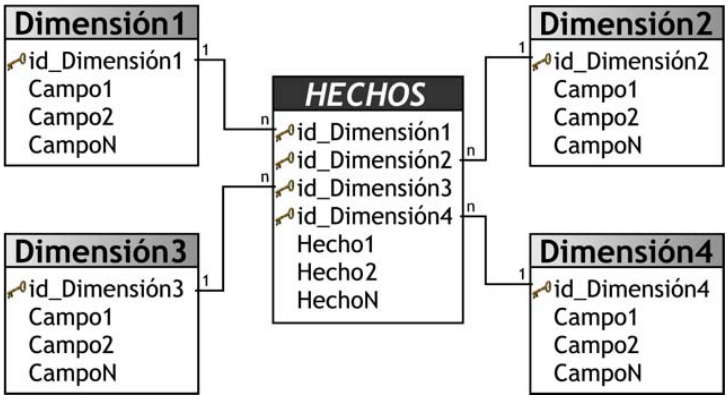


FIGURA 2.3.3.1.a. Esquema Estrella

El esquema en estrella es el más simple de interpretar y optimiza los tiempos de respuesta ante las consultas de los usuarios. Este modelo es soportado por casi todas las herramientas de consulta y análisis, aquí los metadatos son fáciles de documentar y mantener, sin embargo es el menos robusto para la carga y más redundante.

A continuación se destacarán algunas características de este modelo, que ayudarán a comprender mejor el porqué de sus ventajas:

- Posee los mejores tiempos de respuesta.
- Su diseño es fácilmente modificable.
- Simplifica el análisis.
- Facilita la interacción con herramientas de consulta y análisis.

➢ Esquema en Copo de Nieve (Snowflake Schema)

La otra alternativa de modelado es el uso del esquema en copo de nieve (snowflake schema). Esta es una estructura más compleja que el esquema en estrella. La diferencia es que algunas de las dimensiones no están relacionadas directamente con la tabla de hechos, sino que se relacionan con ella a través de otras dimensiones. [9]

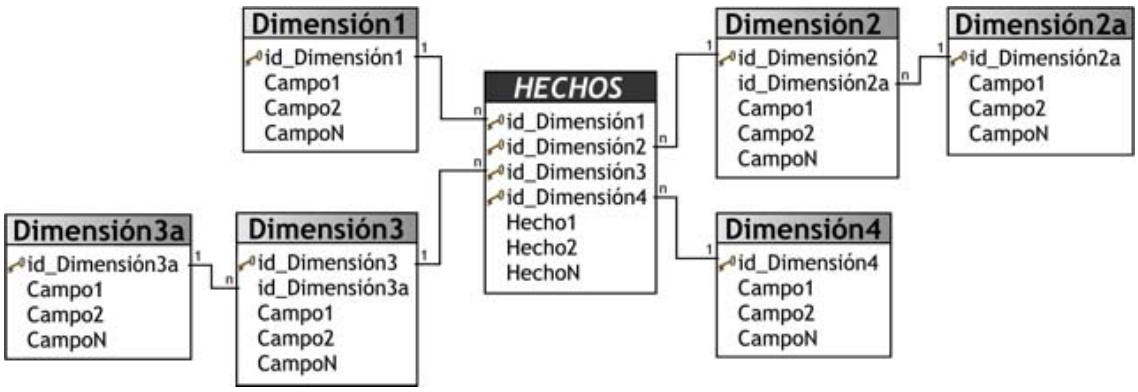


FIGURA 2.3.3.1.b. Esquema Copo de Nieve

Como se puede apreciar en la FIGURA 2.3.3.1.b., existe una tabla de hechos central que está relacionada con una o más tablas de dimensiones, quienes a su vez pueden estar relacionadas o no con una o más tablas de dimensiones.

Este modelo es más cercano a un modelo de entidad relación, que al modelo en estrella, debido a que sus tablas de dimensiones están normalizadas.

Una de los motivos principales de utilizar este tipo de modelo, es la posibilidad de segregar los datos de las tablas de dimensiones y proveer un esquema que sustente los requerimientos de diseño. Otra razón es que es muy flexible y puede implementarse después de que se haya desarrollado un esquema en estrella [9]. Se pueden definir las siguientes características de este tipo de modelo:

- Posee mayor complejidad en su estructura.
- Hace una mejor utilización del espacio de almacenamiento.
- Es muy útil en tablas de dimensiones de muchos registros.
- Las tablas de dimensiones están normalizadas, por lo que requiere menos esfuerzo de diseño.
- Puede desarrollar clases de jerarquías fuera de las tablas de dimensiones, que permiten realizar análisis de lo general a lo detallado y viceversa

- **Esquema Constelación (Constellation Schema)**

Este modelo está compuesto por una serie de esquemas en estrella, tal como se puede apreciar en la FIGURA 2.3.3.1.c., está formado por dos o más tablas de hechos (“HECHOS_A” y “HECHOS_B”), las cuales pueden estar orientadas a temas distintos. Dichas tablas yacen en el centro del modelo y están relacionadas con sus respectivas tablas de dimensiones.

No es necesario que las diferentes tablas de hechos compartan las mismas tablas de dimensiones, ya que cada tabla de hechos puede vincularse con solo algunas de las tablas de dimensiones asignadas a la tabla de hechos principal y también pueden hacerlo con nuevas tablas de dimensiones. [9]

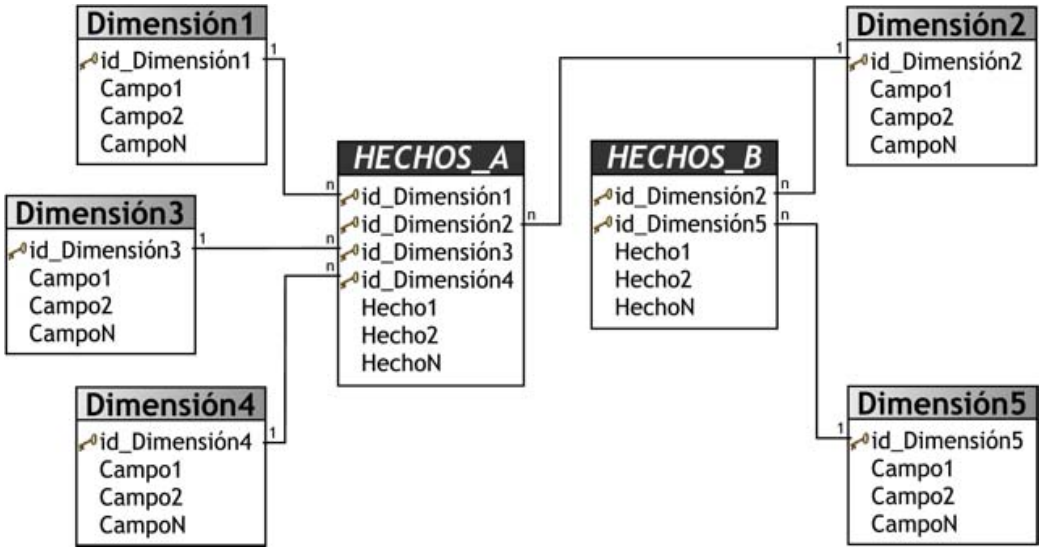


FIGURA 2.3.3.1.c. Esquema Constelación

Su diseño y cualidades son muy similares a las del esquema en estrella, pero posee una serie de diferencias con el mismo, que son precisamente las que lo destacan y caracterizan. Entre ellas se pueden mencionar:

- Permite tener más de una tabla de hechos, por lo cual se podrán analizar más aspectos claves del negocio con un mínimo esfuerzo adicional de diseño.
- Contribuye a la reutilización de las tablas de dimensiones, ya que una misma tabla de dimensión puede utilizarse para varias tablas de hechos.
- No es soportado por todas las herramientas de consulta y análisis.

2.3.3.2. Tablas de Dimensiones

Las tablas de dimensiones definen cómo están los datos organizados lógicamente además proveen el medio para analizar el contexto de los hechos. Contienen datos cualitativos que representan los aspectos de interés, mediante los cuales los usuarios podrán filtrar y manipular la información almacenada en la tabla de hechos. [9]

Cada tabla posee un identificador único y al menos un campo o dato de referencia que describe los criterios de análisis relevantes para la organización, estos son por lo general de tipo texto. Los datos dentro de estas tablas, que proveen información del negocio o que describen alguna de sus características, son llamados datos de referencia.

En un DWH, la creación y el mantenimiento de una tabla de dimensión Tiempo es obligatoria, así como la definición de granularidad y estructuración de la misma depende del nivel de detalle del negocio que se esté analizando. Toda la información dentro del depósito, como ya se ha explicado, posee su propio periodo de tiempo que determina la ocurrencia de un hecho específico, representando de esta manera diferentes versiones de una misma situación. [9]

2.3.3.3. Tablas de Hechos

Las tablas de hechos contienen, precisamente, los datos agregados, que serán utilizados por los usuarios para apoyar el proceso de toma de decisiones. Contienen datos cuantitativos y los hechos son datos instantáneos en el tiempo, que son filtrados, agrupados y explorados a través de condiciones definidas en las tablas de dimensiones. [9]

Los datos presentes en las tablas de hechos constituyen el volumen del DWH, y pueden estar compuestos por millones de registros dependiendo de su nivel granularidad (nivel de detalle) y antigüedad de la organización.

El registro del hecho posee una clave primaria que está compuesta por las claves primarias de las tablas de dimensiones relacionadas a este.

Existen dos tipos de hechos, los básicos y los derivados, a continuación se destacan las características de cada uno de ellos:

- Hechos básicos: son los que se encuentran representados por un campo de tipo numérico de una tabla de hechos. [9]

- Hechos derivados: son los que se forman al combinar uno o más hechos con alguna operación matemática o lógica y que también residen en la tabla de hechos. Estos poseen la ventaja de almacenarse previamente calculados, por lo cual pueden ser accedidos a través de consultas SQL sencillas y devolver resultados rápidamente, pero requieren más espacio físico en el DWH, además de necesitar más tiempo de proceso en los ETL que los calculan. [9]

2.3.3.4. Cubos OLAP

Un cubo OLAP, *OnLine Analytical Processing* o procesamiento Analítico en Línea, término acuñado por Edgar Frank Codd de *EF Codd & Associates*, que lo define como una base de datos multidimensional. Mientras que otros lo definen básicamente como una estructura de datos organizada mediante jerarquías y estas a su vez mantienen intersecciones que son llamados indicadores, con el objetivo de ser una de las formas más populares de analizar la información.

Un cubo OLAP o cubo multidimensional, representa o convierte los datos planos que se encuentran en filas y columnas, en una matriz de N dimensiones. Los objetos más importantes que se pueden incluir en un cubo, son los siguientes: [9]

- Indicadores: son las sumarizaciones[7] que se efectúan sobre algún hecho o expresiones basadas en sumarizaciones, pertenecientes a una tabla de hechos. [9]

- Atributos: son los campos o criterios de análisis, pertenecientes a las tablas de dimensiones. [9]

- Jerarquías: representa una relación lógica entre dos o más atributos de una tabla de dimensiones. [9]

De esta manera en un cubo, los atributos existen a lo largo de varios ejes o dimensiones y la intersección de las mismas representa el valor que tomará el indicador que se está evaluando.

[7] Las sumarizaciones no están referidas solo a sumas, sino también a promedios, mínimos, máximos, totales, porcentajes, formulas predefinidas, etc, dependiendo de los requerimientos del DWH.

En la siguiente representación matricial se puede apreciar más claramente lo que se acaba de mencionar:

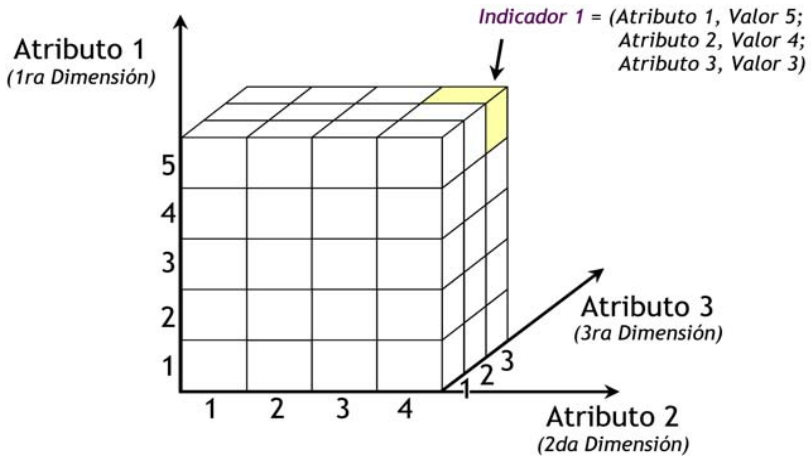


FIGURA 2.3.3.4. Estructura de un Cubo OLAP

Para la creación del cubo de la FIGURA 2.3.3.4., se definieron tres Atributos (“Atributo 1”, “Atributo 2” y “Atributo 3”) y se definió un Indicador (“Indicador 1”). Entonces el cubo quedó compuesto por 3 dimensiones o ejes (una por cada Atributo), cada una con sus respectivos valores asociados. También, se ha seleccionado una intersección al azar para demostrar la correspondencia con los valores de las Atributos. En este caso, el indicador “Indicador 1”, representa el cruce del valor “5” de “Atributo 1”, con el valor “4” de “Atributo 2” y con el valor “3” de “Atributo 3”. [9]

2.3.3.4.1. Indicadores, Atributos y Jerarquías

Los indicadores son las sumarizaciones efectuadas sobre algún hecho o expresiones basadas en sumarizaciones, que serán incluidos en algún cubo multidimensional, con el fin de analizar los datos almacenados en el DWH. El valor que estos adopten estará condicionado por los atributos y jerarquías que se utilicen para analizarlos. [9]

Mientras que los atributos constituyen los criterios de análisis que se utilizarán para analizar los indicadores dentro de un cubo multidimensional. Los mismos se basan en su gran mayoría, en los campos de las tablas de dimensiones y/o expresiones. Dentro de un cubo multidimensional, los atributos son los ejes del mismo.

Y una jerarquía representa una relación lógica entre dos o más atributos pertenecientes a un cubo multidimensional; siempre y cuando posean su correspondiente relación padre-hijo. Las jerarquías poseen las siguientes características:

- Pueden existir varias en un mismo cubo multidimensional.
- Están compuestas por dos o más niveles.
- Se tiene una relación 1-n o padre-hijo entre atributos consecutivos de un nivel superior y uno inferior.

2.3.3.4.2. Relación

Una relación representa la forma en que dos atributos interactúan dentro de una jerarquía. Existen básicamente dos tipos de relaciones:

- Explícitas: son las más comunes, se pueden modelar a partir de atributos directos y están en línea continua de una jerarquía, por ejemplo, un país posee una o más provincias y una provincia pertenece solo a un país. [9]

- Implícitas: son las que ocurren en la vida real, pero su relación no es de vista directa, por ejemplo, una provincia tiene uno o más ríos, pero un río pertenece a una o más provincias. Como se puede observar, en este caso, se trata de una relación muchos a muchos. [9]

2.3.3.4.3. Granularidad

La granularidad representa el nivel de detalle al que se desea almacenar la información sobre el negocio que se esté analizando. La unidad de medida determinará esa granularidad, cuanto más pequeña sea esta unidad de medida más fina será esta granularidad (grano fino); si las unidades de medida son mayores, entonces hablaremos de granularidad gruesa (grano grueso).

Por ejemplo, los datos referentes a ventas o compras realizadas por una empresa, pueden registrarse día a día, en cambio, los datos pertinentes a pagos de sueldos o cuotas de socios, podrán almacenarse a nivel de mes.

Mientras mayor sea el nivel de detalle de los datos, se tendrán mayores posibilidades analíticas, ya que los mismos podrán ser resumidos o sumarizados. Es decir, los datos que posean granularidad fina (nivel de detalle) podrán ser resumidos hasta obtener una granularidad media o gruesa. No sucede lo mismo en sentido contrario, ya que, los datos almacenados con granularidad media podrán resumirse, pero no tendrán la facultad de ser analizados a nivel de detalle. Lo cual se explica así, si la granularidad con que se guardan los registros es a nivel de día, estos datos podrán sumarizarse por semana, mes, semestre y año, en cambio, si estos registros se almacenan a nivel de mes, podrán sumarizarse por semestre y año, pero no lo podrán hacer por día y semana. [9]

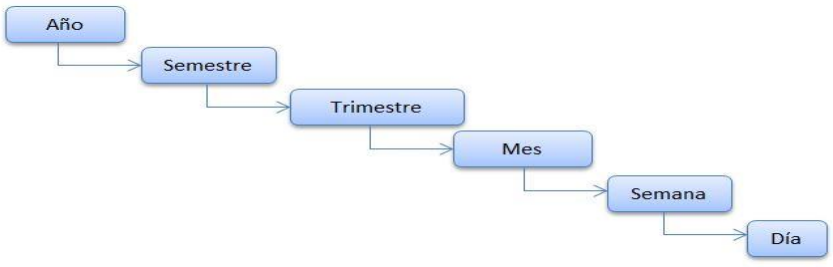


FIGURA 2.3.3.4.3. Nivel de granularidad sobre la dimensión Tiempo

2.3.4. Usuarios

Los usuarios que poseen el DWH son aquellos que se encargan de tomar decisiones y de planificar las actividades del negocio, es por ello que se hace tanto énfasis en la integración, limpieza de datos, etc., para poder conseguir que la información tenga toda la calidad posible. Es a través de las herramientas de consulta y análisis, que los usuarios exploran los datos en busca de respuestas para poder tomar decisiones proactivas.

Para comprender mejor a los usuarios del almacén de datos, se hará referencia a las diferencias que estos tienen con respecto a los del OLTP:

- Los usuarios que acceden al DWH concurrentemente son pocos, en cambio los que acceden a los OLTP en un tiempo determinado son muchos más, pueden ser cientos o incluso miles. Esto se debe principalmente al tipo de información que contiene cada fuente.

- Los usuarios del DWH generan por lo general consultas complejas, no predecibles y no anticipadas. Usualmente, cuando se encuentra una respuesta a una consulta se formulan nuevas preguntas más detalladas y así sucesivamente. Es decir, primero se analiza la información a nivel de datos actual para averiguar el “qué”, luego para obtener mayor detalle y examinar el “cómo”, se trabajan con los datos ligeramente resumidos, derivados de la consulta anterior y desde allí se puede explorar los datos altamente resumidos. Teniendo en cuenta siempre la posibilidad de utilizar el detalle de datos histórico. Al contrario, los usuarios de los OLTP solo manejan consultas predefinidas.

- Los usuarios del DWH, generan consultas sobre una gran cantidad de registros, en cambio los del OLTP lo hacen sobre un pequeño grupo. Esto se debe a que como ya se ha mencionado, el depósito contiene información histórica e integra varias fuentes de datos.

- Las consultas de los usuarios del DWH no tienen tiempos de respuesta críticos, aunque sí se espera que se produzcan en el mismo día en que fueron realizadas. Mientras mayor sea el tamaño del depósito y mientras más compleja sea la consulta, mayores serán los tiempos de respuestas. En cambio, las respuestas de las consultas en un OLTP son y deben ser inmediatas.

- En los OLTP, los usuarios típicamente realizan actualizaciones, tales como agregar, modificar, eliminar y consultar algún registro. En cambio en un DWH, la única operación que pueden realizar es la de consulta.

CAPÍTULO 3. Herramientas de Software Libre para la construcción de un DWH

En nuestra vida diaria toda empresa, dependencia u organización debe tomar en cuenta que es necesario contar con herramientas o aplicaciones que le permitan procesar y desarrollar cualquier sistema complejo que cubra las necesidades de dicha empresa, por lo tanto esto nos llevara a pensar y a decidir el uso de herramientas que coincidan con nuestros requerimientos tanto de plataforma como de hardware.

En este capítulo se explicarán las características de las distintas herramientas que se utilizarán y nos ayudarán para facilitarnos la tarea del desarrollo así como la implementación de un Data Warehouse para el Instituto de Investigaciones sobre la Universidad y la Educación.

3.1. ¿Qué es Software Libre?

Software Libre (o programas libres) es aquel software que garantiza al usuario los derechos de ejecución, acceso a su código fuente para revisión y modificación, generación del programa a partir de su código fuente y libre distribución tanto de la versión original obtenida como de sus modificaciones. Por lo tanto retomaremos el argumento de Richard Stallman[8], que lo definió como: un programa es Software Libre si los usuarios tienen las cuatro libertades esenciales:

- La libertad de ejecutar el programa para cualquier propósito (libertad 0). [10]

- La libertad de estudiar cómo funciona el programa y cambiarlo para que haga lo que usted quiera (libertad 1). El acceso al código fuente es una condición necesaria para ello. [10]

- La libertad de redistribuir copias para ayudar a su prójimo (libertad 2). [10]

- La libertad de distribuir copias de sus versiones modificadas a terceros (libertad 3). Esto le permite ofrecer a toda la comunidad la oportunidad de beneficiarse de las modificaciones. El acceso al código fuente es una condición necesaria para ello. [10]

El ser libre de hacer estas cosas significa, entre otras, que no tiene que pedir ni pagar el permiso. Para que estas libertades sean reales, deben ser permanentes e irrevocables siempre que usted no cometa ningún error; si el programador del software tiene el poder de revocar la licencia, o de añadir restricciones a las condiciones de uso, sin que haya habido ninguna acción de parte del usuario que lo justifique, el software no es libre. [10]

[8] Líder fundador de la Fundación para el Software Libre (FSF) y creador del proyecto GNU

Sin embargo, ciertos tipos de reglas sobre la manera de distribuir Software Libre son aceptables, cuando no entran en conflicto con las libertades principales. Por ejemplo, el copyleft es la regla en base a la cual, cuando redistribuye el programa, no puede agregar restricciones para denegar a los demás las libertades principales. Esta regla no entra en conflicto con las libertades principales, más bien las protege. [10]

3.2. Elección de Software

En base al tema expuesto anteriormente y tomando en cuenta los altos costos que tiene la compra de licencias y el soporte técnico que otorga el Software Privado para el desarrollo de un DWH, hemos decidido inclinarnos por usar productos de Software Libre, ya que para empezar no tienen un costo de licencia, además nos permiten ejecutarlos para cualquier propósito y entre otras cosas nos da la libertad de poder usarlos o modificarlos de acuerdo a nuestros requerimientos.

Para realizar el desarrollo del DWH hemos propuesto elegir como plataforma o base de implementación, un sistema operativo Linux denominado CentOS (Community ENTerprise Operating System) 5.8 ya que es un sistema operativo basado y compilado por voluntarios a partir del código fuente liberado por Red Hat Enterprise Linux RHEL.

3.3. Pentaho Business Intelligence (BI)

Pentaho Business Intelligence (BI) es una suite de herramientas de código libre que sirve para generar Inteligencia de Negocios (BI) y que incluye una serie de módulos integrados enfocados a crear y entregar soluciones en el desarrollo de un DWH.

FIGURA 3.3. Logo de Pentaho

3.3.1. Descripción

Pentaho es una plataforma de Inteligencia de Negocios (BI), que está formado por un grupo de programas desarrollados en Java de código abierto, flexible y muy potente, asegurando de esta forma la escalabilidad, integración y portabilidad. Estos programas trabajan en conjunto para crear y brindar soluciones a la construcción de un DWH, este grupo de programas también denominados módulos, cubren amplias necesidades o requerimientos que son esenciales en la construcción de un DWH. [11]

Gracias a Pentaho que reúne todo este conjunto de módulos, permite a las organizaciones acceder, integrar, visualizar y exportar los datos a fines que sean de gran interés para ellas y con ello brindar un soporte en la toma de decisiones que estará basado en información útil y confiable que impactará positivamente en el rendimiento y competitividad de la o las organizaciones.

Pentaho es capaz de ofrecer una alternativa de Software Libre, ya que supera a otras soluciones de Software Privado orientadas a la Inteligencia de Negocios *(*Business Objects, Cognos, Microstrategy, Microsoft, etc*),* una buena razón de ello es que es multiplataforma, flexible, robusto y tiene simplicidad de implantación. También permite acceder, explorar, dar formato y distribuir fácilmente la información a empleados, clientes y asociados.

Pentaho provee acceso a diversas fuentes de datos relacionales, OLAP o basadas en XML, además de ofrecer varios formatos de salida como PDF, HTML, Excel o hasta Texto plano. También permite llevar esta información a los usuarios finales vía web, e-mail, portales corporativos o aplicaciones propias. [11]

Pentaho ofrece dos versiones de su solución:

- **Community Edition:** Es una versión comunitaria gratuita orientada principalmente al mundo académico.

- **Enterprise Edition**: Es una versión privada comercial orientada a la implementación profesional tanto en empresas privadas como en instituciones gubernamentales u otras, que pretendan potenciar sus capacidades analíticas para mejorar su gestión.

En la tabla siguiente se da una descripción detallada sobre las versiones de Pentaho existentes.

Software y Servicios	Community Edition	Enterprise Edition
Informes	✓	✓
Análisis	✓	✓
Dashboards	✓	✓
Integración de datos / ETL	✓	✓
Plataforma Business Intelligence	✓	✓
Data Mining	✓	✓
Interacción con foros de la comunidad	✓	✓
Ambiente de desarrollo unificado para Agile BI	✓	✓
Documentación Web comunitaria (wiki)	✓	✓
Soporte profesional		
• Soporte telefónico	✗	✓
• Soporte a través de e-mail	✗	✓
• Acuerdo de nivel de servicios (SLA)	✗	✓
• Ilimitados casos de soporte	✗	✓
Mantenimiento del software		
• Mantenimiento del software	Por la comunidad	Por ingenieros Pentaho
• Acceso a parches	✗	✓
• Fixes incluidos en futuras versiones	✗	✓
Funcionalidades mejoradas		
• Analizador de Pentaho	✗	✓
• Formateo condicional	✗	✓
• Navegación en profundidad de detalle	✗	✓
• Arrastre y suelte los usuarios finales el Diseñador de paneles.	✗	✓
• Pentaho Enterprise Consola	✗	✓
• Single Sign-On	✗	✓
• Configuración de seguridad simplificada	✗	✓
• Diagnóstico de la aplicación	✗	✓
• Herramientas del repositorio	✗	✓

Software y Servicios	Community Edition	Enterprise Edition
• Administración del ciclo de vida	☒	☑
• Reportes de auditoría	☒	☑
• Expiración de contenido automática	☒	☑
• Clustering	☒	☑
• Monitoreo de desempeño	☒	☑
• Administración y monitoreo de ETL	☒	☑
• Seguridad avanzada	☒	☑
• Administración de usuarios y roles controlando las acciones de usuarios y roles	☒	☑
• Seguridad de permisos al contenido (dueño, crear, leer, actualizar, borrar)	☒	☑
• Seguridad del repositorio mejorada	☒	☑
• Administración de contenido mejorada	☒	☑
• Administración y programación de tareas remota	☒	☑
• Repositorio del equipo de desarrollo incluyendo control de versiones y bloqueo	☒	☑
• BI suite para Hadoop	☒	☑ (Add-on)
Especialización del Producto	☒	☑
Documentación profesional	☒	☑
Base de conocimiento	☒	☑
Soporte a consultas	☒	☑
Paquetes de asistencia remota	☒	☑
Paquetes de Instalación/configuración	☒	☑
Paquetes de diseño e integración	☒	☑
Paquetes de resolución de problemas y optimización	☒	☑
Foro en línea Enterprise Edition	☒	☑
Entrenamiento basado en Web	☒	☑

TABLA 3.3.1. Servicios de Community Edition y Enterprise Edition [Pentaho open source business intelligence, 2010, a]

Al estudiar esta tabla podemos observar que para el diseño y la implementación de nuestro DWH es suficiente con elegir la versión de Pentaho Community Edition, porque cuenta con los servicios de análisis, informes, cubos OLAP y la integración de datos (ETL), que son las funcionalidades necesarias para el correcto desarrollo de proyectos de BI.

3.3.2. Arquitectura

La arquitectura de Pentaho incluye una serie de componentes que trabajan en conjunto brindando así una Plataforma de Inteligencia de Negocios "orientada a la solución" y "centrada en procesos". [23]

La FIGURA 3.3.2. muestra el esquema de la arquitectura de Pentaho:

FIGURA 3.3.2. Esquema de la arquitectura de Pentaho

Pentaho es una arquitectura que define los procesos y bloques de construcción, pero no te obliga a utilizar toda la pila de ellos. La ventaja que tiene Pentaho es que el usuario puede construir su estructura libremente y realizar diferentes mezclas con respecto a todos sus componentes.

3.3.3. Módulos de Pentaho BI

La suite de Pentaho incluye todos los principales componentes requeridos para implementar soluciones de Inteligencia de Negocios. Algunos de los módulos tienen funciones básicas, como la autenticación de usuario o la administración de conexiones de bases de datos y el desarrollo de un DHW. Mientras que otro módulo es el que se encarga de ser la fachada para el análisis y exploración de los datos (reportes ad-hoc, tablas y graficas) del mismo.

Los módulos principales que integran la plataforma Pentaho BI son:

-  **Pentaho Analysis:** es un sistema de análisis multidimensional avanzado que sirve para ganar perspicacia y entender lo necesario para tomar óptimas decisiones. Es proporcionada por el servidor OLAP Mondrian[9] y la biblioteca JPivot para la navegación y la exploración [12]. El usuario puede ir ajustando la visión de los datos, los filtros de visualización, añadiendo o quitando los campos de agregación.

-  **Pentaho Reporting:** es el motor de informes que permite diseñar, crear y distribuir reportes en varios formatos conocidos (HTML, PDF, etc) a partir de diferentes tipos de fuentes. Los reportes o informes creados en Pentaho BI se basan principalmente en la biblioteca JFreeReport, pero es posible integrar los informes creados con las bibliotecas de informes externos, tales como informes de Jasper o BIRT [12]. Cuenta con una herramienta para realizar informes (Pentaho

[9] Es un servidor OLAP open source que gestiona comunicación entre una aplicación OLAP (escrita en Java) y la base de datos con los datos fuente. Actúa como "JDBC para OLAP"

Report Designer), con un motor de ejecución y una herramienta de Metadata para la realización de informes Ad-hoc, hecha para desarrolladores.

-  **Pentaho Dashboards:** esta solución provee a los usuarios de los negocios información crítica que necesitan para entender y mejorar el rendimiento organizacional. La herramienta utiliza para monitorear y analizar, indicadores clave de rendimientos (KPIs). También ejecuta el seguimiento de excepciones, permitiendo preestablecer alertas basadas en reglas del negocio. [12]

-  **Pentaho Data Integration:** se utiliza para integrar la información dispersa de diferentes fuentes (aplicaciones, bases de datos, archivos) y hacer que la información integrada quede a disposición del usuario final. Se realiza con una herramienta Kettle ETL (Pentaho Data Integration) que permite implementar los procesos ETL [12]. Kettle incluye herramientas como:

  - **SPOON:** permite diseñar de manera gráfica los procesos ETL.
  - **PAN:** sirve para ejecutar las transformaciones diseñadas con spoon en modo consola.
  - **CHEF:** permite diseñar los trabajos (integrado a SPOON) que ejecutarán a las trasformaciones.
  - **KITCHEN:** sirve para ejecutar los trabajos en modo consola.

-  **Data Mining:** la minería de datos se está ejecutando a través de algoritmos de datos con el fin de entender el negocio y hacer análisis predictivos. La minería de datos es posible gracias al **Proyecto de Weka** [12]**.**

El proyecto Weka (Waikato Enviroment for Knowledge Analysis) es un conjunto de librerías Java para la extracción de conocimientos desde bases de datos. Es un software que ha sido desarrollado bajo licencia GPL, e incluye las siguientes características:

- Diferentes fuentes de datos.
- Interfaz visual basada en procesos o flujos de datos.
- Distintas herramientas de minería de datos:
  - Reglas de asociación
  - Agrupación / Segmentación / Conglomerado
  - Clasificación (redes neuronales, reglas y árboles de decisión, aprendizaje bayesiano)
  - Regresión (regresión lineal)
  - Manipulación de datos
  - Combinación de modelos
  - Entornos de experimento

3.3.3.1. Elección de Módulos de Pentaho BI

Una vez conceptualizados los módulos que integran la suite de Pentaho, se determina que tales módulos pueden ser usados de manera independiente o conjuntamente, es decir, la elección de los módulos está determinada en base a las necesidades o requerimientos que se tengan en un proyecto de DWH. Dado nuestro caso de estudio, hemos decidido usar los siguientes:

- Módulo de Pentaho Analysis: porque nos provee todo lo necesario para la exploración y acceso a la información del DWH; para ello se hará uso de dos de sus herramientas, Pentaho BI Server Community Edition 4.8 y Pentaho Schema Workbench Community Edition 3.5.

- Módulo de Pentaho Data Integration: porque nos otorga todo lo necesario para la construcción de los procesos ETL que alimentarán el DWH; para ello se hará uso de su herramienta, Pentaho Data Integration (Kettle) Community Edition 4.4.

3.3.4. Ventajas de Pentaho BI

Pentaho BI es una potente y novedosa suite de desarrollo de Inteligencia de Negocios, sus herramientas se adaptan a distintos entornos de ejecución, es decir, es multiplataforma y aparte puede acceder a un gran número de sistemas de archivos, bases de datos, ficheros, sistemas, correos electrónicos, web services, etc., permitiendo así una fácil coexistencia y migración entre plataformas de grandes volúmenes de datos (soporte de escalabilidad).

El desarrollo de ETL es gráfico, simple, fácil de usar, fácil de mantener y sobre todo cuenta con una amplia gama de herramientas que bien sirven al momento de realizar transformaciones sobre los datos. [11]

Provee de un sistema web robusto y estable (BI Server Pentaho Community Edition) para el análisis, la exploración de datos, para la elaboración de reportes ad-hoc (reportes personalizados en tiempo real) y reportes prediseñados. Brindando también acceso desde cualquier ubicación, siempre y cuando se tenga una conexión a red. Además de que es un sistema multi-usuario, multi-idioma, mantiene seguridad basada en roles de acceso, buen rendimiento, distribuido y escalable. [11]

Gracias a este conjunto de herramientas hacen que sea una suite completa para el desarrollo de un DWH y las ventajas que han sido mencionadas han postulado a Pentaho BI desde sus comienzos en 2004, como una de las mejores herramientas de Software Libre para el desarrollo de Inteligencia de Negocios.

3.3.5. Complemento de Pentaho (Saiku Analytics)

Saiku Analytics forma parte de Pentaho BI Suite Community Edition como un complemento o plug-in, es una aplicación web de código libre que proporciona un excelente visor de consulta y análisis OLAP, la cual permite operaciones como:

- Selección de cubos.
- Selección de dimensiones y medidas.
- Representación y navegación mediante tablas dinámicas.
- Desglose y agrupamiento.
- Filtrado.
- Soporte de Consultas MDX.
- Representación de gráficas.
- Exportación a PDF, CSV y Excel.
- Almacenamiento y mantenimiento de repositorios de consultas.

Saiku es el nuevo nombre que se le dio a PAT (Pentaho Analysis Tool). PAT estaba destinada a ser el visualizador OLAP de la versión community de Pentaho, reemplazando a JPivot, una librería de componentes JSP que se utiliza para construir tablas OLAP generadas de forma dinámica. PAT había llegado hasta la versión 0.8, pero el equipo de Saiku decidió reescribir la interfaz visual, orientándola fuertemente a las nuevas tecnologías web 2.0 (ajax, jquery, etc.). [17]

Lo cierto es que Saiku es un navegador OLAP muy liviano e intuitivo, que saca provecho de muchos recursos que mejoran la experiencia del usuario, cosas tan simples como técnicas de *Drag & Drop* para el armado de las consultas. Al igual que Pentaho, Saiku también presenta una arquitectura modular, lo que permita usar Saiku de manera independiente, o embeberlo en nuestras aplicaciones. [17]

La versión 2.5 de Saiku, destaca la opción de filtrar y acotar los valores de las dimensiones (están basados en sentencias MDX) o las opciones contextuales de la propia tabla de resultados que permiten incluir nuevos niveles de las jerarquías de forma muy cómoda.

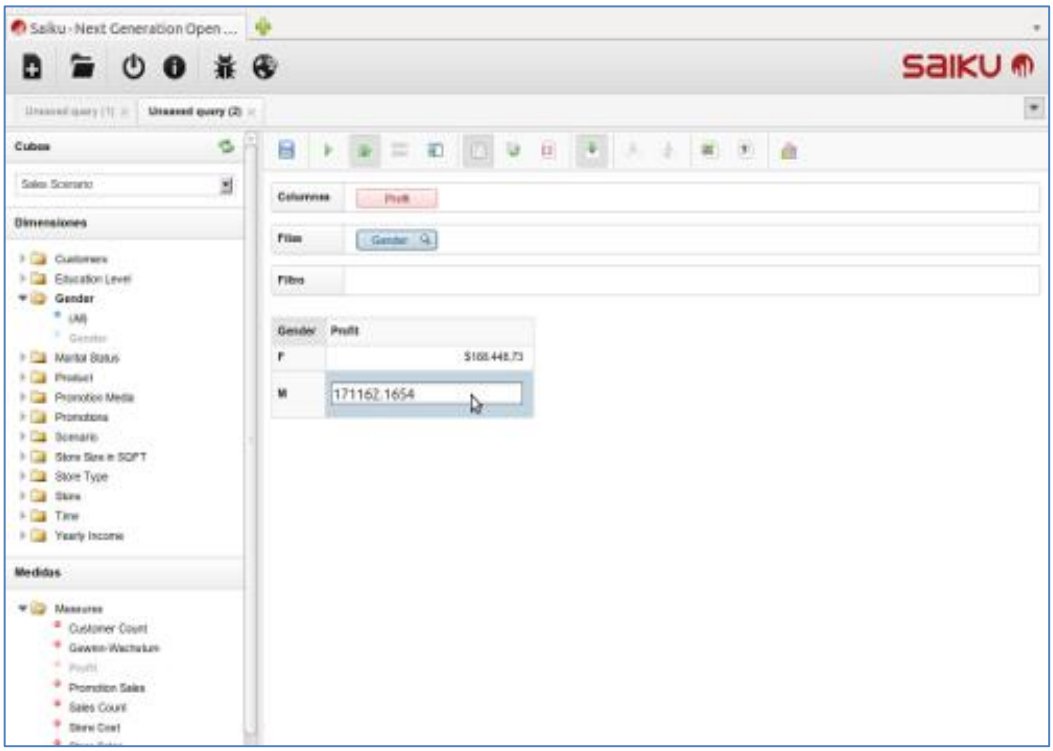


FIGURA 3.3.5. Saiku

3.4. PostgreSQL

PostgreSQL es un Sistema Gestor de Bases de Datos relacional orientado a objetos y libre, su desarrollo no es elaborado por una empresa y/o persona, sino que es dirigido por una comunidad de desarrolladores (PGDG[10]) que trabajan de forma desinteresada, altruista, libre y/o apoyada por organizaciones comerciales. Con cerca de dos décadas de desarrollo tras él, PostgreSQL es el gestor de bases de datos de código abierto más avanzado hoy en día, ofreciendo control de concurrencia multi-versión, soportando casi toda la sintaxis SQL (incluyendo subconsultas, transacciones, tipos y funciones definidas por el usuario), contando también con un amplio conjunto de enlaces con lenguajes de programación (incluyendo C, C++, Java, Perl, TCL y Python). [18]

PostgreSQL ha tenido una larga evolución, la cual se inicia en 1982 con el proyecto Ingres en la Universidad de Berkeley. Este proyecto, liderado por Michael Stonebraker, fue uno de los primeros intentos en implementar un motor de base de datos relacional. Después de haber trabajado un largo tiempo en *Ingres* y de haber tenido una experiencia comercial con el mismo, Michael decidió volver a la Universidad en 1985 para trabajar en un nuevo proyecto sobre la experiencia de Ingres, dicho proyecto fue llamado post-ingres o simplemente POSTGRES. [18]

Después de que el proyecto POSTGRES terminara, dos graduados de la universidad, Andrew Yu y Jolly Chen, comenzaron a trabajar sobre el código de POSTGRES, esto fue posible dado que POSTGRES estaba licenciado bajo la BSD, y lo primero que hicieron fue añadir soporte para el lenguaje SQL a POSTGRES, dado que anteriormente contaba con un intérprete del lenguaje de consultas QUEL (basado en Ingres), creando así el sistema al cual denominaron Postgres95. [18]

[10] PGDD (PostgreSQL Global Development Group) es una comunidad de desarrolladores

Para el año 1996 se unieron al proyecto personas ajenas a la Universidad como Marc Fournier de Hub.Org Networking Services, Bruce Momjian y Vadim B. Mikheev quienes proporcionaron el primer servidor de desarrollo no universitario para el esfuerzo de desarrollo de código abierto y comenzaron a trabajar para estabilizar el código de Postgres95. [18]

En el año 1996 decidieron cambiar el nombre de Postgres95 de tal modo que refleje la característica del lenguaje SQL y lo terminaron llamando PostgreSQL, cuya primera versión de código abierto fue lanzada el 1 de agosto de 1996. La primera versión formal de PostgreSQL (6.0) fué liberada en enero de 1997. Desde entonces, muchos desarrolladores entusiastas de los motores de base de datos se unieron al proyecto, coordinaron vía Internet y entre todos comenzaron a incorporar muchas características al motor. [18]

A mediados de 2005 otras dos compañías anunciaron planes para comercializar PostgreSQL con énfasis en nichos separados de mercados. EnterpriseDB añadió funcionalidades que le permitían a las aplicaciones escritas para trabajar con Oracle ser más fáciles de ejecutar con PostgreSQL. Greenplum contribuyó mejoras directamente orientadas a aplicaciones de Data Warehouse e Inteligencia de negocios, incluyendo el proyecto BizGres. [18]

En agosto de 2007 EnterpriseDB anunció el Postgres Resource Center y EnterpriseDB Postgres, diseñados para ser una distribución completamente configurada de PostgreSQL incluyendo muchos módulos contribuidos y agregados. EnterpriseDB Postgres fue renombrado Postgres Plus en marzo de 2008. [18]

PostgreSQL puede funcionar en múltiples plataformas (en general, en todas las modernas basadas en Unix) y a partir de la versión 8.0, también en Windows de forma nativa. Para las versiones anteriores existen versiones binarias para este sistema operativo, pero no tienen respaldo oficial.

PostgreSQL se puede descargar de dos páginas de la comunidad y de EnterpriseDB sin embargo, se realiza a continuación una comparativa de ambas versiones.

		
Producto	PostgreSQL	Postgres Plus Advanced Server
Arquitectura	Modelo objeto-relacional	Modelo objeto-relacional
Sistema Operativo	Windows ☑ Mac OS X ☑ Linux ☑ UNIX ☑ BSD ☑	Windows ☑ Mac OS X ☑ Linux ☑ UNIX ☑ BSD ☑
Demo		Versión de prueba
Interfaz	GUI ☑ SQL ☑	GUI ☑ SQL ☑
Capacidades de base de datos	Union ☑ Intersect ☑ Inner joins ☑ Outer joins ☑ Inner selects ☑	Union ☑ Intersect ☑ Inner joins ☑ Outer joins ☑ Inner selects ☑

TABLA 3.4. PostgreSQL vs Postgres Plus Advanced Server

3.4.1. Características de PostgreSQL

La última serie de producción es la 9.3, sus características técnicas la hacen una de las bases de datos más potentes y robustas del mercado. Su desarrollo comenzó hace más de 16 años, y durante este tiempo, estabilidad, potencia, robustez, facilidad de administración e implementación de estándares han sido las características que más se han tenido en cuenta durante su desarrollo. PostgreSQL funciona muy bien con grandes cantidades de datos y una alta concurrencia de usuarios accediendo a la vez al sistema. [13]

A continuación se presentan las características más importantes y soportadas por PostgreSQL:

- Es una base de datos que cumple con el ACID[11].
- Integridad referencial.
- Soporte de Tablespaces.
- Soporte para transacciones distribuidas.
- Soporte de Juegos de caracteres internacionales.
- Regionalización por columna.
- Alta concurrencia, Multi-Version Concurrency Control (MVCC).
- Multiples métodos de autentificación
- Disponible para Linux y UNIX en todas sus variantes (AIX, BSD, HP-UX, SGI IRIX, Mac OS X, Solaris, Tru64) y Windows 32/64bit.
- Soporte de Llaves primarias (primary keys) y foráneas (foreign keys).
- Columnas auto-incrementales.
- Índices compuestos, únicos, parciales y funcionales en cualquiera de los métodos de almacenamiento disponibles, B-tree, R-tree, hash ó GiST.
- Subconsultas.

[11] ACID es un acrónimo de Atomicidad, Consistencia, Aislamiento y Durabilidad, representando un conjunto de características necesarias para que una seria de instrucciones pueda ser considerada una transacción.

- Consultas recursivas.
- Vistas (views)
- Disparadores (triggers) comunes, por columna, condicionales.
- Reglas (Rules).
- Herencia de tablas (Inheritance)
- Numerosos tipos de datos y posibilidad de definir nuevos tipos. Además de los tipos estándares en cualquier base de datos, tenemos disponibles, entre otros, tipos geométricos, de direcciones de red, de cadenas binarias, UUID, XML, matrices, etc.
- Soporta el almacenamiento de objetos binarios grandes (gráficos, videos, sonido, etc).
- Soporte de Vistas Materializadas a partir de la versión 9.3.

3.4.2. Ventajas y desventajas de PostgreSQL

Las principales ventajas de usar PostgreSQL se centran en ser un SGDB robusto, estable y ampliamente seguro, contando principalmente con integridad referencial, alto nivel de ejecución de transacciones, distribuido y escalable. Esto hace que PostgreSQL sea un serio candidato para realizar proyectos de DWH, porque cuenta con un extenso surtido de características (tablespace, particionamiento, vistas materializadas, subconsultas, etc) y que también te da la libertad de desarrollar nuevas funcionalidades bajo distintos lenguajes de programación, que a diferencia de otros SGDB no te lo permiten.

Por otro lado cuenta con algunas desventajas o limitantes, estas son expuestas en la siguiente tabla:

Límite	Valor
Máximo tamaño base de datos	Ilimitado (Depende de tu sistema de almacenamiento)
Máximo tamaño de tabla	32 TB
Máximo tamaño de fila	1.6 TB
Máximo tamaño de campo	1 GB
Máximo número de filas por tabla	Ilimitado
Máximo número de columnas por tabla	250 - 1600 (dependiendo del tipo)
Máximo número de índices por tabla	Ilimitado

TABLA 3.4.2. Limitantes de PostgreSQL [13]

3.4.3. Arquitectura de PostgreSQL

PostgreSQL utiliza un modelo cliente/servidor y usa multiprocesos en vez de multihilos para garantizar la estabilidad del sistema. Un fallo en uno de los procesos no afectará el resto y el sistema continuará funcionando. [13]

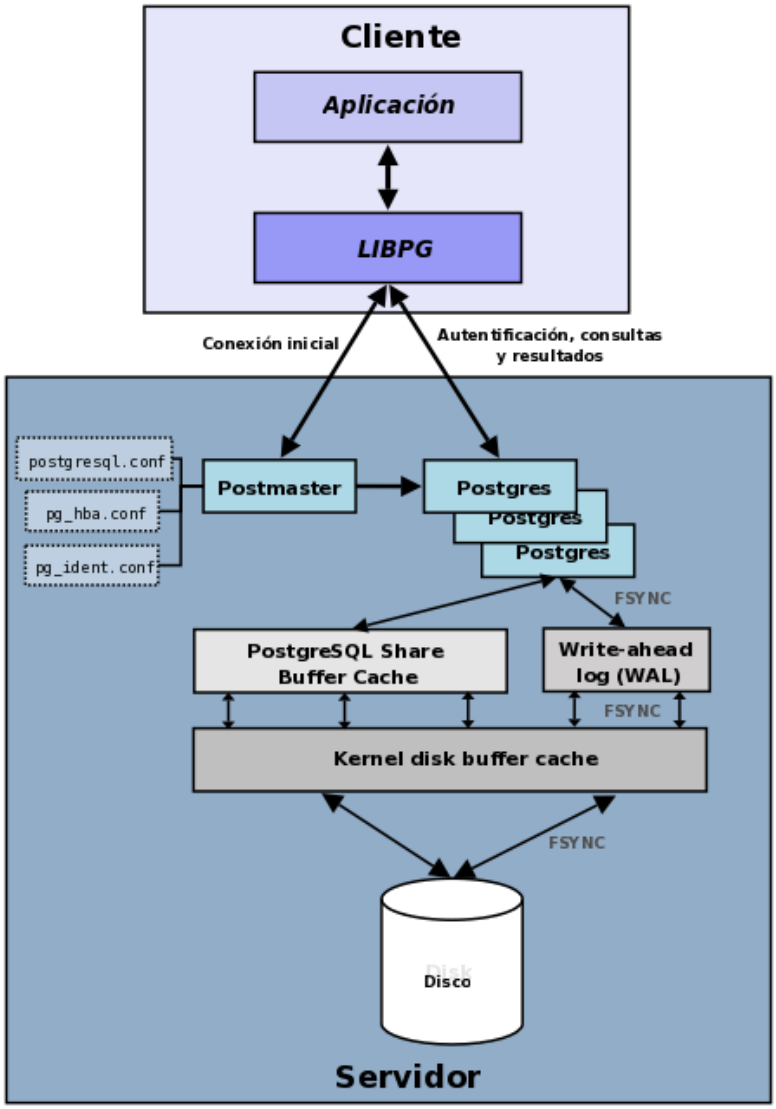


FIGURA 3.4.3. Esquema de arquitectura de PsotgreSQL

La FIGURA 3.4.3. ilustra de manera general los componentes más importantes en un sistema PostgreSQL. [13]

- Aplicación cliente: Esta es la aplicación cliente que utiliza PostgreSQL como administrador de bases de datos. La conexión puede ocurrir vía TCP/IP ó sockets locales. [13]

- Demonio postmaster: Este es el proceso principal de PostgreSQL. Es el encargado de escuchar por un puerto/socket por conexiones entrantes de clientes. También es el encargado de crear los procesos hijos que se encargarán de autentificar estas peticiones, gestionar las consultas y mandar los resultados a las aplicaciones clientes. [13]

- Ficheros de configuración: Los tres ficheros principales de configuración utilizados por PostgreSQL, postgresql.conf, pg_hba.conf y pg_ident.conf. [13]

- Procesos hijos postgres: Procesos hijos que se encargan de autentificar a los clientes, de gestionar las consultas y mandar los resultados a las aplicaciones clientes. [13]

- PostgreSQL share buffer cache: Memoria compartida usada por PostgreSQL para almacenar datos en caché. [13]

- Write-Ahead Log (WAL): Componente del sistema encargado de asegurar la integridad de los datos (recuperación de tipo REDO). [13]

- Kernel disk buffer cache: Caché de disco del sistema operativo. [13]

- Disco: Disco físico donde se almacenan los datos y toda la información necesaria para que PostgreSQL funcione. [13]

3.4.4. PostgreSQL vs. MySQL

En esta subtema hacemos la comparación de PostgresSQL contra MySQL, que también es un SGDB muy popular, estable, seguro y robusto en el mercado y que al igual que PostgreSQL mantiene algunas características enfocadas a la Inteligencia de Negocios.

MySQL es un producto desarrollado como Software Libre pero con licenciamiento dual, es decir, si se utiliza en productos privativos es necesario adquirir una licencia específica que permita el uso de este SGBD, sino este puede ser usado libremente. Una característica muy importante de MySQL es que nos permite seleccionar el tipo de mecanismo de almacenamiento el cual nos dará un rendimiento más óptimo en aplicaciones que se ajusten a tal mecanismo de almacenamiento. [19]

Aun así, en base a lo expuesto en el párrafo anterior MySQL cuenta con algunas características relacionadas con el uso en Data Warehousing en comparación con PostgreSQL. En la siguiente tabla se detallan específicamente las diferencias que tienen estos dos potentes SGDB en cuestión al Data Warehousing.

BI / Data Warehousing	PostgresSQL	MySQL 5.5 Enterprise	Comentarios
Índices Bitmap	☑	☒	
Cuadros resumen	✓	☒	
Funciones de agregado	☑	☑	
Funciones de ventana	☑	☒	
Expresiones de tabla comunes	☑	☒	

BI / Data Warehousing	Postgres Plus Adv. Server	MySQL 5.5 Enterprise	Comentarios
Expresión basada en los índices	☑	☒	
Vistas materializadas	Emulado con Procedimientos Almacenados	☒	
Las tablas externas		☑	
Unión	☑	☑	
Intersección	☑	☒	
Excepto	☑	☒	
Inner Joins	☑	☑	
Outer Joins	☑	☑	
Inner Selects	☑	☑	
Merge Joins	☑	☑	
Consulta en paralelo	☑	☑	Con GridSQL para Postgres y el almacenamiento de terceros proveedores de motores para MySQL
Optimizador de estadísticas mgmt	☑	☑	
Establecer funciones que devuelven	☑	☑	Uso SETOF función
Ejemplo de consulta scan	☑	☑	

TABLA 3.4.4. Características de PostgreSQL y MySQL [20]

PostgreSQL contiene muchas características de base de datos críticos que se utiliza en aplicaciones de almacenamiento de datos, incluyendo GridSQL para la creación de una base de datos compartida distribuida capaz de establecer consultas en paralelo para el rendimiento más rápido. MySQL utiliza generalmente el Motor MyISAM almacenamiento, que es muy adecuado para pequeñas aplicaciones de almacenamiento de datos.

CAPÍTULO 4. Metodología para el desarrollo de un DWH

Existen muchas metodologías de diseño y construcción de DWH. Cada fabricante de software de inteligencia de negocios busca imponer una metodología con sus productos. Sin embargo, se imponen entre la mayoría dos metodologías, la de Kimball y la de Inmon.

Desde el punto de vista arquitectónico, la mayor diferencia entre los dos autores es el sentido de la construcción del DWH, esto es comenzando por los Data Marts o ascendente (Bottom-up, Kimball), o comenzando con todo el DWH desde el principio, o descendente (Top-Down, Inmon). Por otra parte, la metodología de Inmon se basa en conceptos bien conocidos del diseño de bases de datos relacionales; la metodología para la construcción de un sistema de este tipo es la habitual para construir un sistema de información, utilizando las herramientas habituales, al contrario de la de Kimball, que se basa en un modelado dimensional (no normalizado).

4.1. Planificación

La planificación es la fase más importante para la construcción del DWH ya que es donde se determinarán todos y cada uno de los pasos a seguir para su desarrollo e implementación. Primero se comienza por elegir una metodología que se acople a los requerimientos que tiene un proyecto de DWH. Existen distintas metodologías que surgen por parte de los fabricantes de software de Inteligencia de Negocios. Sin embargo, dos metodologías son las que destacan entre todas; la primera es la Top-Down (De arriba-abajo) hecha por Bill Inmon y la segunda es Bottom-Up (De abajo-arriba) de Ralph Kimball. [21]

Partiendo de los requerimientos y alcances que contiene nuestro caso de estudio, es más conveniente usar una metodología que proporcione un enfoque de menor a mayor, es decir, que pueda ir arrojando resultados de calidad a corto plazo y pueda ir manteniendo escalabilidad a medida que el proyecto vaya creciendo (existan nuevas fuentes de información) o se definan nuevos requerimientos. Para tales necesidades la metodología que más se acomoda es la Bottom-Up de Ralph Kimball. Porque es aquella que comienza con el diseño de Data Marts que van poco a poco cubriendo necesidades específicas por área, contexto de estudio o fuente de información.

El uso de la metodología Bottom-Up se basa en lo que Kimball denomina Ciclo de Vida Dimensional del Negocio (Business Dimensional Lifecycle). Este ciclo de vida del proyecto de DWH, está basado en cuatro principios básicos:

- Centrarse en el caso de estudio: Hay que concentrarse en la identificación de los requerimientos del caso de estudio y su valor asociado, así se podrá usar estos esfuerzos para desarrollar relaciones sólidas con el caso de estudio, agudizando el análisis del mismo y la competencia consultiva de los implementadores. [22]

- Construir una infraestructura de información adecuada: Diseñar una base de información única, integrada, fácil de usar, de alto rendimiento donde se reflejará la amplia gama de requerimientos del caso de estudio identificados en la organización. [22]

- Realizar entregas en incrementos significativos: Crear el almacén de datos (DWH) en incrementos a corto plazo. Hay que definir prioridades por cada elemento identificado para así determinar el orden de aplicación de los incrementos. [22]

- Ofrecer una solución completa: Proporcionar todos los elementos necesarios para entregar un DWH de calidad a los usuarios finales. Es decir, mantener un DWH sólido, bien diseñado, con calidad y accesible. Esto incluye entregar herramientas de exploración, análisis de información (consultas ad-hoc), soporte y documentación. [22]

La construcción de un DWH suele ser una actividad compleja, pero Kimball nos propone una metodología que nos ayuda a simplificar esa complejidad. Las tareas de esta metodología (Ciclo de Vida Dimensional del Negocio) se muestran en la FIGURA 4.1.a.

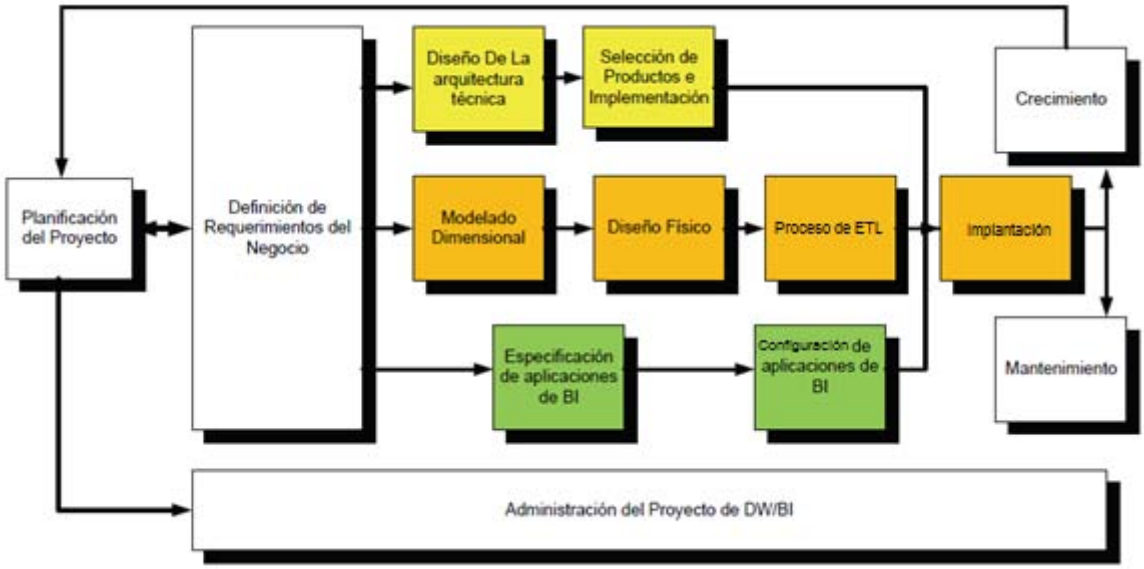


FIGURA 4.1.a. Tareas de la metodología Kimball denominada Ciclo de Vida Dimensional del Negocio

Una vez ya elegida la metodología de Kimball (Ciclo de Vida Dimensional del Negocio) esta misma comienza con la planeación del proyecto. Durante esta etapa se realizarán los siguientes pasos:

1. Definir el Alcance del proyecto

Definir el contexto del proyecto, permite precisar cuál es el alcance del desarrollo, es decir, que vamos a realizar y que no. Es necesario que el contexto sea significativo en términos del valor para la organización o institución. El alcance de un proyecto debe ser realizado mediante un documento en el que se empieza a definir el trabajo y todas sus entregas clave. [Ver Anexo V] [21]

2. Definir el equipo de trabajo

Es esencial involucrar a todo un conjunto de personas, considerando desde los usuarios hasta el equipo de desarrollo, con el fin de asegurar que el DWH contenga información que satisfaga los requerimientos. [21]

Construir un DWH es un proyecto que requiere la integración de un personal muy cualificado, que debe cubrir con los siguientes roles:

- ***Administrador del proyecto:*** Es el que se encarga del manejo de los tiempos, fases, recursos y la productividad del equipo de trabajo. [21]

- ***Analistas:*** Es un grupo de personas quienes se encargan de recopilar y analizar la información obtenida de la empresa, la cual es necesaria para el diseño posterior del DWH y de los DMs. [21]

- ***Diseñadores:*** Es el grupo de personas encargadas de diseñar la base de datos del DWH y del proceso de extracción, transformación y carga de datos (ETL) de acuerdo a los requerimientos que han sido recopilados y analizados anteriormente. [21]

- ***Implementadores:*** Son los encargados de traducir las especificaciones de diseño en: tablas de hechos y dimensiones, paquetes de extracción, transformación, carga (ETL), cubos y reportes multidimensionales. [21]

- ***Personal de Pruebas:*** Grupo de personas quienes llevaran a cabo el plan de pruebas diseñado para el proyecto con el fin de comprobar el cumplimiento de las especificaciones funcionales determinadas para la creación del DWH.

- ***Usuarios clave:*** Persona o grupo de personas encargadas de la coordinación general del proyecto por parte de la organización, asegurándose el cumplimiento de los requerimientos del proyecto para el beneficio y los intereses de la organización. [21]

FIGURA 4.1.b. Organigrama de usuarios IISUE (Usuarios clave)

3. Selección de una estrategia para la implementación

El seleccionar una estrategia para la implementación de un DWH suele ser una forma importante para el éxito del mismo. Es aquí en donde se elegirá el ámbito de implementación, en la mayoría de las organizaciones se empieza con una implementación que produzca beneficios rápidos a un determinado grupo de usuarios, después de definir un rumbo general y un conjunto general de objetivos. [21]

El ámbito se puede restringir teniendo en cuenta:

- Presupuesto.
- Número de fuentes de datos seleccionadas.
- Número de departamentos o áreas atendidas.
- Tiempo asignado al proyecto.

Tomando en cuenta lo anterior y en base a nuestro caso de estudio la mejor estrategia será aquella que pueda ir arrojando resultados incrementales a corto plazo (Metodología de Kimball) y que no requiera de un costo inicial (software libre), es decir, implantar el DWH sobre un ámbito piloto con una alta probabilidad de éxito.

4. Elaboración del Plan de Proyecto

En este punto es importante definir todas las tareas necesarias para el desarrollo del DWH, junto con sus tiempos y precedencias. El uso de un plan de proyecto nos permite llevar un control del avance en el que se encuentra nuestro proyecto, y por tanto nos brinda una brújula de seguimiento dentro del desarrollo de un proyecto de DWH, ya que sin ello, el desarrollo del mismo sería un desastre, debido a que no se tendría un orden especifico de las tareas realizadas o que vayan a ser realizadas. [Ver Anexo VI] [21]

4.2. Definición de requerimientos

La fase de requerimientos de un Data Warehouse es también un factor esencial que determina el éxito de éste, porque es en donde se especifican las funciones y características que tendrá el DWH. [22]

El volumen de recolección de información para los requerimientos debe de ser de manera adecuada y debe ser basada en constantes reuniones con los diferentes usuarios que están involucrados en el proyecto, para que den sus diferentes puntos de vista y así lograr que el Data Warehouse cumpla con las necesidades del instituto.

En este proyecto se dieron constantes reuniones con los distintos usuarios vinculados en la elaboración de reportes del IISUE y se lograron establecer varios requerimientos que permiten la construcción de un DWH.

La idea general es desarrollar un proyecto de DWH, con la finalidad de obtener el máximo provecho a la información proveniente del SIAH. Para ello es necesario disponer de toda la información referente a los académicos en un único almacén de datos y éste permita generar consultas ad-hoc con sus respectivas gráficas. También contará con la navegación dimensional sobre los datos, es decir, se podrá analizar la información desde distintos niveles de detalle.

Se definieron los siguientes requerimientos en base a las necesidades más importantes del IISUE:

1. El IISUE tiene la necesidad de saber el total de académicos en base al tipo de académico (Investigador, Profesor o Técnico académico) anualmente.
2. El IISUE necesita saber el total de académicos en base al nivel de estudios que poseen sus académicos anualmente.
3. El IISUE solicitó saber el total de académicos en base al departamento en el que están adscritos y al tipo de nacionalidad que tienen.
4. El IISUE tiene la necesidad de saber los tipos de estímulos (SNI, PRIDE, FOMDOC, PAIPA) que se les brindan a sus académicos.
5. El IISUE pretende conocer el total de asignaturas impartidas por sus académicos, en base al grado de estudio impartido y a la entidad adscripta.
6. El IISUE pidió conocer el total de tesis y tesinas dirigidas por sus académicos, en base al nivel de estudio impartido y a la entidad adscripta.
7. El IISUE solicitó conocer el total de informes emitidos por sus académicos, en base al nivel de estudio impartido y a la entidad adscripta.
8. El IISUE requiere la información de sus diferentes tipos de proyectos (colectivo e individual) por cada año.
9. El IISUE solicitó conocer el total de proyecto en base a su estatus (en proceso, reiniciado, suspendido y terminado), por año.
10. El IISUE necesita conocer el total de proyectos en base al tipo de participación de cada uno de sus académicos, por año.

Para lograr estos requerimientos fue necesario tener constantes entrevistas con el personal administrativo y explorar todas sus fuentes de información. Una vez definidos varios requerimientos de suma importancia para el IISUE, podemos ahora si comenzar a hacer el análisis e ir apoyándonos de una herramienta llamada Matriz de Procesos y Dimensiones. [Ver Anexo VII]

La Matriz de Procesos y Dimensiones es una herramienta útil para el análisis de Dimensiones que contendrá el DWH, en ella se pueden ir marcando las posibles Dimensiones que corresponderán a los Procesos de Negocio identificados (Data Marts) todo en base a los requerimientos y a la información que se tiene.

Esta matriz tiene en sus filas los Procesos de Negocio identificados, y en las columnas, las Dimensiones identificadas. Para iniciar tenemos que recolectar todos los Orígenes de Datos, en este caso el Origen de los Datos sería la Base de Datos denominada SIAH y demás documentos tangibles que nos fueron expuestos (Encuestas, Reportes en PDF, Hojas de cálculo y Archivos de texto). Una vez recolectados seleccionamos la información que nos será útil en este proyecto, quedando como único Origen de Datos la Base de Datos SIAH del periodo 2011. Este origen de información aunque es uno sólo es bastante bueno porque en él esta capturada toda la información relacionada al Personal Académico del IISUE y además reúne la información que el IISUE va cotejando manualmente en Hojas de cálculo y Archivos de texto.

FIGURA 4.2.a. Origen de Datos (SIAH_2011_IISUE.mdb)

El Diagrama de datos Entidad - Relación de la Base de Datos SIAH está acotado de tal manera que se seleccionaron las Tablas que muy probablemente cubrirán con los requerimientos que se han definido y se presenta a continuación:

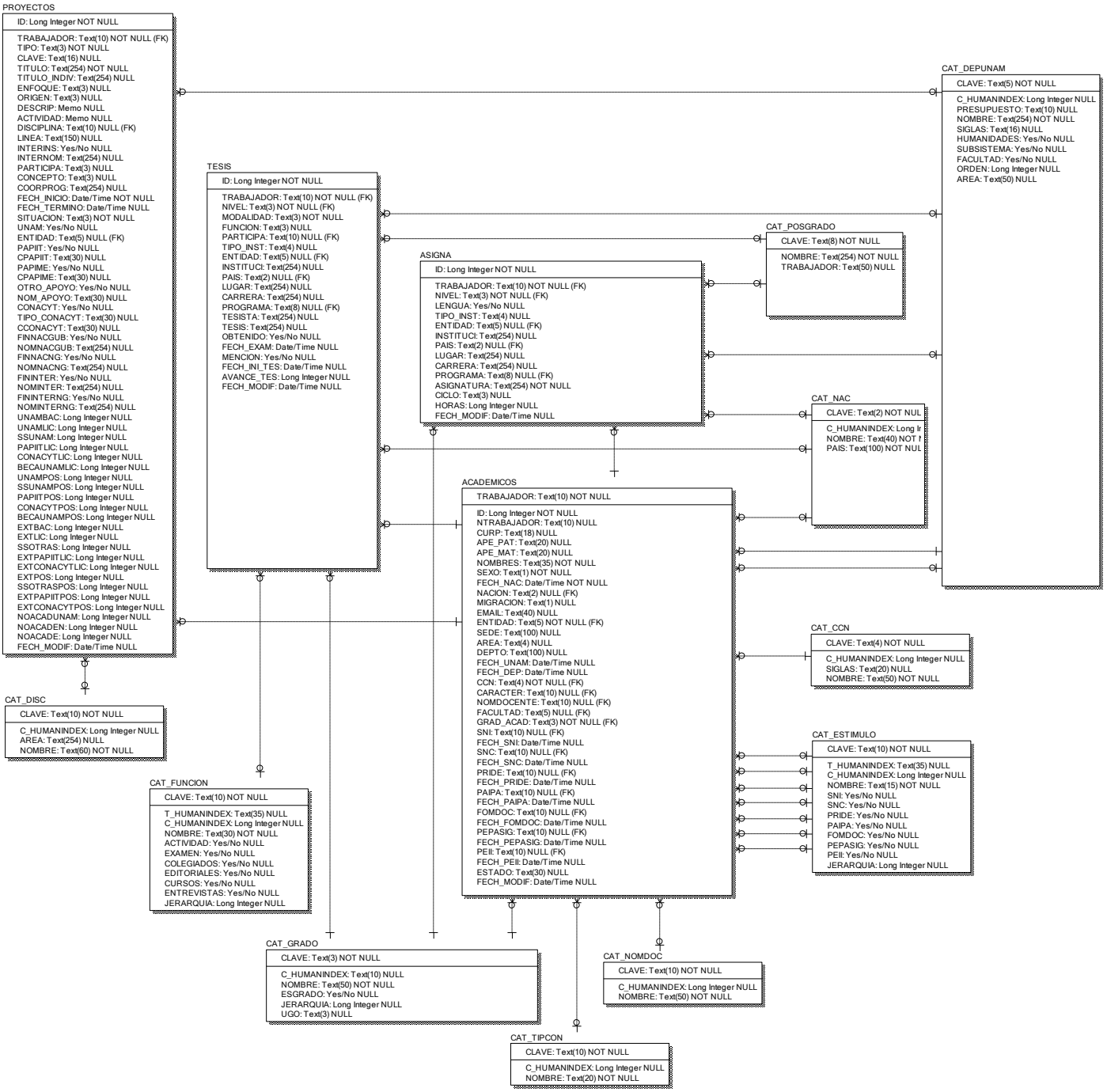


FIGURA 4.2.b. Diagrama Entidad - Relación (SIAH)

También debemos contar con el Diccionario de Datos[12] de cada una de las Tablas previamente seleccionadas.

Tabla – ACADEMICOS

Nombre	Tipo	Tamaño	Nulo	Descripción	Valores
ID	Entero largo	4	No	Identificador único	
TRABAJADOR	Texto	10	No	Número de trabajador, con este campo se relacionan las tablas del informe de actividades.	
NTRABAJADOR	Texto	10	Sí	Número de trabajador	
CURP	Texto	18	Sí	Clave única de registro de población	
APE_PAT	Texto	20	Sí	Apellido paterno	
APE_MAT	Texto	20	Sí	Apellido materno	
NOMBRES	Texto	35	No	Nombres	
SEXO	Texto	1	No	Género	"M", "F"
FECH_NAC	Fecha/Hora	8	No	Fecha de nacimiento	
NACION	Texto	2	Sí	Nacionalidad	Catálogo de nacionalidades (CAT_NAC)
MIGRACION	Entero	1	Sí	Calidad migratoria	1="Inmigrante", 2="No inmigrante", 5="Inmigrado"
EMAIL	Texto	40	Sí	Dirección de correo electrónico	
ENTIDAD	Texto	5	No	Dependencia de adscripción	Catálogo de dependencias (CAT_DEPUNAM)
SEDE	Texto	100	Sí	Nombre de la cede, si existe	
AREA	Texto	4	Sí	Área de trabajo de los técnicos académicos	BIBL=Biblioteca, CMPU=Cómputo y sistemas, PUBL=Publicaciones, OTRA=Otra
DEPTO	Texto	100	Sí	Departamento dentro de la entidad	
FECH_UNAM	Fecha/Hora	8	Sí	Fecha de ingreso a la UNAM	
FECH_DEP	Fecha/Hora	8	Sí	Fecha de ingreso a la dependencia	
CCN	Texto	4	No	Clase, categoría y nivel	Catálogo de clases, categorías y niveles (CAT_CCN)
CARÁCTER	Texto	10	Sí	Carácter de contratación	Catálogo de tipos de contrataciones (CAT_TIPCON)
NOMDOCENTE	Texto	10	Sí	Nombramiento docente en la UNAM	Catálogo de nombramientos docentes (CAT_NOMDOC)
FACULTAD	Texto	5	Sí	Facultad de adscripción	Catálogo de dependencias (CAT_DEPUNAM) filtrado por "FACULTAD=True"
GRAD_ACAD	Texto	3	No	Grado académico	Catálogo de grados académicos filtrado por "ESGRADO=True"
SNI	Texto	10	Sí	Nivel en el SNI	Catálogo de estímulos (CAT_ESTIMULO) filtrado por "SNI=True"
FECH_SNI	Fecha/Hora	8	Sí	Fecha de obtención o ratificación de nivel en el SNI	
SNC	Texto	10	Sí	Nivel en el SNC	Catálogo de estímulos (CAT_ESTIMULO) filtrado por "SNC=True"
FECH_SNC	Fecha/Hora	8	Sí	Fecha de obtención o ratificación de nivel en el SNC	
PRIDE	Texto	10	Sí	Nivel en el PRIDE	Catálogo de estímulos (CAT_ESTIMULO) filtrado por "PRIDE=True"
FECH_PRIDE	Fecha/Hora	8	Sí	Fecha de obtención o ratificación de nivel en el PRIDE	
PAIPA	Texto	10	Sí	Nivel en el PAIPA	Catálogo de estímulos (CAT_ESTIMULO) filtrado por "PAIPA=True"
FECH_PAIPA	Fecha/Hora	8	Sí	Fecha de obtención o ratificación de nivel en el PAIPA	
FOMDOC	Texto	10	Sí	Nivel en el FOMDOC	Catálogo de estímulos (CAT_ESTIMULO) filtrado por "FOMDOC=True"
FECH_FOMDOC	Fecha/Hora	8	Sí	Fecha de obtención o ratificación de nivel en el FOMDOC	
PEPASIG	Texto	10	Sí	Nivel en el PEPASIG	Catálogo de estímulos (CAT_ESTIMULO) filtrado por "PEPASIG=True"
FECH_PEPASIG	Fecha/Hora	8	Sí	Fecha de obtención o ratificación de nivel en el PEPASIG	
PEII	Texto	10	Sí	Nivel en el PEII	Catálogo de estímulos (CAT_ESTIMULO) filtrado por "PEII=True"
FECH_PEII	Fecha/Hora	8	Sí	Fecha de obtención o ratificación de nivel en el PEII	
ESTADO	Texto	30	Sí	Estado del académico en el subsistema	
FECH_MODIF	Fecha/Hora	8	Sí	Fecha de última modificación	

[12] Un diccionario de datos es un conjunto de metadatos que almacena información específica de cada Tabla (nombre, tipo de dato, tamaño, descripción y contenido) que es usada en un sistema.

Tabla - ASIGNA

Nombre	Tipo	Tamaño	Nulo	Descripción	Valores
ID	Entero largo	4	No	Identificador único	
TRABAJADOR	Texto	10	No	Número de trabajador	
NIVEL	Texto	3	No	Nivel de la asignatura	Catálogo de grados académicos (CAT_GRADO) sin filtrar
LENGUA	Sí/No	1	Sí	Es una asignatura de lengua	
TIPO_INST	Texto	4	Sí	Tipo de institución	"UNAM", "NAC", "EXT"
ENTIDAD	Texto	5	Sí	Nombre de la entidad, en su caso	Catálogo de dependencias (CAT_DEPUNAM)
INSTITUCI	Texto	254	Sí	Nombre de la institución, en su caso	
PAIS	Texto	2	Sí	País	Catálogo de paises (CAT_NAC)
LUGAR	Texto	254	Sí	Ciudad y estado, en caso de institución ajena a la UNAM	
CARRERA	Texto	254	Sí	Nombre de la carrera o programa educativo	
PROGRAMA	Texto	8	Sí	Programa de posgrado	Catálogo de programas de posgrado de la UNAM (CAT_POSGRADO)
ASIGNATURA	Texto	254	No	Nombre de la asignatura	
CICLO	Texto	3	Sí	Duración del ciclo: Trimestral, Semestral, Anual	"TRI", "SEM", "ANU"
HORAS	Entero	2	Sí	Horas a la semana	
FECH_MODIF	Fecha/Hora	8	Sí	Fecha de última modificación	

Tabla – TESIS

Nombre	Tipo	Tamaño	Nulo	Descripción	Valores
ID	Entero largo	4	No	Identificador único	
TRABAJADOR	Texto	10	No	Número de trabajador	
NIVEL	Texto	3	No	Nivel de la tesis	Catálogo de grados académicos (CAT_GRADO) sin filtrar
MODALIDAD	Texto	3	No	Modalidad: Tesis, tesina, candidatura, informe académico	"TES", "TEA", "CAN", "INF"
FUNCION	Texto	3	Sí	Función en la dirección de la tesis: Director, comité tutorial	"DIR", "TUT"
PARTICIPA	Texto	10	Sí	Participación en el jurado	Catálogo de funciones (CAT_FUNCION) filtrado por "EXAMEN=True"
TIPO_INST	Texto	4	Sí	Tipo de institución	"UNAM", "NAC", "EXT"
ENTIDAD	Texto	5	Sí	Nombre de la entidad, en su caso	Catálogo de dependencias (CAT_DEPUNAM)
INSTITUCI	Texto	254	Sí	Nombre de la institución, en su caso	
PAIS	Texto	2	Sí	País	Catálogo de países (CAT_NAC)
LUGAR	Texto	254	Sí	Lugar donde se realizó el exámen, en su caso	
CARRERA	Texto	254	Sí	Carrera	
PROGRAMA	Texto	8	Sí	Programa de posgrado	Catálogo de programas de posgrado (CAT_POSGRADO)
TESISTA	Texto	254	Sí	Nombres de los tesistas	
TESIS	Texto	254	Sí	Título de la tesis	
OBTENIDO	Sí/No	1	Sí	Grado obtenido	
FECH_EXAM	Fecha/Hora	8	Sí	Fecha en la que se realizó el exámen, en su caso	
MENCION	Sí/No	1	Sí	Mención honorífica	
FECH_INI_TES	Fecha/Hora	8	Sí	Fecha de inicio de la tesis, en su caso	
AVANCE_TES	Entero	2	Sí	Porcentaje de avance de la tesis, en su caso	
FECH_MODIF	Fecha/Hora	8	Sí	Fecha de última modificación	

Tabla – PROYECTOS

Nombre	Tipo	Tamaño	Nulo	Descripción	Valores
ID	Entero largo	4	No	Identificador único	
TRABAJADOR	Texto	10	No	Número de trabajador	
TIPO	Texto	3	No	Tipo de proyecto: Individual o Colectivo	"IND", "COL"
CLAVE	Texto	16	Sí	Clave de proyecto asignada por presupuesto	
TITULO	Texto	254	No	Título del proyecto	
TITULO_INDIV	Texto	254	Sí	Título individual, parte de un proyecto colectivo	
ENFOQUE	Texto	3	Sí	Disciplinario, Interdisciplinario, Multidisciplinario	"DIS", "INT", "MUL"
ORIGEN	Texto	3	Sí	Principal, Derivado, Secundario	"PRI", "DER", "SEC"
DESCRIP	Memo	-	Sí	Descripción del proyecto	
ACTIVIDAD	Memo	-	Sí	Resumen del actividades en el periodo	
DISCIPLINA	Texto	10	Sí	Disciplina	Catálogo de disciplinas (CAT_DISC)
LINEA	Texto	150	Sí	Línea de investigación	
INTERINS	Sí/No	1	Sí	Proyecto interinstitucional	
INTERNOM	Texto	254	Sí	Instituciones participantes, en su caso	
PARTICIPA	Texto	3	Sí	Participación: Responsable, Corresponsable, Participante, Becario	"RES","COR","PAR","BEC"
CONCEPTO	Texto	3	Sí	Concepto	
COORPROG	Texto	254	Sí	Coordinación de programas de investigación, solo para el CRIM	
FECH_INICIO	Fecha/Hora	8	No	Año en el que inició el proyecto	
FECH_TERMINO	Fecha/Hora	8	Sí	Año tentativo de término del proyecto	
SITUACION	Texto	3	No	Situación del proyecto: En proceso, Reiniciado, Terminado, Suspendido, Cancelado	"PRO", "REI", "TER", "SUS", "CAN"
UNAM	Sí/No		Sí	Financiamiento de la UNAM	
ENTIDAD	Texto	5	Sí	Nombre de la entidad que financia el proyecto, en su caso	Catálogo de dependencias (CAT_DEPUNAM)
PAPIIT	Sí/No	1	Sí	Financiamiento PAPIIT	
CPAPIIT	Texto	30	Sí	Clave PAPIIT	
PAPIME	Sí/No	1	Sí	Financiamiento PAPIME	
CPAPIME	Texto	30	Sí	Clave PAPIME	
OTRO_APOYO	Sí/No	1	Sí	Otro financiamiento	
NOM_APOYO	Texto	30	Sí	Nombre de otro financiamiento	
CONACYT	Sí/No	1	Sí	Financiamiento CONACYT	
TIPO_CONACYT	Texto	30	Sí	Tipo de financiamiento CONACYT	
CCONACYT	Texto	30	Sí	Clave CONACYT	CA=Ciencia Aplicada,RT=Redes Temáticas,RR=Repatriaciones,IR=Índice de Revistas Mexicanas de Investigación,EN=Estancias Sabáticas y Posdoctorales Nacionales,EE=Estancias Sabáticas y Posdoctorales al Extranjero,CT=Cooperación Tecnológica Bilateral,AV=AVANCE,CI=Cooperación Internacional,FO=FONCICYT
FINNACGUB	Sí/No	1	Sí	Financiamiento de institución nacional gubernamental	
NOMNACGUB	Texto	254	Sí	Nombre de la institución gubernamental, en su caso	
FINNACNG	Sí/No	1	Sí	Financiamiento de institución nacional no gubernamental	
NOMNACNG	Texto	254	Sí	Nombre de la institución gubernamental, en su caso	
FININTER	Sí/No	1	Sí	Financiamiento de institución internacional gubernamental	
NOMINTER	Texto	254	Sí	Nombre de la institución internacional gubernamental, en su caso	
FININTERNG	Sí/No	1	Sí	Financiamiento de institución internacional no gubernamental	
NOMINTERNG	Texto	254	Sí	Nombre de la institución internacional no gubernamental, en su caso	
UNAMBAC	Entero	2	Sí	Alumnos de bachillerato de la UNAM	
UNAMLIC	Entero	2	Sí	Alumnos de licenciatura de la UNAM	
SSUNAM	Entero	2	Sí	Alumnos de licenciatura de servicio social de la UNAM	
PAPIITLIC	Entero	2	Sí	Alumnos de licenciatura del PAPIIT	
CONACYTLIC	Entero	2	Sí	Alumnos de licenciatura del CONACYT	
BECAUNAMLIC	Entero	2	Sí	Alumnos de licenciatura con beca UNAM	
UNAMPOS	Entero	2	Sí	Alumnos de posgrado de la UNAM	
SSUNAMPOS	Entero	2	Sí	Alumnos de posgrado de servicio social de la UNAM	
PAPIITPOS	Entero	2	Sí	Alumnos de posgrado del PAPIIT	
CONACYTPOS	Entero	2	Sí	Alumnos de posgrado del CONACYT	
BECAUNAMPOS	Entero	2	Sí	Alumnos de posgrado con beca UNAM	
EXTBAC	Entero	2	Sí	Alumnos externos de bachillerato	
EXTLIC	Entero	2	Sí	Alumnos externos de licenciatura	
SSOTRAS	Entero	2	Sí	Alumnos externos de licenciatura de servicio social	
EXTPAPIITLIC	Entero	2	Sí	Alumnos externos de licenciatura de PAPIIT	
EXTCONACYTLIC	Entero	2	Sí	Alumnos externos de licenciatura de CONACYT	
EXTPOS	Entero	2	Sí	Alumnos externos de posgrado	
SSOTRASPOS	Entero	2	Sí	Alumnos externos de posgrado de servicio social	
EXTPAPIITPOS	Entero	2	Sí	Alumnos externos de posgrado de PAPIIT	
EXTCONACYTPOS	Entero	2	Sí	Alumnos externos de posgrado de CONACYT	
NOACADUNAM	Entero	2	Sí	Académicos de la UNAM	
NOACADEN	Entero	2	Sí	Académicos de instituciones extranjeras	
NOACADE	Entero	2	Sí	Académicos de instituciones nacionales	
FECH_MODIF	Fecha/Hora	8	Sí	Fecha de última modificación	

Tabla – CAT_CCN

Nombre	Tipo	Tamaño	Nulo	Descripción	Valores
CLAVE	Texto	4	No	Clave de la categoría	
C_HUMANINDEX	Entero	2	Sí	Clave HUMANINDEX	
SIGLAS	Texto	20	Sí	Texto resumido de la categoría	
NOMBRE	Texto	50	No	Texto completo de la categoría	

Tabla – CAT_DEPUNAM

Nombre	Tipo	Tamaño	Nulo	Descripción	Valores
CLAVE	Texto	5	No	Clave de la entidad	
C_HUMANINDEX	Entero	2	Sí	Clave HUMANINDEX	
PRESUPUESTO	Texto	5	Sí	Código presupuestal	
NOMBRE	Texto	254	No	Nombre oficial de la dependencia	
SIGLAS	Texto	16	Sí	Nombre corto de la dependencia	
HUMANIDADES	Sí/No	1	Sí	Presenta informe de actividades en Humanidades	
SUBSISTEMA	Sí/No	1	Sí	Pertenece al Subsistema de Humanidades	
FACULTAD	Sí/No	1	Sí	Es una facultad	
Area	Sí/No	1	Sí	Área	
ORDEN	Sí/No	1	Sí	Orden	

Tabla – CAT_DISC

Nombre	Tipo	Tamaño	Nulo	Descripción	Valores
CLAVE	Texto	10	No	Clave de la disciplina	¿Clasificación de libros?
C_HUMANINDEX	Entero	2	Sí	Clave HUMANINDEX	
AREA	Texto	254	Sí	Área de conocimiento	
NOMBRE	Texto	60	No	Nombre de la disciplina	

Tabla – CAT_ESTIMULO

Nombre	Tipo	Tamaño	Nulo	Descripción	Valores
CLAVE	Texto	10	No	Clave del estímulo	
T_HUMANINDEX	Texto	35	Sí	Catálogo HUMANINDEX	
C_HUMANINDEX	Entero	2	Sí	Clave HUMANINDEX	
NOMBRE	Texto	15	No	Nombre del estímulo	
SNI	Sí/No		Sí	Estímulos del SNI	
SNC	Sí/No		Sí	Estímulos del SNC	
PRIDE	Sí/No		Sí	Estímulos del PRIDE	
PAIPA	Sí/No		Sí	Estímulos del PAIPA	
FOMDOC	Sí/No		Sí	Estímulos del FOMDOC	
PEPASIG	Sí/No		Sí	Estímulos del PEPASIG	
PEII	Sí/No		Sí	Estímulos del PEII	
JERARQUIA	Entero		Sí	Jerarquía utilizada para ordenar los registros	

Tabla – CAT_FUNCION

Nombre	Tipo	Tamaño	Nulo	Descripción	Valores
CLAVE	Texto	10	No	Clave de la función	
T_HUMANINDEX	Texto	35	Sí	Catálogo HUMANINDEX	
C_HUMANINDEX	Entero	2	Sí	Clave HUMANINDEX	
NOMBRE	Texto	30	No	Descripción de la función	
ACTIVIDAD	Sí/No	1	Sí	Actividades académicas	
EXAMEN	Sí/No	1	Sí	Exámenes profesionales	
COLEGIADOS	Sí/No	1	Sí	Órganos colegiados	
EDITORIALES	Sí/No	1	Sí	Órganos editoriales	
CURSOS	Sí/No	1	Sí	Cursos de superación académica	
ENTREVISTAS	SÍ/No	1	Si	Entrevistas	
JERARQUIA	Byte	2	Sí	Jerarquía utilizada para ordenar los registros	

Tabla – CAT_GRADO

Nombre	Tipo	Tamaño	Nulo	Descripción	Valores
CLAVE	Texto	3	No	Clave del grado	"DOC", "MTR", "ESP", "LIC", "DIP", "BAC", "S/G"
C_HUMANINDEX	Entero	2	Sí	Clave HUMANINDEX	
NOMBRE	Texto	15	No	Grado académico	
ESGRADO	Sí/No	1	Sí	Es un grado aceptado por la Coordinación de Humanidades	
UGO	Texto	3	Sí	Último grado obtenido	"DOC", "MTR", "LIC", "S/G"
JERARQUIA	Byte	1	Sí	Jerarquía utilizada para ordenar los registros	

Tabla – CAT_NAC

Nombre	Tipo	Tamaño	Nulo	Descripción	Valores
CLAVE	Texto	2	No	Clave de la nacionalidad	ISO 3166-1 alpha-2
C_HUMANINDEX	Entero	2	Sí	Clave HUMANINDEX	
NOMBRE	Texto	40	No	Nacionalidad	
PAIS	Texto	100	No	Nombre del país	

Tabla – CAT_NOMDOC

Nombre	Tipo	Tamaño	Nulo	Descripción	Valores
CLAVE	Texto	10	No	Clave del nombramiento	
C_HUMANINDEX	Entero	2	Sí	Clave HUMANINDEX	
NOMBRE	Texto	50	No	Nombramiento docente	

Tabla – CAT_POSGRADO

Nombre	Tipo	Tamaño	Nulo	Descripción	Valores
CLAVE	Texto	10	No	Clave del programa	
NOMBRE	Texto	254	No	Nombre del programa	
TRABAJADOR	Texto	10	Sí	Trabajador que registra el posgrado	

Tabla – CAT_TIPCON

Nombre	Tipo	Tamaño	Nulo	Descripción	Valores
CLAVE	Texto	10	No	Clave de contratación	
C_HUMANINDEX	Entero	2	Sí	Clave HUMANINDEX	
NOMBRE	Texto	20	No	Tipo de contratación	

Finalmente una vez recolectada y organizada la información se puede comenzar a inferir cuáles serán los Procesos de Negocio y sus posibles Dimensiones.

4.3. Diseño dimensional

El diseño dimensional es un proceso dinámico y altamente interactivo, Kimball nos define cuatro pasos para la elaboración de este:

Paso 1.- Seleccionar el proceso de negocio

El primer paso es elegir el área a modelar, éste depende fundamentalmente del análisis de requerimientos y de los alcances que tiene el proyecto. Para este paso se seleccionó primero el proceso de negocio identificado como Grupos Académicos, con la intención de entregar información totalitaria de una forma general y económica para no estar generando duplicación de datos con diferentes terminologías o enfoques. [22]

Paso 2.- Establecer el nivel de granularidad

La granularidad significa especificar el nivel de detalle. La elección de la granularidad depende de los requerimientos del negocio y lo que es posible a partir de los datos actuales. La experiencia en el desarrollo de un DWH nos dice que debemos diseñar el mayor nivel de detalle, ya que a futuro se podrían realizar agrupamientos sobre distintos niveles y no estar acotados a unos cuantos. La granularidad es el nivel de detalle que posee cada registro en una tabla de hechos. [22]

Paso 3.- Elegir las dimensiones

Las dimensiones surgen naturalmente de las discusiones del equipo y son facilitadas por la elección del nivel de granularidad, de la matriz de procesos y dimensiones. Las tablas de dimensiones tienen un conjunto de atributos o indicadores que brindan una perspectiva de análisis sobre una medida en una tabla de hechos. [22]

Paso 4.- Identificar las tablas de hechos y medidas

El último paso consiste en identificar las medidas que surgen de los procesos de negocio. Una medida es un atributo de una tabla que se desea analizar, sumarizando o agrupando sus datos, usando los criterios de corte conocidos en dimensiones [22]. Las medidas están vinculadas con el nivel de granularidad y serán quienes expongan la información que cumpla con los requerimientos del instituto.

Una vez completados estos pasos, se termina elaborando un gráfico denominado Modelo Dimensional de Alto Nivel o Gráfico de Burbujas, con el fin de documentar adecuadamente la forma en que las dimensiones son asociadas a la tabla de hechos.

➢ Grupos Académicos

El Proceso de Negocio identificado como Grupos Académicos se seleccionó debido a que la gran mayoría de los requerimientos están relacionados con los Académicos y también es la principal fuente de análisis estadístico en el IISUE.

Dimensión TIEMPO: Incluirá el atributo año.

Dimensión TIPO DE ACADEMICO: Incluirá los atributos clase, categoría, nivel y género (masculino o femenino).

Dimensión NIVEL DE ESTUDIO: Incluirá el atributo grado de estudio y su correspondiente abreviación.

Dimensión NACIONALIDAD: Incluirá el atributo nacionalidad, su correspondiente abreviación y el nombre del país.

Dimensión ESTIMULO: Incluirá los atributos estimulo, sni, pride, fomdoc y paipa.

Dimensión LUGAR: Incluirá los atributos tipo de entidad, su correspondiente abreviación, entidad, de igual manera su correspondiente abreviación, facultad, departamento y su correspondiente abreviación.

Tabla de Hechos: Incluirá los identificadores de cada una de las dimensiones y su medida será el total de académicos.

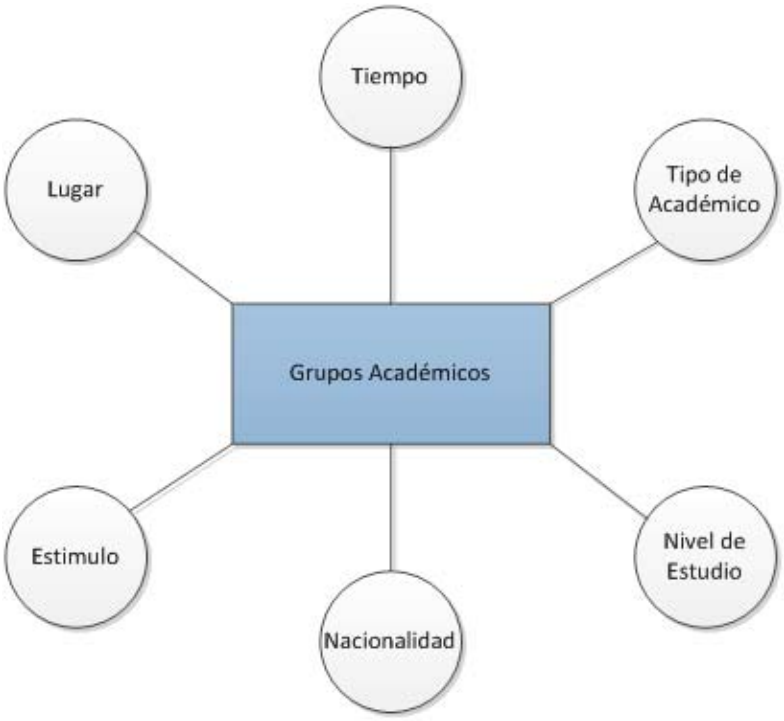


FIGURA 4.3.a. Modelo Dimensional de Alto Nivel (Grupos Académicos)

➢ Actividades Docentes

El segundo Proceso de Negocio identificado como Actividades Docentes fue seleccionado debido a que en éste se detallan específicamente todas las actividades relacionadas con el aspecto docente de cada Académico.

Dimensión TIEMPO: Incluirá el atributo año.

Dimensión ACADEMICO: Incluirá los atributos número y nombre completo.

Dimensión TIPO DE ACADEMICO: Incluirá los atributos clase, categoría, nivel y género (masculino o femenino).

Dimensión NIVEL DE ESTUDIO: Incluirá el atributo grado de estudio y su correspondiente abreviación.

Dimensión LUGAR: Incluirá los atributos tipo de entidad, su correspondiente abreviación, entidad, de igual manera su correspondiente abreviación, facultad, departamento y su correspondiente abreviación.

Tabla de Hechos: Incluirá los identificadores de cada una de las dimensiones y sus medidas serían el total de asignaturas impartidas, el total de tesis dirigidas y el total de tesinas dirigidas.

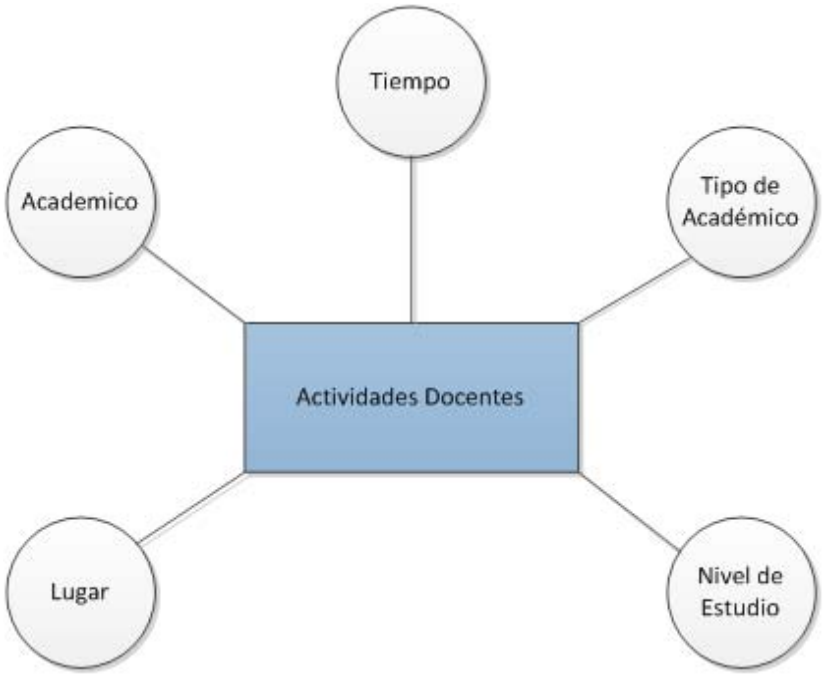


FIGURA 4.3.b. Modelo Dimensional de Alto Nivel (Actividades Docentes)

- Proyectos

Este otro Proceso de Negocio identificado como Proyectos, alberga gran parte de la información que se requiere conocer ya que es donde se encuentran la principal actividad que realizan los Académicos del IISUE.

Dimensión TIEMPO: Incluirá el atributo año.

Dimensión ACADEMICO: Incluirá los atributos número y nombre completo.

Dimensión TIPO DE ACADEMICO: Incluirá los atributos clase, categoría, nivel y género (masculino o femenino).

Dimensión TIPO DE PARTICIPACION: Incluirá los atributos de tipo de participación y su correspondiente abreviación.

Dimensión TIPO DE PROYECTO: Incluirá los atributos de tipo de proyecto y su correspondiente abreviación.

Dimensión ESTATUS DE PROYECTO: Incluirá los atributos de estatus del proyecto y su correspondiente abreviación.

Tabla de Hechos: Incluirá los identificadores de cada una de las dimensiones y su medida sería el total de proyectos.

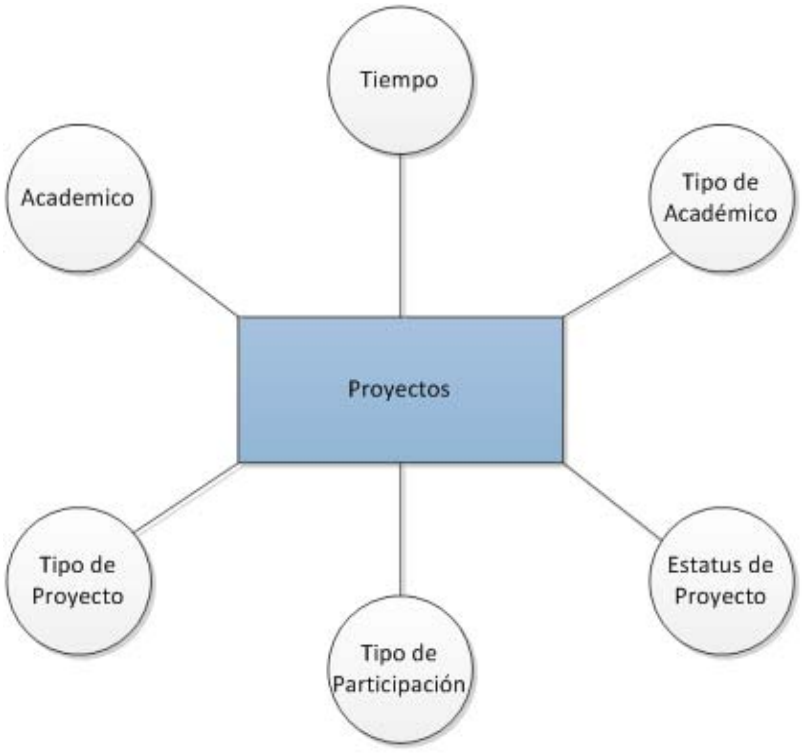


FIGURA 4.3.c. Modelo Dimensional de Alto Nivel (Proyectos)

Posteriormente los Procesos de negocio pasan a ser llamados Data Marts una vez modelados de manera dimensional. El modelado se basa en 3 tipos de esquemas (Esquema en Estrella, Esquema en Copo de Nieve y Esquema Constelación), en donde se debe seleccionar algún esquema. Nosotros en este proyecto de DWH hemos seleccionado el Esquema en Estrella puesto que es el más simple de interpretar y optimiza los tiempos de respuesta ante las consultas de los usuarios. Además este modelo es soportado por casi todas las herramientas de consulta y análisis, y los metadatos son fáciles de documentar y mantener, sin embargo es el más redundante. También cabe mencionar que al tener tres Data Marts, estos pueden modelarse en un solo Esquema Constelación que comparta las Dimensiones y existan dos o más Tablas de Hechos.

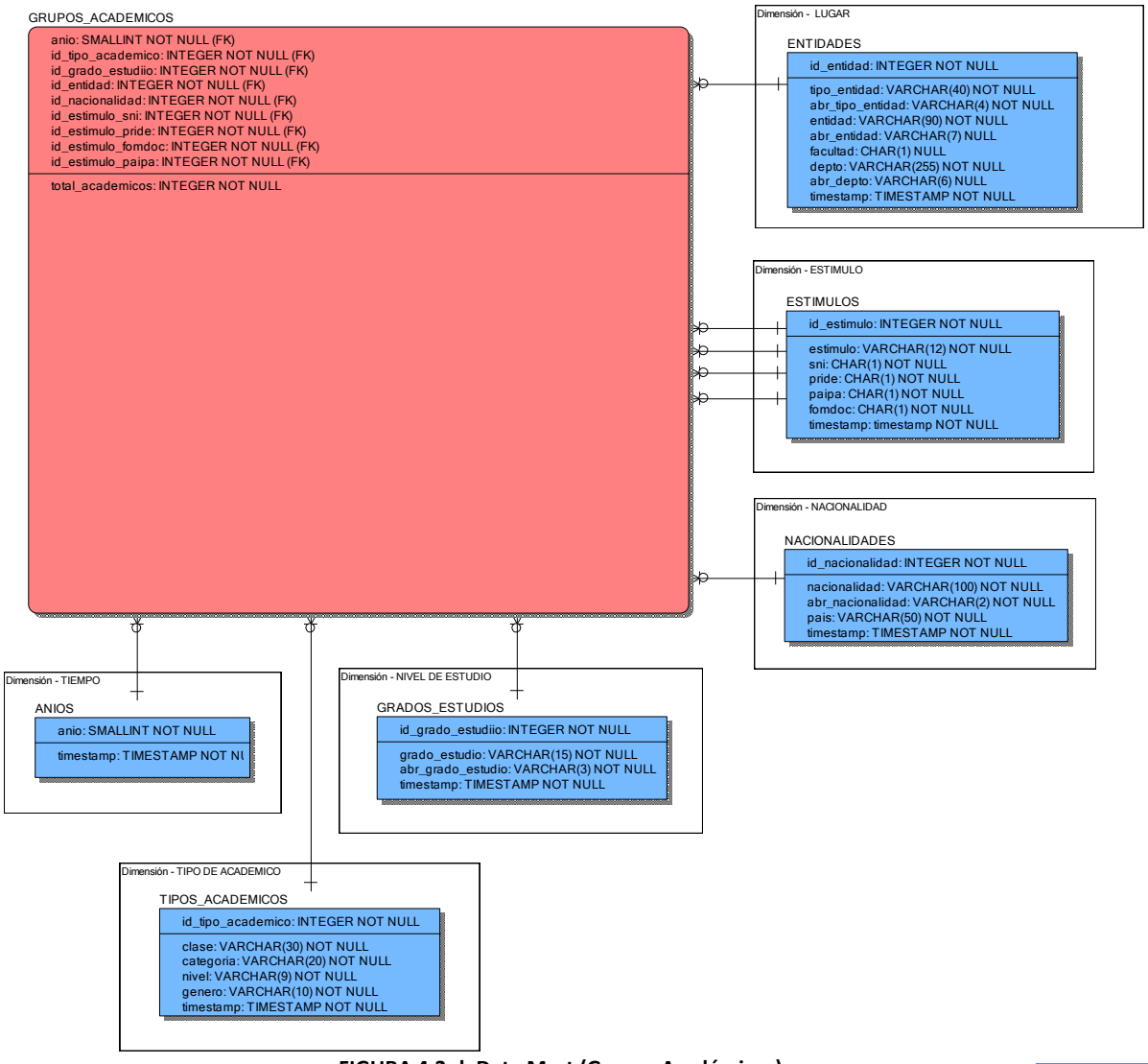


FIGURA 4.3.d. Data Mart (Grupos Académicos)

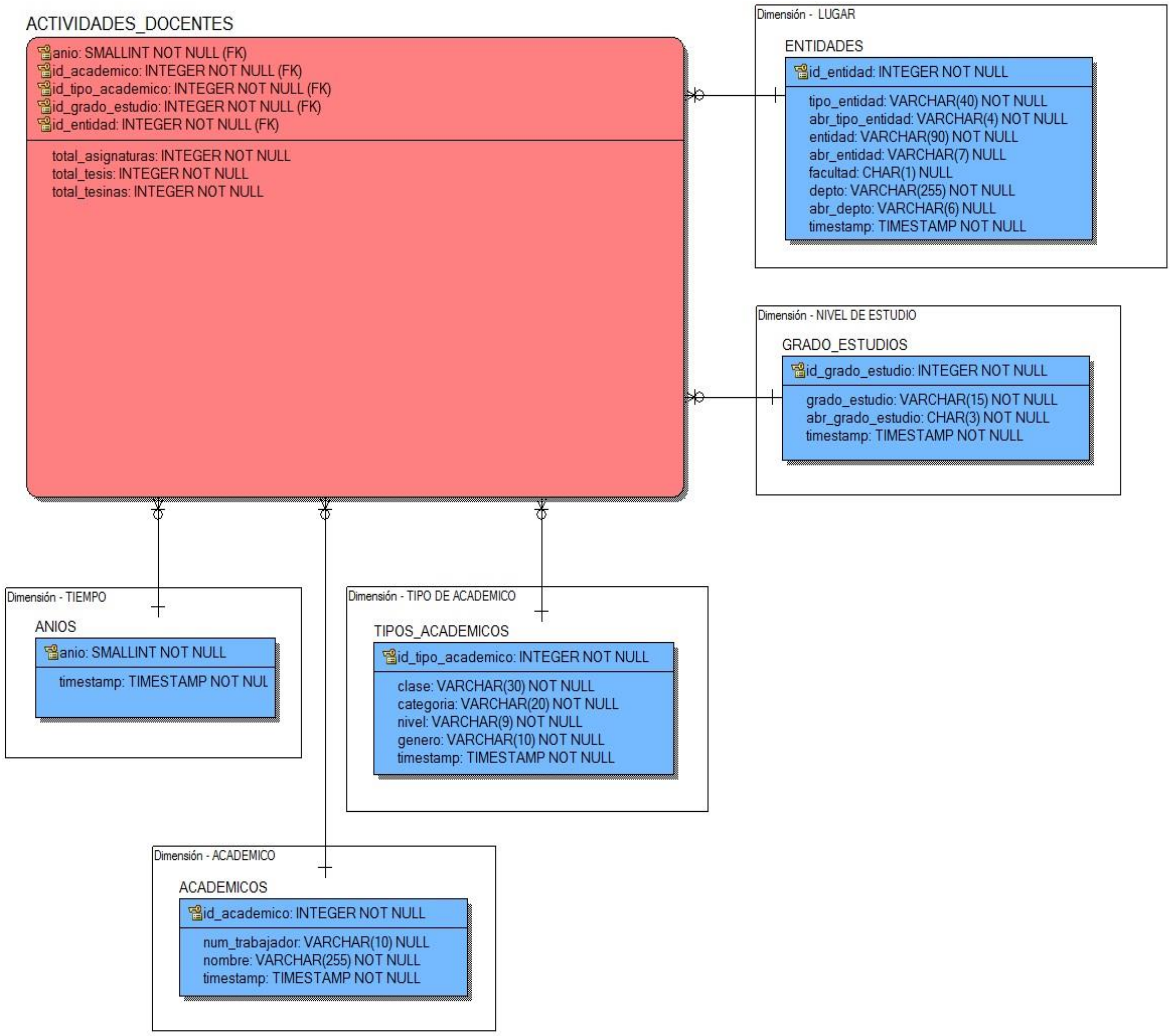


FIGURA 4.3.e. Data Mart (Actividades Docentes)

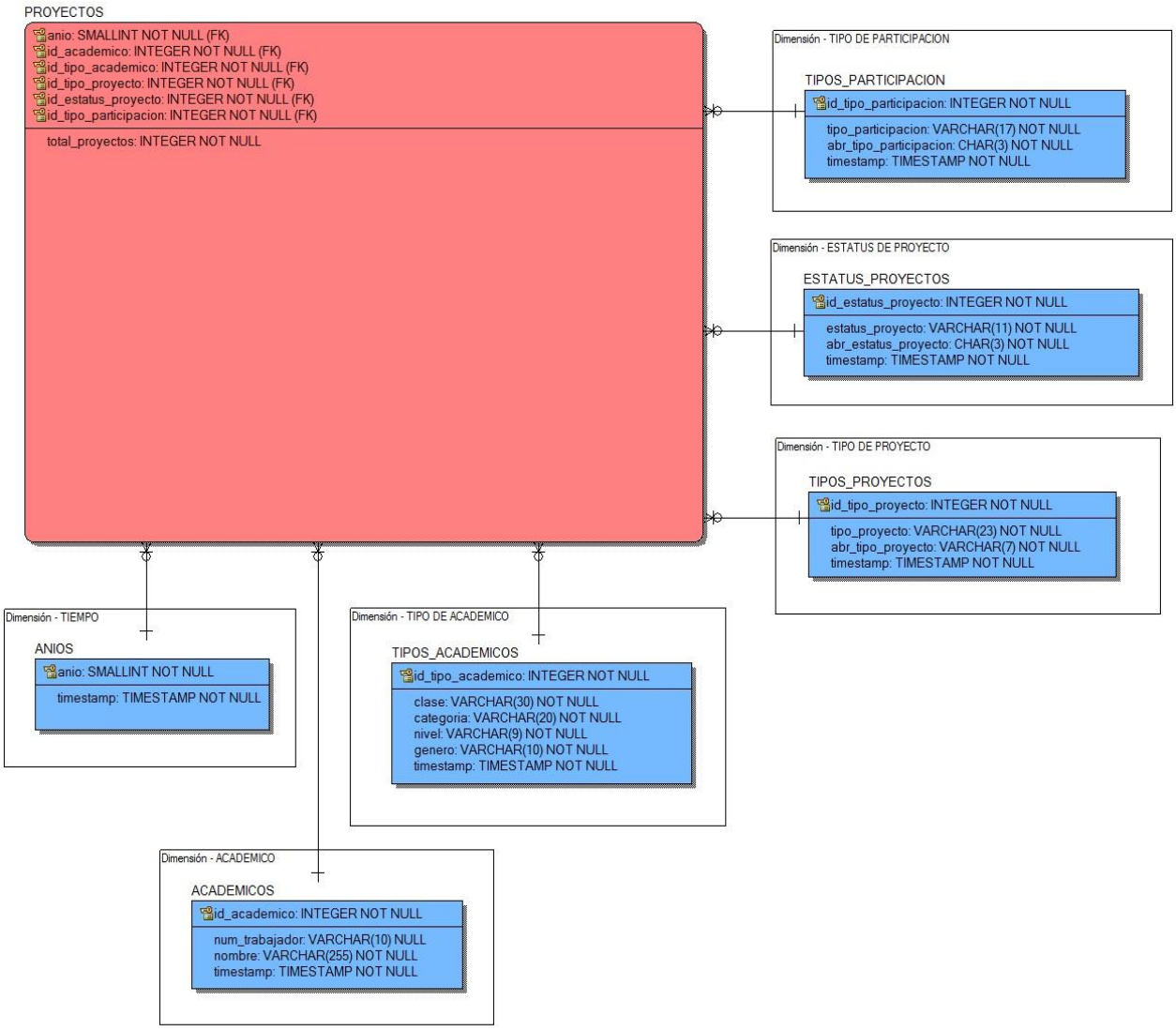


FIGURA 4.3.f. Data Mart (Proyectos)

Una vez terminados de diseñar los Data Marts se deben de identificar y validar correctamente cada uno de los atributos de las Tablas de Dimensión y de las Tablas de Hechos, con el fin de dar sustento a cada una de las Tablas que compondrá nuestro DWH.

Tabla: ANIOS Dimensión - TIEMPO					
Nombre	**Tipo de Dato**	**Descripción**	**PK**	**Origen de Datos**	**Transformación**
anio	SMALLINT	Año.	Si		Obtener el año a partir del nombre del archivo de MS Access.
Timestamp	TIMESTAMP	Fecha de último cambio.	No		

Tabla: TIPOS_ACADEMICOS Dimensión - TIPO DE ACADÉMICO					
Nombre	**Tipo de Dato**	**Descripción**	**PK**	**Origen de Datos**	**Transformación**
id_tipo_academico	INTEGER	Identificador único del tipo de académico.	Si		Auto incremento del máximo de la tabla.
clase	VARCHAR(30)	Clase del académico.	No	ACADEMICOS CAT_CCN	Se mapeara el campo “CAT_CCN.nombre” para obtener la clase
categoria	VARCHAR(20)	Categoría del académico.	No	ACADEMICOS CAT_CCN	Se mapeara el campo “CAT_CCN.nombre” para obtener la clase.
nivel	VARCHAR(9)	Nivel del académico.	No	ACADEMICOS CAT_CCN	Se mapeara el campo “CAT_CCN.nombre” para obtener el nivel.
genero	VARCHAR(10)	Genero del académico.	No	ACADEMICOS	Se mapeara el campo “ACADEMICOS.sexo”.
Timestamp	TIMESTAMP	Fecha de último cambio.	No		

Tabla: GRADOS_ESTUDIOS Dimensión - NIVEL DE ESTUDIO					
Nombre	**Tipo de Dato**	**Descripción**	**PK**	**Origen de Datos**	**Transformación**
id_grado_estudiio	INTEGER	Identificador único del grado de estudio	Si		Auto incremento del máximo de la tabla.
grado_estudio	VARCHAR(15)	Nombre del grado de estudio.	No	CAT_GRADO	Se extrae del campo “CAT_GRADO.nombre”
abr_grado_estudio	VARCHAR(3)	Abreviatura del grado de estudio.	No	CAT_GRADO	Se extrae del campo “CAT_GRADO.clave”
Timestamp	TIMESTAMP	Fecha de último cambio.	No		

Tabla: NACIONALIDADES Dimensión - NACIONALIDAD					
Nombre	**Tipo de Dato**	**Descripción**	**PK**	**Origen de Datos**	**Transformación**
id_nacionalidad	INTEGER	Identificador único de la nacionalidad.	Si		Auto incremento del máximo de la tabla.
nacionalidad	VARCHAR(100)	Nacionalidad.	No	CAT_NAC ACADEMICOS ASIGNA TESIS	Se extrae del campo “CAT_NAC.nombre”
abr_nacionalidad	CHAR(2)	Abreviatura de la nacionalidad.	No	CAT_NAC ACADEMICOS ASIGNA TESIS	Se extrae del campo “CAT_NAC.clave”
país	VARCHAR(50)	País.	No	CAT_NAC ACADEMICOS ASIGNA TESIS	Se extrae del campo “CAT_NAC.pais”
Timestamp	TIMESTAMP	Fecha de último cambio.	No		

Tabla: ESTIMULOS Dimensión – ESTIMULO (SNI, PRIDE, PAIPA, FOMDOC)					
Nombre	**Tipo de Dato**	**Descripción**	**PK**	**Origen de Datos**	**Transformación**
id_estimulo	INTEGER	Identificador único del estímulo.	Si		Auto incremento del máximo de la tabla.
estimulo	VARCHAR(12)	Descripción de estímulo.	No	CAT_ESTIMULO ACADEMICOS	Se extrae del campo “CAT_ESTIMULO.nombre”.
sni	CHAR(1)	Estímulos que da el SNI (Sistema Nacional de Investigadores).	No	CAT_ESTIMULO ACADEMICOS	Se extrae del campo “CAT_ESTIMULO.sni”.
pride	CHAR(1)	Estímulos que da el PRIDE (Programa de Primas al Desempeño del Personal Académico de Tiempo Completo).	No	CAT_ESTIMULO ACADEMICOS	Se extrae del campo “CAT_ESTIMULO.pride”.
paipa	CHAR(1)	Estímulos que da el PAIPA (Programa de Apoyo a la Incorporación de Personal Académico de Tiempo Completo).	No	CAT_ESTIMULO ACADEMICOS	Se extrae del campo “CAT_ESTIMULO.paipa”.
fomdoc	CHAR(1)	Estímulos que da el FOMDOC (Programa de Estímulos de Fomento a la Docencia).	No	CAT_ESTIMULO ACADEMICOS	Se extrae del campo “CAT_ESTIMULO.fomdoc”.
Timestamp	TIMESTAMP	Fecha de último cambio.	No		

Tabla: ENTIDADES Dimensión – LUGAR					
Nombre	**Tipo de Dato**	**Descripción**	**PK**	**Origen de Datos**	**Transformación**
id_entidad	INTEGER	Identificador único de la entidad.	Si		Auto incremento del máximo de la tabla.
tipo_entidad	VARCHAR(40)	Tipo de la entidad.	No	CAT_DEPUNAM ASIGNA TESIS	Se mapean los valores de “ASIGNA.tipo_inst” y “TESIS.tipo_inst”.
abr_tipo_entidad	VARCHAR(4)	Abreviatura del tipo de la entidad.	No	CAT_DEPUNAM ASIGNA TESIS	Se mapean los valores de “ASIGNA.tipo_inst” y “TESIS.tipo_inst”.
entidad	VARCHAR(90)	Nombre de la entidad.	No	CAT_DEPUNAM ASIGNA TESIS	Se obtiene el valor de “CAT_DEPUNAM.nombre”.
abr_entidad	VARCHAR(7)	Abreviatura de la entidad.	No	CAT_DEPUNAM ASIGNA TESIS	Se obtiene el valor de “CAT_DEPUNAM.siglas”.
facultad	CHAR(1)	Si la entidad es una facultad.	No	CAT_DEPUNAM ASIGNA TESIS	Se obtiene el valor de “CAT_DEPUNAM.facultad”.
depto	VARCHAR(100)	Departamento dentro de la entidad.	No	CAT_DEPUNAM ACADEMICOS	Se extrae del campo “ACADEMICOS.departamento”.
abr_depto	VARCHAR(50)	Abreviatura del Departamento.	No	CAT_DEPUNAM ACADEMICOS	Se extrae del campo “ACADEMICOS.departamento”.
Timestamp	TIMESTAMP	Fecha de último cambio.	No		

Tabla: ACADEMICOS Dimensión – ACADEMICO					
Nombre	**Tipo de Dato**	**Descripción**	**PK**	**Origen de Datos**	**Transformación**
id_academico	INTEGER	Identificador único del académico.	Si		Auto incremento del máximo de la tabla.
num_trabajador	VARCHAR(10)	Número de trabajador	No	ACADEMICOS	Se extrae del campo “ACADEMICOS.trabajador”
nombre	VARCHAR(255)	Nombre completo del académico.	No	ACADEMICOS	Se extrae de los campos “ACADEMICOS.nombres” “ACADEMICOS.ape_pat” “ACADEMICOS.ape_mat” y se concatenan.
Timestamp	TIMESTAMP	Fecha de último cambio.	No		

Tabla: TIPOS_PARTICIPACION Dimensión – TIPO DE PARTICIPACION					
Nombre	**Tipo de Dato**	**Descripción**	**PK**	**Origen de Datos**	**Transformación**
id_tipo_participacion	INTEGER	Identificador único del tipo de participación.	Si		Auto incremento del máximo de la tabla.
tipo_participacion	VARCHAR(17)	Tipo de participación en el proyecto.	No	PROYECTOS	Se mapeara el valor del campo "PROYECTOS.participa".
abr_tipo_participacion	CHAR(3)	Abreviatura del tipo de participación en el proyecto.	No	PROYECTOS	Se mapeara el valor del campo PROYECTOS.participa.
Timestamp	TIMESTAMP	Fecha de último cambio.	No		

Tabla: TIPOS_PROYECTOS Dimensión – TIPO DE PROYECTO					
Nombre	**Tipo de Dato**	**Descripción**	**PK**	**Origen de Datos**	**Transformación**
id_tipo_proyecto	INTEGER	Identificador único del tipo de proyecto.	Si		Auto incremento del máximo de la tabla.
tipo_proyecto	VARCHAR(23)	Tipo de proyecto.	No	PROYECTOS	Se mapeara el valor del campo "PROYECTOS.tipo".
abr_tipo_proyecto	VARCHAR(7)	Abreviatura del tipo de proyecto.	No	PROYECTOS	El valor del campo "PROYECTOS.tipo".
Timestamp	TIMESTAMP	Fecha de último cambio.	No		

Tabla: ESTATUS_PROYECTO Dimensión – ESTATUS DE PROYECTO					
Nombre	**Tipo de Dato**	**Descripción**	**PK**	**Origen de Datos**	**Transformación**
id_estatus_proyecto	INTEGER	Identificador único del estatus del proyecto.	Si		Auto incremento del máximo de la tabla.
estatus_proyecto	VARCHAR(11)	Estatus del proyecto.	No	PROYECTOS	Se mapeara el valor del campo “PROYECTOS.situacion”.
abr_estatus_proyecto	CHAR(3)	Abreviatura del estatus del proyecto.	No	PROYECTOS	Se mapeara el valor del campo “PROYECTOS.situacion”.
Timestamp	TIMESTAMP	Fecha de último cambio.	No		

Tabla: GRUPOS_ACADEMICOS Hecho					
Nombre	**Tipo de Dato**	**Descripción**	**PK**	**Origen de Datos**	**Transformación**
anio	SMALLINT	Año.	Si		No
id_tipo_academico	INTEGER	Identificador único del tipo de Académico.	Si		No
id_grado_estudiio	INTEGER	Identificador único del grado de estudio del académico.	Si		No
id_entidad	INTEGER	Identificador único de la entidad.	Si		No
id_nacionalidad	INTEGER	Identificador único de la nacionalidad.	Si		No
id_estimulo_sni	INTEGER	Identificador único del estímulo SNI.	Si		No
id_estimulo_pride	INTEGER	Identificador único del estímulo PRIDE.	Si		No
id_estimulo_fomdoc	INTEGER	Identificador único del estímulo FOMDOC.	Si		No
id_estimulo_paipa	INTEGER	Identificador único del estímulo PAIPA.	Si		No
total_academicos	INTEGER	Total de académicos.	No	ACADEMICOS	Conteo de académicos.

Tabla: ACTIVIDADES_DOCENTES Hecho					
Nombre	**Tipo de Dato**	**Descripción**	**PK**	**Origen de Datos**	**Transformación**
anio	SMALLINT	Año.	Si		No
id_academico	INTEGER	Identificador único del académico.	Si		No
id_tipo_academico	INTEGER	Identificador único del tipo de académico.	Si		No
id_grado_estudio	INTEGER	Identificador único del nivel académico.	Si		No
id_entidad	INTEGER	Identificador único de la entidad.	Si		No
total_asignaturas	INTEGER	Total de asignaturas.	No	ASIGNA ACADEMICOS	Conteo de asignaturas.
total_tesis	INTEGER	Total de tesis.	No	TESIS ACADEMICOS	Conteo de tesis.
total_tesinas	INTEGER	Total de tesinas.	No	TESIS ACADEMICOS	Conteo de tesinas.

Tabla: PROYECTOS Hecho					
Nombre	**Tipo de Dato**	**Descripción**	**PK**	**Origen de Datos**	**Transformación**
anio	SMALLINT	Año.	Si		No
id_academico	INTEGER	Identificador único del académico.	Si		No
id_tipo_academico	INTEGER	Identificador único del tipo de académico.	Si		No
id_tipo_proyecto	INTEGER	Identificador único del tipo de proyecto.	Si		No
id_estatus_proyecto	INTEGER	Identificador único del estatus del proyecto.	Si		No
id_tipo_participacion	INTEGER	Identificador único del tipo de participación.	Si		No
total_proyectos	INTEGER	Total de proyectos.	No	PROYECTOS ACADEMICOS	Conteo de proyectos.

CAPÍTULO 5. Desarrollo de un DWH

Data Warehouse es una solución utilizada por muchas empresas para la centralización de sus datos de negocio permitiendo la generación de informes y exploración de datos estratégicos, de tal manera que nos permita tomar decisiones correctas.

Por lo tanto el Data Warehouse, aparece como un proceso de integración, consolidación y administración de distintas fuentes, cuyo propósito es satisfacer necesidades críticas que puedan cambiar el rumbo del negocio.

En este capítulo se explica paso a paso el proceso de ETL y finalmente se consolida el desarrollo del Data Warehouse, con la intención de dar solución a la lista de requerimientos por parte del IISUE y pueda servir de base para perseguir nuevos objetivos que den soporte a la toma de decisiones estratégicas en otras áreas o institutos.

5.1. Plataforma de desarrollo

Antes de comenzar la construcción del DWH para el IISUE, debemos de especificar la plataforma que alojará nuestro sistema piloto de DWH, debido a los costos que implica la compra y mantenimiento de recursos que son generados para mantener un ambiente de pruebas y producción, hemos propuesto la idea de que este proyecto se desarrolle sobre una plataforma virtual, en pocas palabras, el desarrollo del presente proyecto será sobre una máquina virtual[13], con un Sistema Operativo, CentOS (Community ENTerprise Operating System) 5.8.

5.1.1. Sistema Operativo

Utilizamos Centos 5.8 debido a que es un S.O. empresarial basado en Red Hat Enterprise Linux, compuesto de software libre que está disponible para arquitecturas de 32 y 64 bits, puede ser descargado y usado libremente. Además cada versión es mantenida durante 7 años por medio de actualizaciones de seguridad.

5.1.2. Bases de Datos

Para la construcción de la Base de Datos utilizamos PostgreSQL 9.1 ya que es uno de los motores más potentes de código abierto y es desarrollado por una comunidad de desarrolladores que trabajan de forma desinteresada, altruista, libre y/o apoyada por organizaciones comerciales. Además de que contiene excelente documentación, estabilidad, extensibilidad, número ilimitado de registros, ilimitado tamaño de la base de datos (depende del sistema de almacenamiento) y muchas otras característica que la hacen una opción atractiva.

[13] Una máquina virtual es un software que emula y ejecuta software como si fuera una computadora real.

5.1.3. Herramientas para el DWH

Para la elaboración del proceso de ETL que nos permita alimentar nuestro DWH previamente diseñado, utilizaremos el módulo de Pentaho Data Integration 4.4. Y finalmente como nuestra herramienta de acceso, exploración y análisis de datos será Pentaho BI Server Community Edition 4.8. También tenemos que agregar el uso de otra herramienta que modela la construcción de los esquemas ROLAP y además permite publicarlos en el Pentaho BI Server para que este mismo pueda analizar grandes cantidades de datos almacenados en bases de datos SQL (Lenguaje de Consulta Estructurado) de una forma interactiva sin necesidad de escribir consultas, tal herramienta se llama, Pentaho Schema Workbench Community Edition 3.

5.2. Creación de la base de datos del DWH

Una vez diseñados los modelos multidimensionales de los Data Marts podemos comenzar a construir físicamente la base de datos del DWH, en este caso como PostgreSQL 9.1 fue elegido para almacenar los datos que contendrá el DWH.

A nivel físico de almacenamiento PostgreSQL 9.1 es un SGDB que delimita el tamaño de las bases de datos en base al sistema de almacenamiento que se esté utilizando, mientras que las tablas, campos, índices y demás objetos están sujetos a tamaños bastante considerables para dar soporte al uso de Tecnologías como el DWH, entonces tanto el tamaño de almacenamiento y el rendimiento para ejecutar consultas no suele ser un factor determinante que nos límite a la hora de definir la estructura de las Tablas e Índices que compondrán nuestro DWH, además de que también estará alimentado anualmente con bases de datos de un tamaño total de +/-5 MB. Y los índices que nos permitirán dar rendimiento a las consultas son de tipo B-tree, ya que es el que se da por default en las llaves primarias de las tablas. [Ver ANEXO VIII].

5.3. Proceso de ETL

El proceso de ETL (Extracción, Transformación y Carga) está definido por un solo flujo que contempla la alimentación de los 3 Data Marts, además de que contiene reglas de extracción de datos para mantener la integridad y la calidad de los datos en el DWH. También el flujo se diseñó en gran medida para identificar y alertar a los usuarios por si existen errores en la definición de la estructura de las tablas y/o metadatos de origen o para notificarles que el proceso ha sido ejecutado con éxito.

5.3.1. ETL Dimensión TIEMPO

La Dimensión TIEMPO contiene los años de cada informe académico. Este se obtendrá del nombre del archivo de la base de datos de Access, se aplicarán algunas transformaciones para el manejo de los datos, por ultimo pasaremos la información a la tabla “anios”.

El esquema de transformación se muestra en la FIGURA 5.3.1.a.

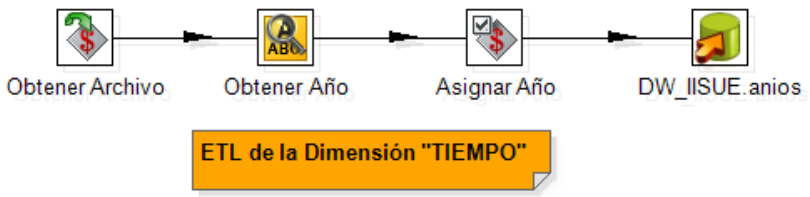


FIGURA 5.3.1.a. TEL Dimensión Tiempo

Veremos en detalle cada uno de los pasos:

Obtener Archivo: con el paso "Obtener Variable" pasamos la variable "${ARCHIVO}" a un campo con el nombre "archivo" de tipo string, para que esté disponible en el resto de la transformación y así poder manipularlo.

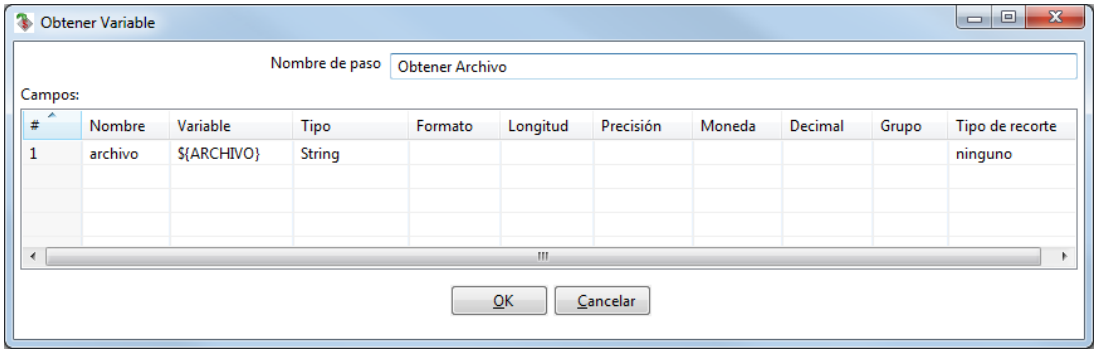


FIGURA 5.3.1.b. Obtener archivo

Obtener año: en este paso se crearán 4 expresiones regulares. La primera obtendrá el nombre del sistema. El segundo obtendrá el año del archivo de la base de datos. El tercero la dependencia, que en este caso será el nombre de la institución (IISUE). El cuarto y último la extensión del archivo. Para que esto funcione el nombre del archivo tiene que cumplir con el formato "sistema_año_dependencia.mdb|.accdb", un ejemplo podría ser "SIAH_2011_IISUE.mdb".

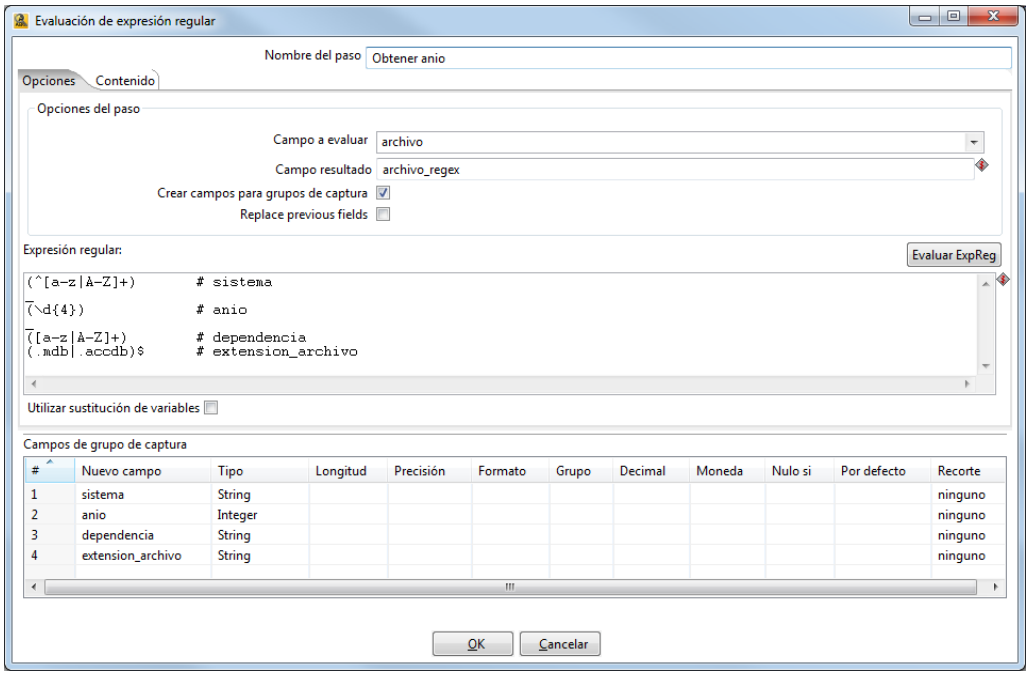


FIGURA 5.3.1.c. Obtener año

Asignar Año: en el paso "Asignar valor a variables" se le asignara el valor del campo "anio" a la variable "ANIO" para ser usado en otras transformaciones que se verán más adelante.

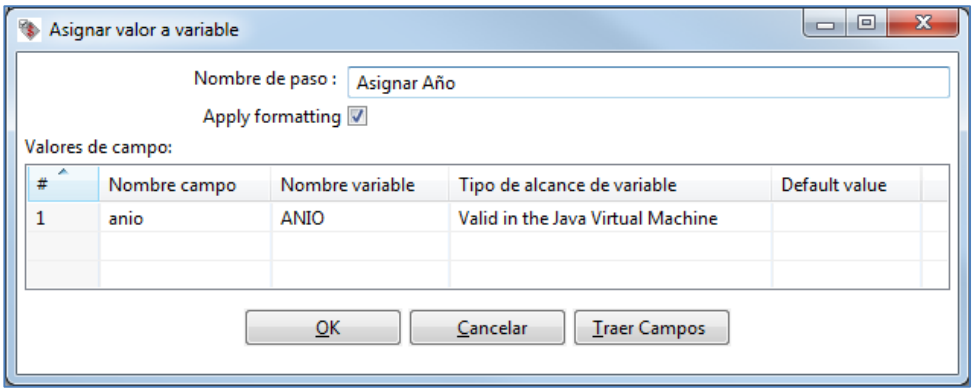


FIGURA 5.3.1.d. Asignar Año

Tabla - DW_IISUE.anios: como paso final en la transformación, utilizaremos el paso "Tabla de salida" para insertar los registros generados en la tabla "anios". Donde se asignará la conexión a la base de datos, su esquema, la tabla destino de los registros y los "database fields", en "Table field" tendrán los nombres de los campos de la tabla destino "anios" y en "Stream field" estarán los nombres de los campos que vienen del paso anterior.

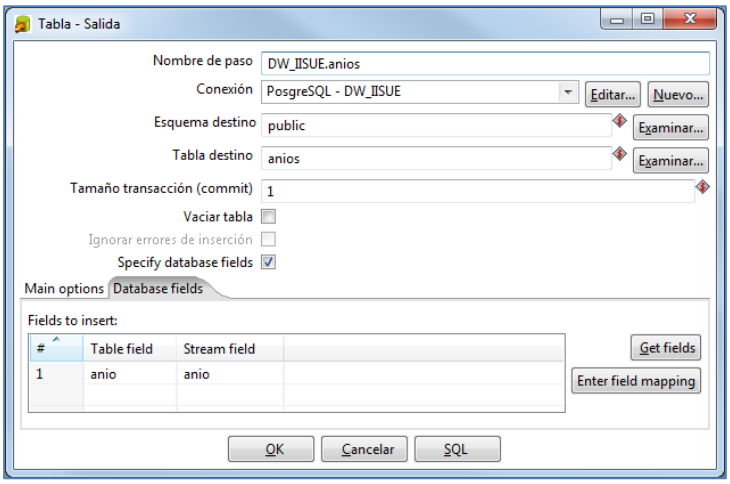


FIGURA 5.3.1.e. Tabla –DW_IISUE

5.3.2. ETL Dimensión TIPO DE ACADEMICO

En esta dimensión se obtendrán las clases, categorías, niveles y el género de los académicos.

Su esquema de trasformación se puede observar en la FIGURA 5.3.2.a.

FIGURA 5.3.2.a. ETL Dimensión TIPO DE ACADEMICO

SIAH.academicos: con el paso "Entrada Tabla" se obtendrá los registros del nombre completo de la categoría de los académicos (PROF ASO A MT, PROF ASO B MT, etc) y el género de cada uno de ellos de la tabla "academicos".

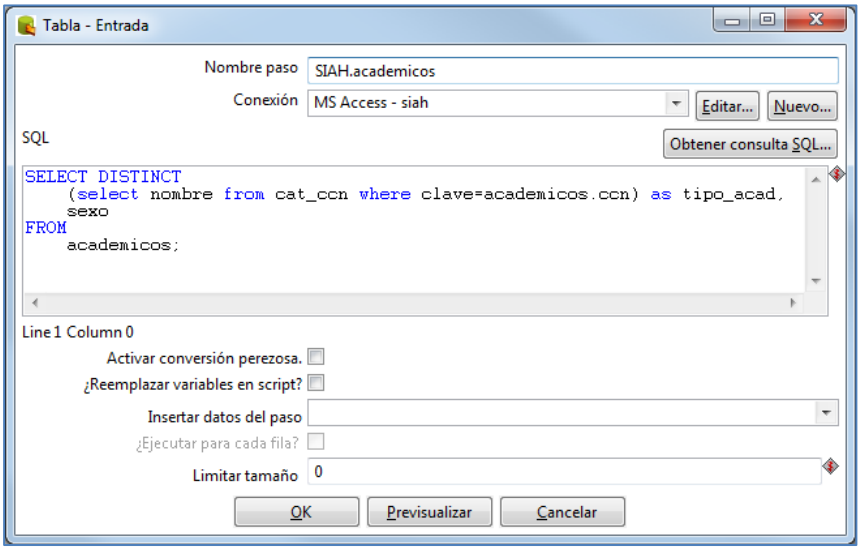


FIGURA 5.3.2.b. SIAH.academicos

Obtener clase, categoría y nivel: en el paso “Valores de Script” obtendremos tres valores de campos (clase, categoría y nivel), que se extraerán de la columna “tipo_acad” para obtener dicha separación.

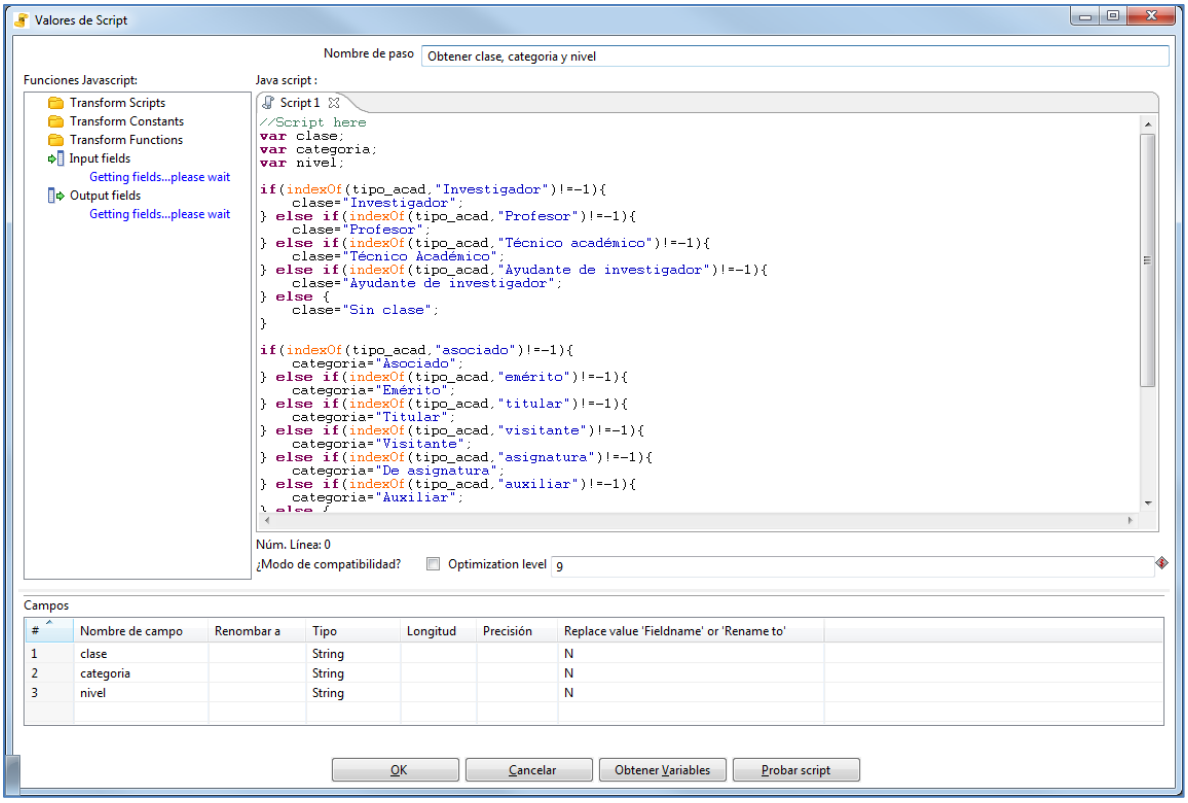


FIGURA 5.3.2.c. Obtener clase, categoría y nivel

En la FIGURA 5.3.2.d. se detalla cómo se obtienen los siguientes valores:

Clase: si la función indexOf encuentra en la cadena “Investigador” en la variable “tipo_acad” el resultado será un número diferente de “-1”, daría la condición verdadero y el valor de la variable clase sería “Investigador”. Si es igual a “-1” hará la misma comparación pero ahora buscando las cadenas "Profesor", "Técnico Académico" y "Ayudante de investigador" para obtener la clase de cada una de estas. Sino encuentra ninguno de estos valores la “clase” sería igual a “Sin clase”.

Categoría: lo mismo que clase, que buscará "Investigador", "Profesor", "Técnico académico" y "Ayudante de investigador" para asignarle a la variable "categoría" su valor correspondiente y sino encuentra nada será igual a "sin categoría".

Nivel: lo mismo que los dos puntos anteriores.

```
//Script here
var clase; var categoria; var nivel;
if(indexOf(tipo_acad,"Investigador")!=-1){
		clase="Investigador";
} else if(indexOf(tipo_acad,"Profesor")!=-1){
		clase="Profesor";
} else if(indexOf(tipo_acad,"Técnico académico")!=-1){
		clase="Técnico Académico";
} else if(indexOf(tipo_acad,"Ayudante de investigador")!=-1){
		clase="Ayudante de investigador";
} else {
		clase="Sin clase";
}
if(indexOf(tipo_acad,"asociado")!=-1){
		categoria="Asociado";
} else if(indexOf(tipo_acad,"emérito")!=-1){
		categoria="Emérito";
} else if(indexOf(tipo_acad,"titular")!=-1){
		categoria="Titular";
} else if(indexOf(tipo_acad,"visitante")!=-1){
		categoria="Visitante";
} else if(indexOf(tipo_acad,"asignatura")!=-1){
		categoria="De asignatura";
} else if(indexOf(tipo_acad,"auxiliar")!=-1){
		categoria="Auxiliar";
} else {
		categoria="Sin categoría";
}
if(indexOf(tipo_acad,"A")!=-1){
		nivel="A";
} else if(indexOf(tipo_acad,"B")!=-1){
		nivel="B";
} else if(indexOf(tipo_acad,"C")!=-1){
		nivel="C";
} else {
		nivel="Sin nivel";
}
```

FIGURA 5.3.2.d. Obtener clase, categoría y nivel

Mapeo de género: en el siguiente paso “Mapeo de valores”, todos los registros del campo “sexo” que tengan el valor de “M” se cambiará a “Hombre”, los que tengan “F” serán “Mujer” y los campos que no tengan nada se les asignará “sin género”.

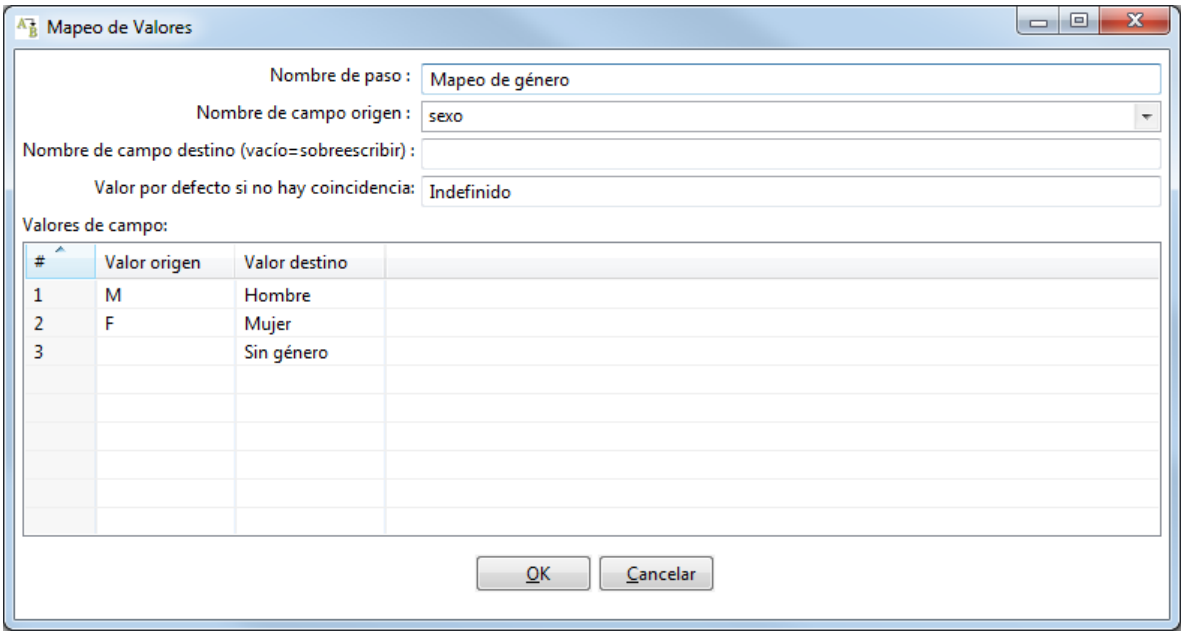


FIGURA 5.3.2.e. Mapeo de género

DW_IISUE.tipos_academicos: con el paso de tipo “Búsqueda /Actualización de Combinación” generamos la actualización de la tabla. El paso recibe el flujo de datos con toda la estructura de los campos y se realiza la verificación de estos contra los de la base de datos.

El “timestamp” es el campo para saber cuál es la fecha de la última actualización de los registros.

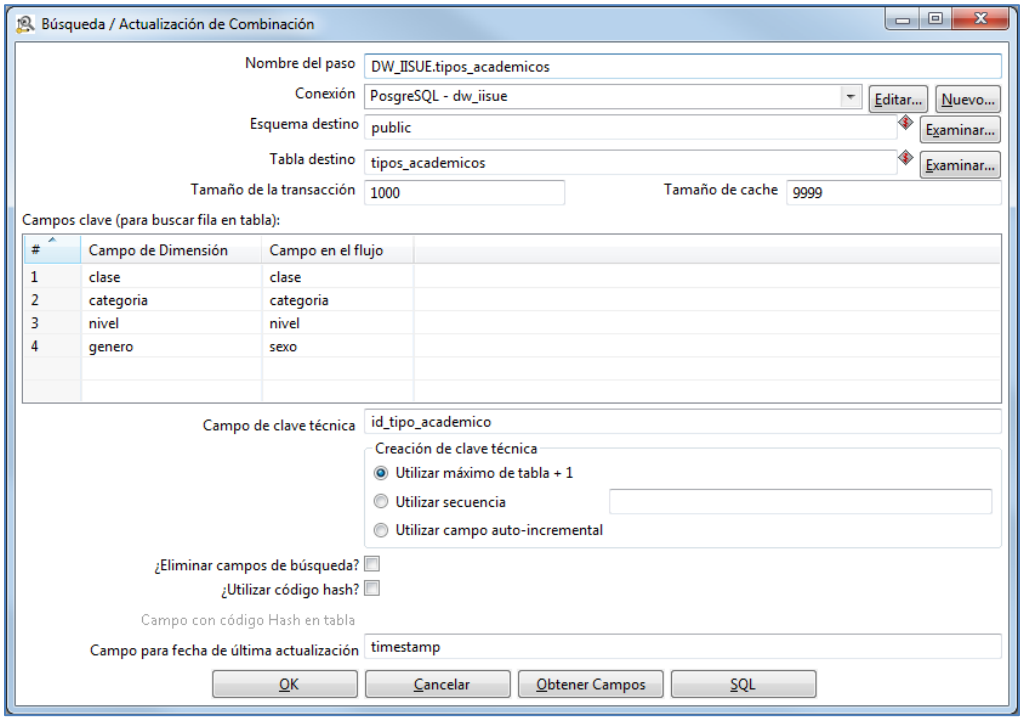


FIGURA 5.3.2.f. DW_IISUE.tipos_academicos

5.3.3. ETL Dimensión NIVEL DE ESTUDIO

En la dimensión NIVEL DE ESTUDIO se obtendrá el grado aceptado por la Coordinación de Humanidades de los académicos.

La FIGURA 5.3.3.a. muestra los pasos del ETL de la dimensión NIVEL DE ESTUDIO.

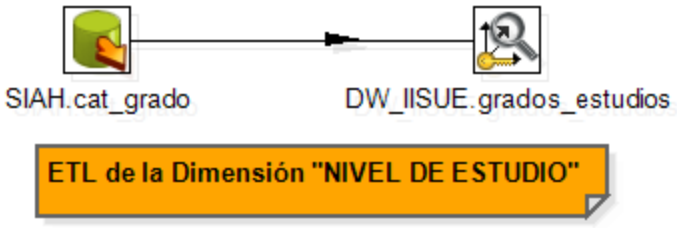


FIGURA 5.3.3.a. ETL de la Dimensión NIVEL DE ESTUDIO

SIAH.cat_grado: este tipo de paso ya se ha detallado anteriormente, así que solo se explicará la consulta siguiente

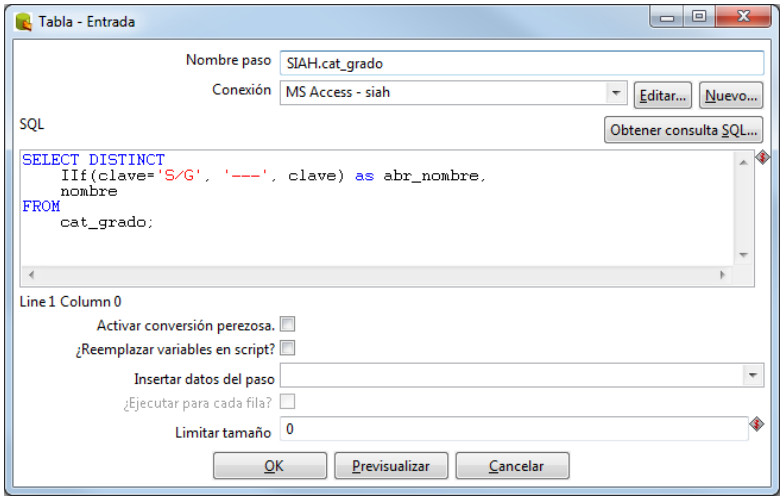


FIGURA 5.3.3.b. SIAH.cat_grado

En la FIGURA 5.3.3.b. se obtendrán las diferentes abreviatura del nombre, con el IIf los valores del campo “clave” que tengan “S/G” se remplazará por “---“. También elegiremos el campo “nombre” que contendrá el grado académico.

DW_IISUE.grados_estudios: “Búsqueda /Actualización de Combinación” como se explicó antes, se generamos la actualización de la tabla destino, recibiendo el flujo de datos con toda la estructura de los campos, y se realiza la verificación de estos contra los de la base de datos. En este caso sólo son dos comparaciones “grado_estudio” contra “nombre” y el “abr_grado_estudio” contra “abr_nombre”.

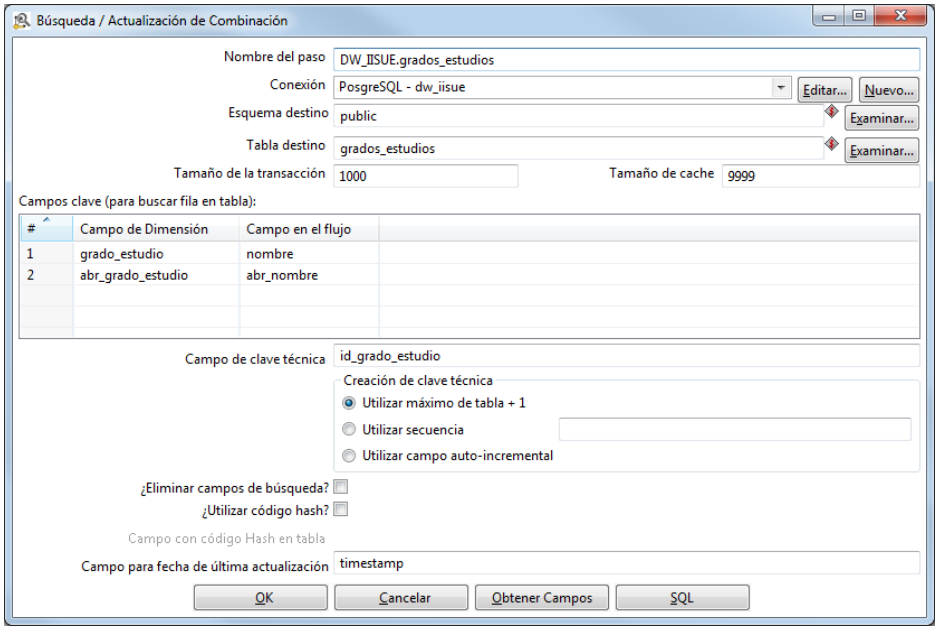


FIGURA 5.3.3.c. SIAH.grados_estudios

5.3.4. ETL Dimensión LUGAR

Dimensión LUGAR contendrá almacenada la nacionalidad y su abreviatura de los diferentes académicos, de las asignaturas impartidas y de las tesis.

En la FIGURA 5.3.4.a. se pueden observar los pasos de la transformación ETL Dimensión LUGAR.

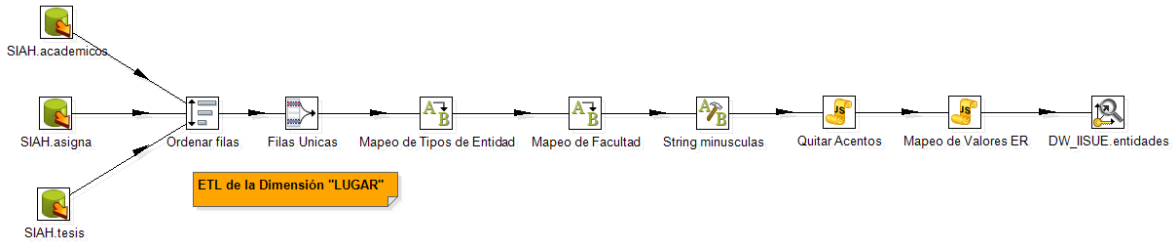


FIGURA 5.3.4.a. ETL de la Dimensión LUGAR

SIAH.academicos, SIAH.asigna y SIAH.tesis: el tipo de paso “Entrada Tabla” ya se ha explicado anteriormente, sin embargo esta vez tenemos tres entradas, así que sólo explicaremos la primera consulta para evitar que sea muy repetitivo.

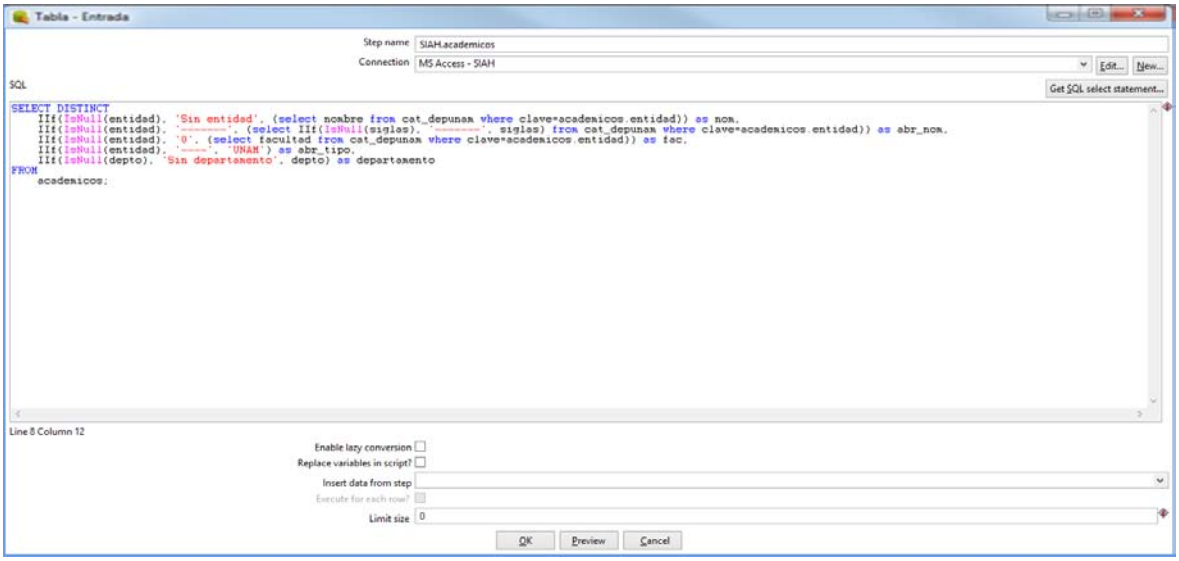


FIGURA 5.3.4.b. SIAH.academicos

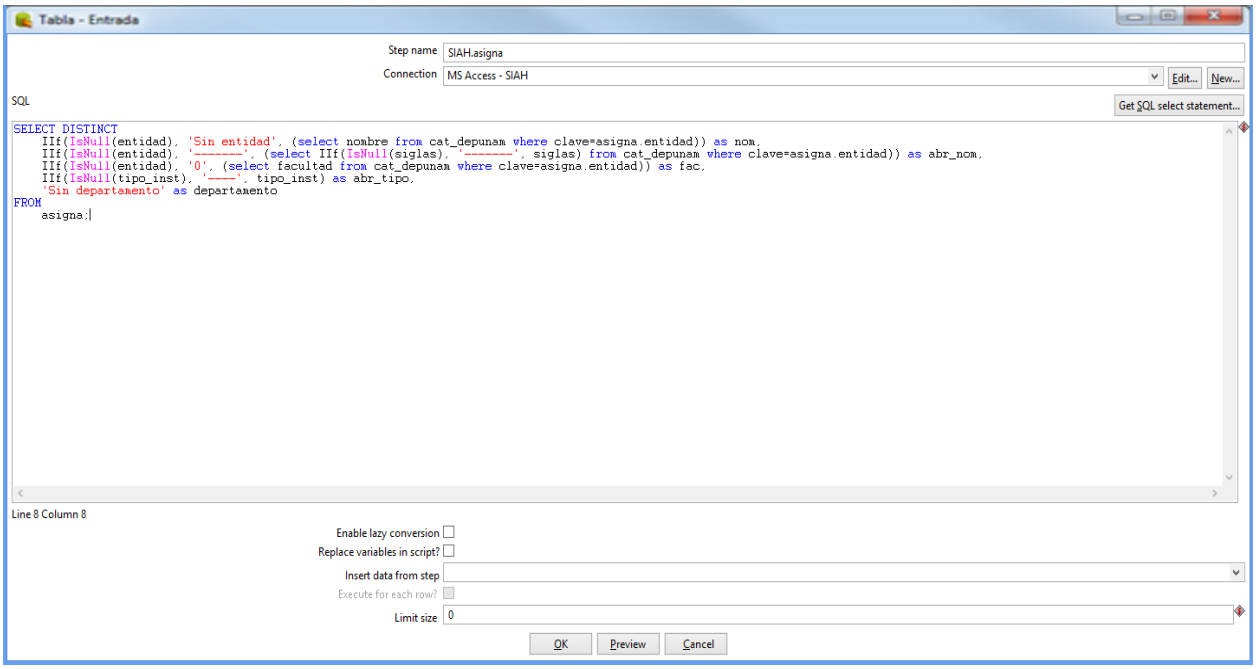


FIGURA 5.3.4.c. SIAH.asigna

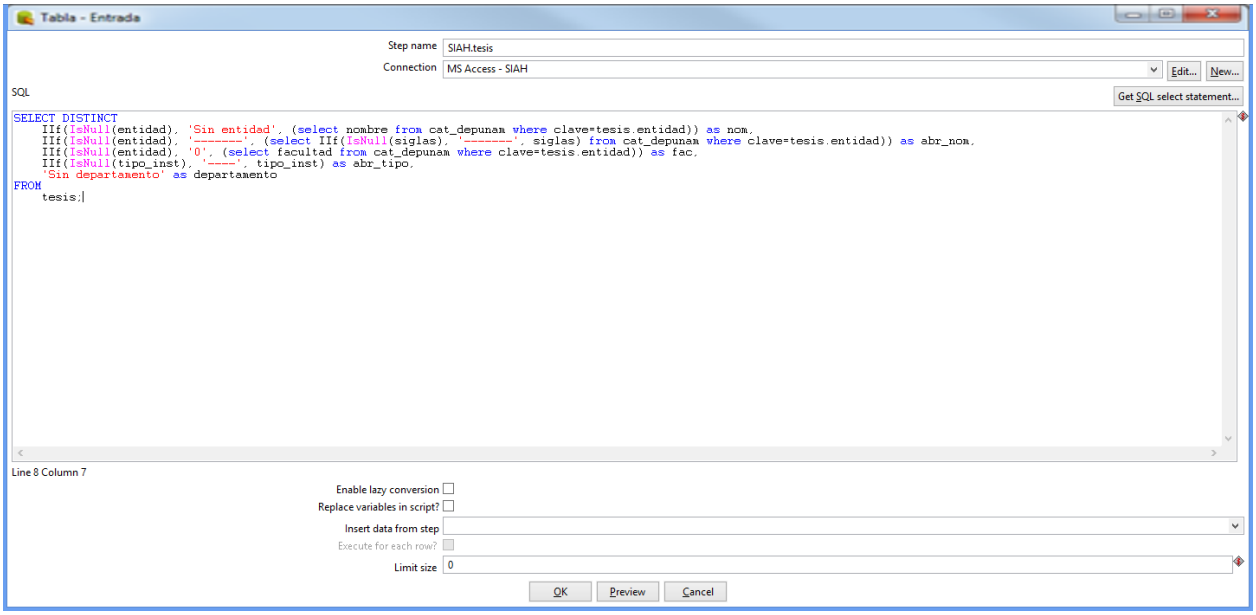


FIGURA 5.3.4.d. SIAH.tesis

En la FIGURA 5.3.4.b. muestra la consulta que obtendrá el campo “nom”, es decir, el nombre de la entidad con la función IIf, si el valor del atributo “entidad” es nulo, entonces el valor que se almacenará será “sin entidad”, si tiene otro valor guardará la entidad que se obtiene de la subconsulta.

Para obtener “abr_nom” se utilizara también la función IIf, comparando el campo “entidad”, si es nulo, en caso de ser así se escribirá en el campo “-------”, sino lo es, se almacenará el valor que se obtiene de la subconsulta de la sigla.

También determinamos si es o no una facultad la entidad que resulte de la consulta, para este caso volvemos a usar la función IIf para obtener el valor de “fac”, solo que esta vez si el valor es nulo se le asignara “0” a la variable, si tiene otro valor guardará “1” que se obtiene de la subconsulta.

Posteriormente hace falta determinar el tipo de entidad, en este caso bautizamos al campo como “abr_tipo”, y finalmente obtenemos el departamento que correspondería a la entidad que hayamos encontrado.

Como se puede observar en las FIGURAS 5.3.4.c. y 5.3.4.d. son un tanto parecidas las consultas que la FIGURA 5.3.4.b., sin embargo cambia la tabla de donde extraemos los datos y también podemos determinar si cumple cada una de las tablas con los campos que deseamos llenar.

Ordenar filas: en este paso y bien como su nombre lo dice vamos a ordenar los registros de forma ascendente, primero ordenaremos el campo “nom”, después “abr_nom”, seguido de “fac”, “abr_tipo” y por último “departamento”.

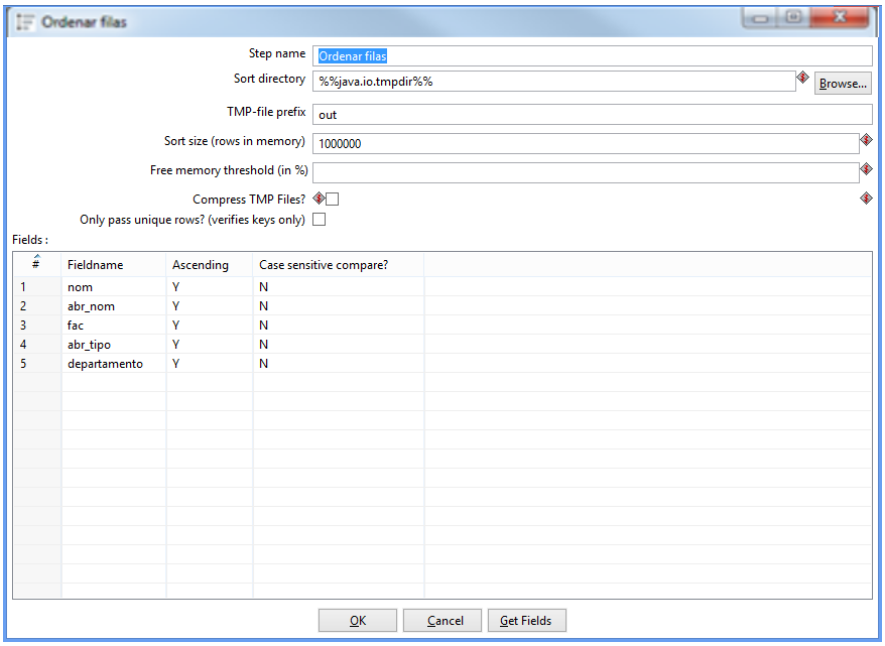


FIGURA 5.3.4.e. Ordenar filas

Filas Unicas: en este paso eliminaremos elementos duplicados del flujo de entrada, para esto los registros deben estar ordenados (se realizó en el paso anterior) porque serán comparados consecutivamente para eliminar duplicidad.

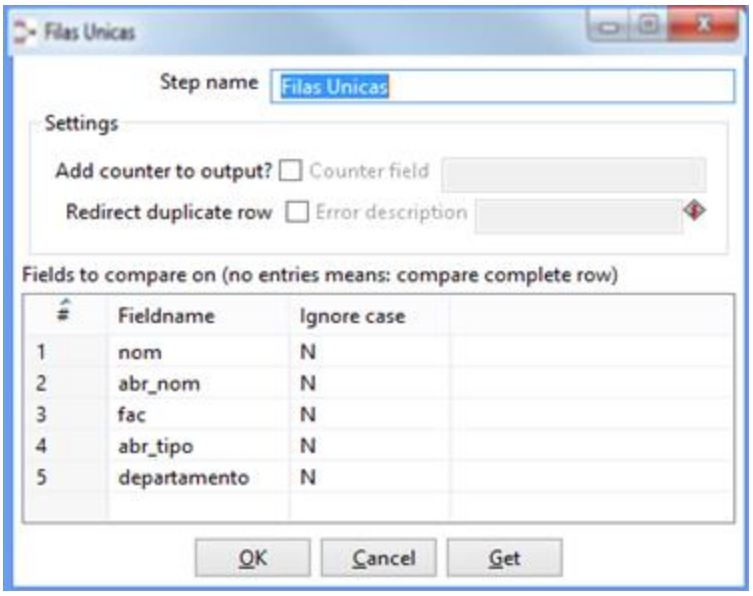


FIGURA 5.3.4.f. Filas Unicas

Mapeo de valores: los dos siguientes pasos (FIGURA 5.3.4.g. y 5.3.4.h.) únicamente mapean los valores de “tipo_abr” y “fac” respectivamente.

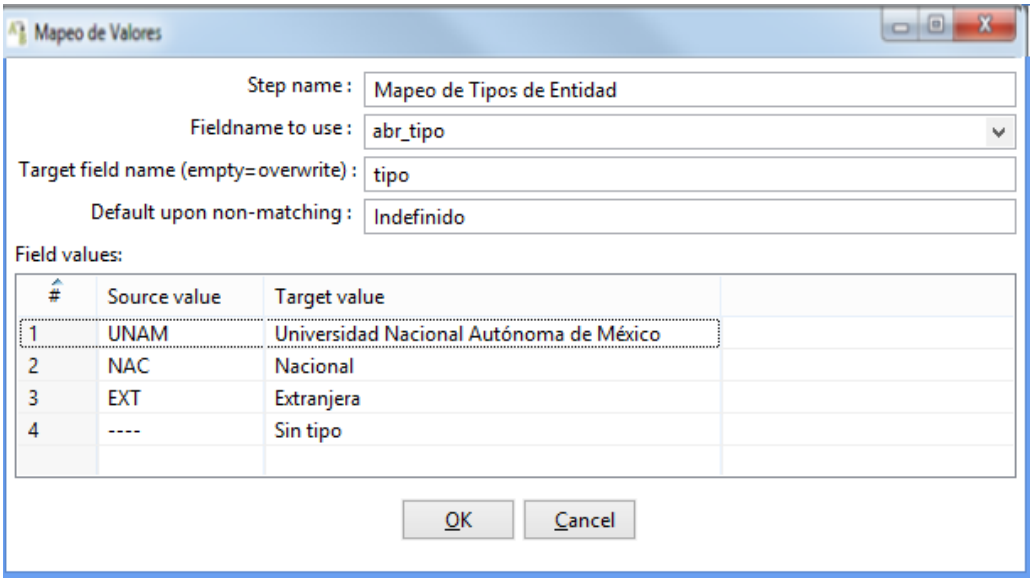


FIGURA 5.3.4.g. Mapeo de Tipos de Entidad

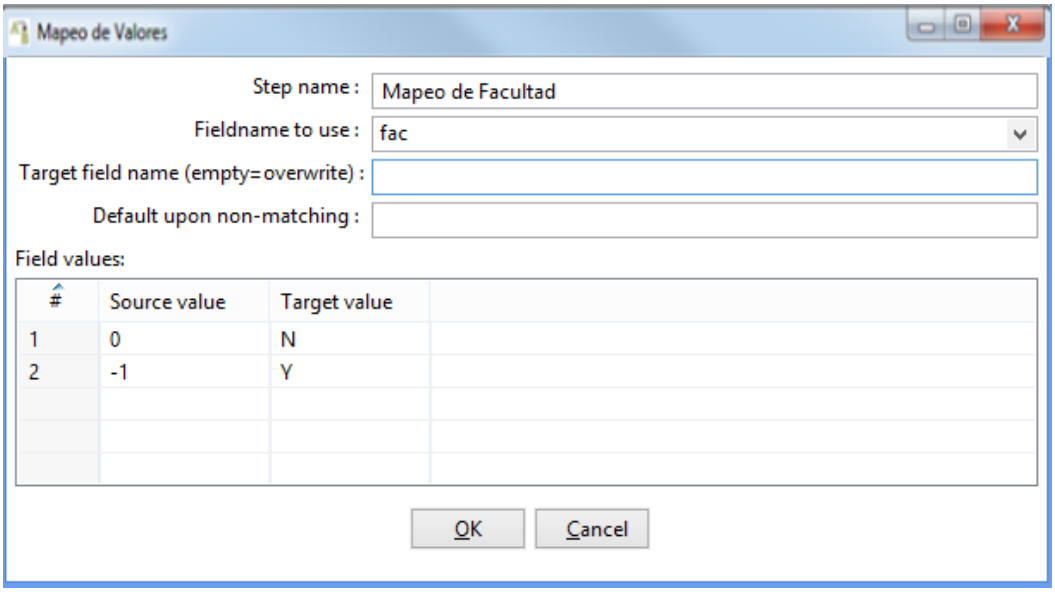


FIGURA 5.3.4.h. Mapeo de Facultad

String minúsculas: esta transformación (String Operations) cambia a minúsculas el texto del atributo "departamento", por medio de la opción "Lower/Upper" que nos permite elegir entre mayúsculas o minúsculas.

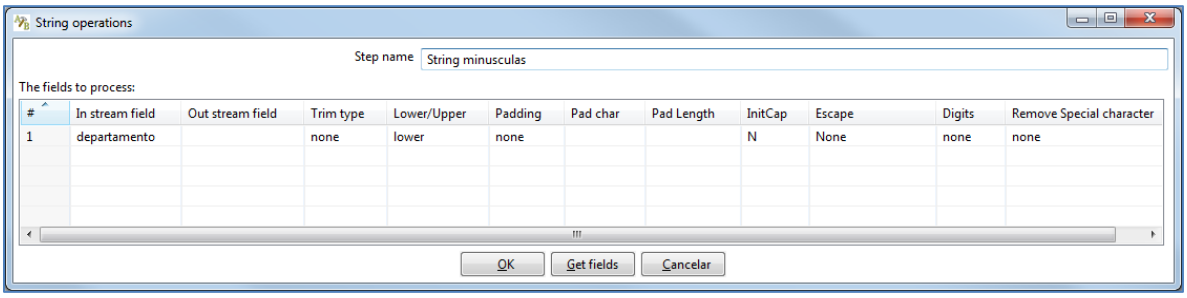


FIGURA 5.3.4.i. String minúsculas

Quitar Acentos: por medio del paso "valores de script" quitaremos acentos (Acento agudo, Acento grave, Acento circunflejo), diéresis, dos puntos, guiones, etc.

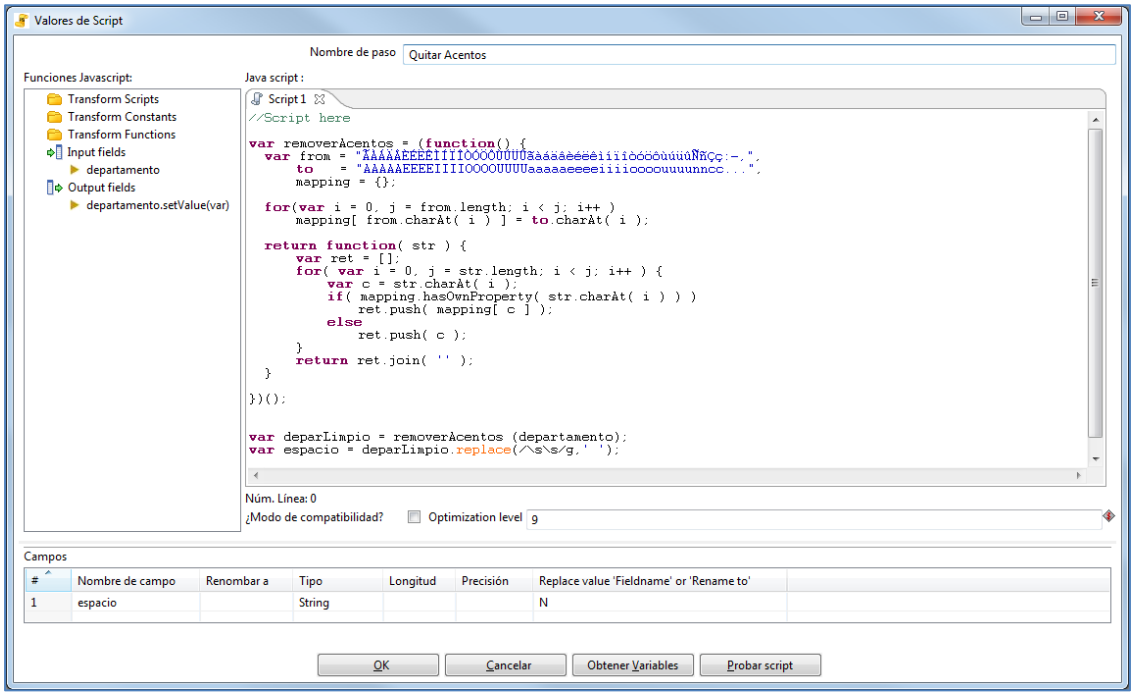


FIGURA 5.3.4.j. Quitar Acentos

Como se ve en la FIGURA 5.3.4.j. hay una variable llamada “deparLimpio “en la que se almacenará el resultado de la función “removerAcentos”. Esta es la encargada de ir revisando la cadena que tiene guardada “departamento” carácter por carácter, si encuentra un carácter que tiene la variable “from” lo reemplaza con el carácter que tiene la variable “to”.

Por último la variable espacio remplazara el doble espacio con un solo espacio, porque al estar analizando la información nos encontramos que hay cadenas que tienen doble espacio y otras sólo uno, también que hay 14 departamentos pero con diferentes formatos.

```
var removerAcentos = (function() {
  var from = "ÃÀÁÄÂÈÉËÊÌÍÏÎÒÓÖÔÙÚÜÛãàáäâèéëêìíïîòóöôùúüûÑñÇç:-,",
      to   = "AAAAAEEEEIIIIOOOOUUUUaaaaaeeeeiiiioooouuuunncc...",
      mapping = {};

  for(var i = 0, j = from.length; i < j; i++ )
      mapping[ from.charAt( i ) ] = to.charAt( i );

  return function( str ) {
      var ret = [];
      for( var i = 0, j = str.length; i < j; i++ ) {
          var c = str.charAt( i );
          if( mapping.hasOwnProperty( str.charAt( i ) ) )
              ret.push( mapping[ c ] );
          else
              ret.push( c );
      }
      return ret.join( '' );
  }

})();


var deparLimpio = removerAcentos (departamento);
var espacio = deparLimpio.replace(/\s\s/g,' ');
```

FIGURA 5.3.4.k. Quitar Acentos

Toda esta limpieza a las cadenas de texto es para tener un mejor proceso en el siguiente paso.

Ejemplo del texto sin procesar (texto sucio) y del texto procesado (texto limpio)

Texto sucio	Texto limpio
AHUNAM. Sección de Organización y Descripción	AHUNAM. Sección de Organización y Descripción
AHUNAM. Sección de Organización y Descripción	AHUNAM. Sección de Organización y Descripción
AHUNAM-SECCON DE ORGANIZACION Y DESCRIPCIONI	AHUNAM. Sección de Organización y Descripción

Mapeo de Valores ER: en esta transformación (valores de script) se usarán expresiones regulares para obtener un patrón de búsqueda de caracteres al campo "espacio" que contiene los departamentos, éstos no tienen un formato limpio, así es que se almacenarán en una nueva variable y con un formato.

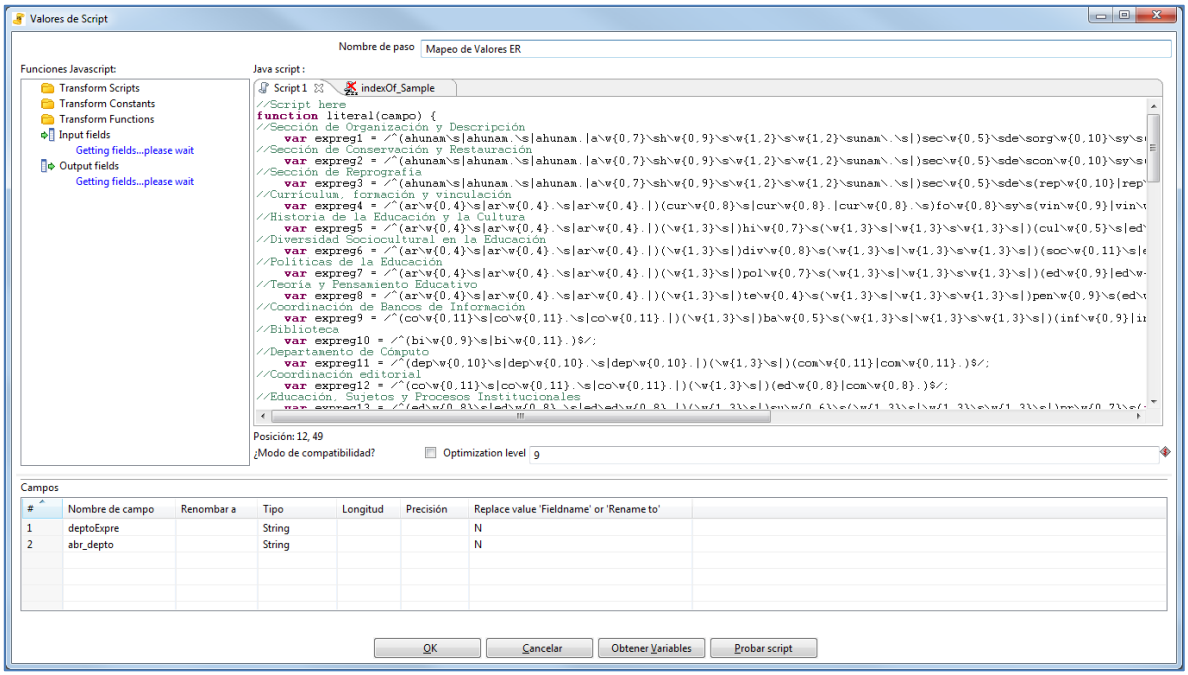


FIGURA 5.3.4.l. Mapeo de Valores ER

En FIGURA 5.3.4.l. hay una variable llamada “expre” que almacena el resultado de la función literal. Esta contiene 14 expresiones regulares que como se mencionó anteriormente son los 14 diferentes departamentos registrados.

Cada expresión será comparada con el atributo “campo” por medio de “.test()” que verifica una expresión regular y regresa un true o false si existe o no el patrón de búsqueda con la cadena.

Si es verdadera la condición se le asigna ese departamento y abreviatura a las variables, sino lo es, pasa a la siguiente condición.

```
//Script here
function literal(campo) {
//Sección de Organización y Descripción
	var expreg1 =
/^(ahunam\s|ahunam.\s|ahunam.|a\w{0,7}\sh\w{0,9}\s\w{1,2}\s\w{1,2}\sunam\.\s|)sec
\w{0,5}\sde\sorg\w{0,10}\sy\s(des\w{0,9}|des\w{0,9}.)$/;
//Sección de Conservación y Restauración
	var expreg2 =
/^(ahunam\s|ahunam.\s|ahunam.|a\w{0,7}\sh\w{0,9}\s\w{1,2}\s\w{1,2}\sunam\.\s|)sec
\w{0,5}\sde\scon\w{0,10}\sy\s(res\w{0,10}|res\w{0,10}.)$/;
//Sección de Reprografía
	var expreg3 =
/^(ahunam\s|ahunam.\s|ahunam.|a\w{0,7}\sh\w{0,9}\s\w{1,2}\s\w{1,2}\sunam\.\s|)sec
\w{0,5}\sde\s(rep\w{0,10}|rep\w{0,10}.)$/;
//Currículum, formación y vinculación
	var expreg4 =
/^(ar\w{0,4}\s|ar\w{0,4}.\s|ar\w{0,4}.|)(cur\w{0,8}\s|cur\w{0,8}.|cur\w{0,8}.\s)f
o\w{0,8}\sy\s(vin\w{0,9}|vin\w{0,9}.)$/;
//Historia de la Educación y la Cultura
	var expreg5 =
/^(ar\w{0,4}\s|ar\w{0,4}.\s|ar\w{0,4}.|)(\w{1,3}\s|)hi\w{0,7}\s(\w{1,3}\s|\w{1,3}
\s\w{1,3}\s|)(cul\w{0,5}\s|ed\w{0,9}\s)(\w{1,3}\s|\w{1,3}\s\w{1,3}\s|)(cul\w{0,5}
|cul\w{0,5}.|ed\w{0,9}|ed\w{0,9}.)$/;
//Diversidad Sociocultural en la Educación
	var expreg6 =
/^(ar\w{0,4}\s|ar\w{0,4}.\s|ar\w{0,4}.|)(\w{1,3}\s|)div\w{0,8}\s(\w{1,3}\s|\w{1,3
}\s\w{1,3}\s|)(soc\w{0,11}\s|ed\w{0,9}\s)(\w{1,3}\s|\w{1,3}\s\w{1,3}\s|)(soc\w{0,
11}|soc\w{0,11}.|ed\w{0,9}|ed\w{0,9}.)$/;
//Políticas de la Educación
	var expreg7 =
/^(ar\w{0,4}\s|ar\w{0,4}.\s|ar\w{0,4}.|)(\w{1,3}\s|)pol\w{0,7}\s(\w{1,3}\s|\w{1,3
}\s\w{1,3}\s|)(ed\w{0,9}|ed\w{0,9}.)$/;
//Teoría y Pensamiento Educativo
	var expreg8 =
```

```
/^(ar\w{0,4}\s|ar\w{0,4}.\s|ar\w{0,4}.|)(\w{1,3}\s|)te\w{0,4}\s(\w{1,3}\s|\w{1,3}
\s\w{1,3}\s|)pen\w{0,9}\s(ed\w{0,9}|ed\w{0,9}.)$/;
//Coordinación de Bancos de Información
	var expreg9 =
/^(co\w{0,11}\s|co\w{0,11}.\s|co\w{0,11}.|)(\w{1,3}\s|)ba\w{0,5}\s(\w{1,3}\s|\w{1
,3}\s\w{1,3}\s|)(inf\w{0,9}|inf\w{0,9}.)$/;
//Biblioteca
	var expreg10 = /^(bi\w{0,9}\s|bi\w{0,11}.)$/;
//Departamento de Cómputo
	var expreg11 =
/^(dep\w{0,10}\s|dep\w{0,10}.\s|dep\w{0,10}.|)(\w{1,3}\s|)(com\w{0,11}|com\w{0,11
}.)$/;
//Coordinación editorial
	var expreg12 =
/^(co\w{0,11}\s|co\w{0,11}.\s|co\w{0,11}.|)(\w{1,3}\s|)(ed\w{0,8}|com\w{0,8}.)$/;
//Educación, Sujetos y Procesos Institucionales
	var expreg13 =
/^(ed\w{0,8}\s|ed\w{0,8}.\s|ed\ed\w{0,8}.|)(\w{1,3}\s|)su\w{0,6}\s(\w{1,3}\s|\w{1
,3}\s\w{1,3}\s|)pr\w{0,7}\s(in\w{0,14}|in\w{0,14}.)$/;
//Estudios Sobre la Universidad
	var expreg14 =
/^(es\w{0,7}\s|es\w{0,7}.\s|es\ed\w{0,8}.|)(\w{1,3}\s|)so\w{0,4}\s(\w{1,3}\s|\w{1
,3}\s\w{1,3}\s|)(un\w{0,10}|in\w{0,10}.)$/;

	var mensaje;
	var mensaje2;

if(expreg1.test(campo)){
	mensaje = 'AHUNAM. Sección de Organización y Descripción';
	mensaje2 = "SOD";
}
else if(expreg2.test(campo)){
	mensaje = 'AHUNAM. Sección de Conservación y Restauración';
	mensaje2 = "SCR";
}
else if(expreg3.test(campo)){
	mensaje = 'AHUMAN. Sección de Reprografía';
	mensaje2 = "SR";
}
else if(expreg4.test(campo)){
	mensaje = 'Currículum, Formación y Vinculación';
	mensaje2 = "CFV";
}
else if(expreg5.test(campo)){
	mensaje = 'Historia de la Educación y la Cultura';
	mensaje2 = "HEC";
}
else if(expreg6.test(campo)){
	mensaje = 'Diversidad Sociocultural en la Educación';
	mensaje2 = "DSE";
}
else if(expreg7.test(campo)){
	mensaje = 'Políticas de la Educación';
	mensaje2 = "PE";
}
else if(expreg8.test(campo)){
	mensaje = 'Teoría y Pensamiento Educativo';
	mensaje2 = "TPE";
}
else if(expreg9.test(campo)){
	mensaje = 'Coordinación de Bancos de Información';
	mensaje2 = "CBI";
}
```

```
else if(expreg10.test(campo)){
       mensaje = 'Biblioteca';
       mensaje2 = "Bibl";
}
else if(expreg11.test(campo)){
       mensaje = 'Departamento de Cómputo';
       mensaje2 = "DC";
}
else if(expreg12.test(campo)){
       mensaje = 'Coordinación editorial';
       mensaje2 = "CE";
}
else if(expreg13.test(campo)){
       mensaje = 'Educación, Sujetos y Procesos Institucionales';
       mensaje2 = "ESPI";
}
else if(expreg14.test(campo)){
       mensaje = 'Estudios Sobre la Universidad';
       mensaje2 = "ESU";
}
  else {
    mensaje = espacio;
       mensaje2 = espacio;
}
return [mensaje, mensaje2];
}

var expre = literal(espacio);
var deptoExpre;
var abr_depto;

if(indexOf(expre[0], "sin departamento")!=-1) {
       deptoExpre="Sin departamento";
       abr_depto="------";
} else {
       deptoExpre=expre[0];
       abr_depto=expre[1];
}
```

FIGURA 5.3.4.m. Mapeo de Valores ER

DW_IISUE.entidades: con este paso se actualiza o inserta en la tabla de la dimensión LUGAR.

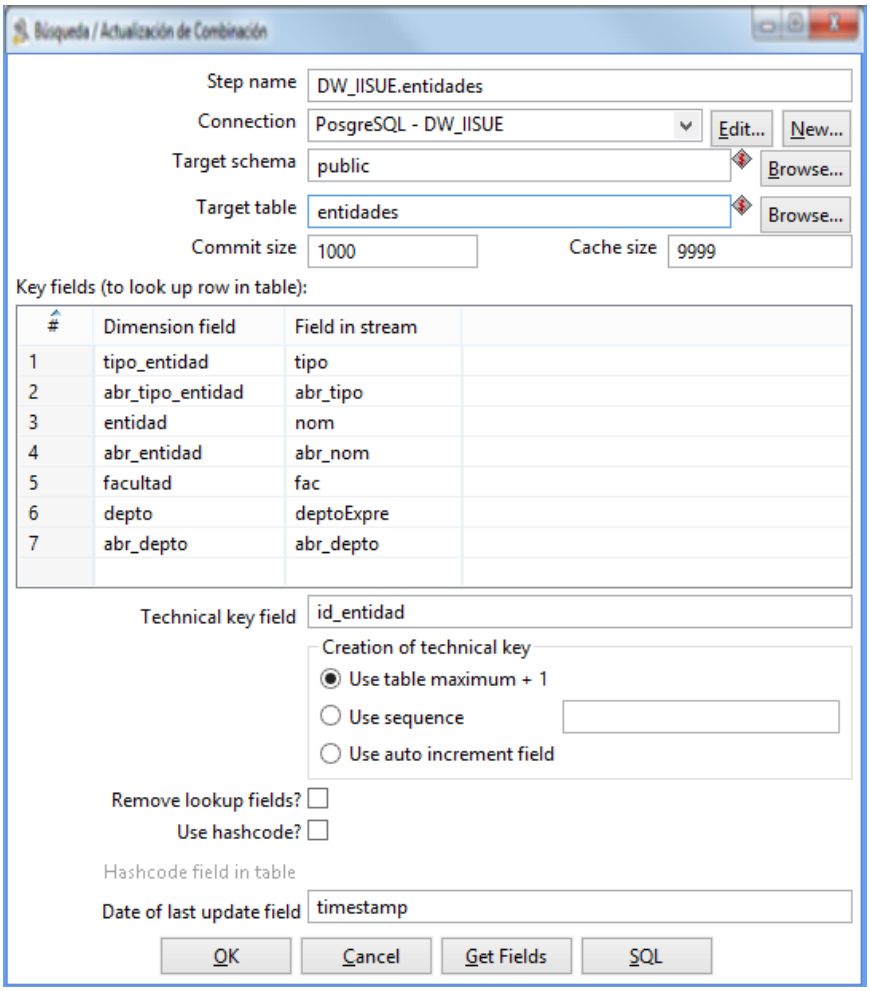


FIGURA 5.3.4.m. DW_IISUE.deptos

5.3.5. ETL Tabla de Hechos “KTR - FACT Grupos Academicos”

Como se mencionó en el capítulo 2 (2.3.3.3. Tablas de Hechos) la tabla de hechos es la tabla central de un esquema en estrella o en copo de nieve, que contiene datos cuantitativos que a su vez serán filtrados y agrupados a través de condiciones definidas en las tablas de dimensiones, dicho de otra forma contiene los valores de las medidas de negocio.

En la tabla de Hechos de “Grupos Academicos” tendremos las medidas del total de académicos (Investigadores, Profesores y Técnicos académicos) y el total de tesis con una clave primaria de las tablas de dimensiones que se vieron en los puntos anteriores (TIEMPO, TIPO DE ACADEMICO, NIVEL DE ESTUDIO, LUGAR, ESTIMULO y NACIONALIDAD), que nos servirán para la granularidad de esta tabla.

El proceso del ETL se puede observar en la FIGURA 5.3.5.a.

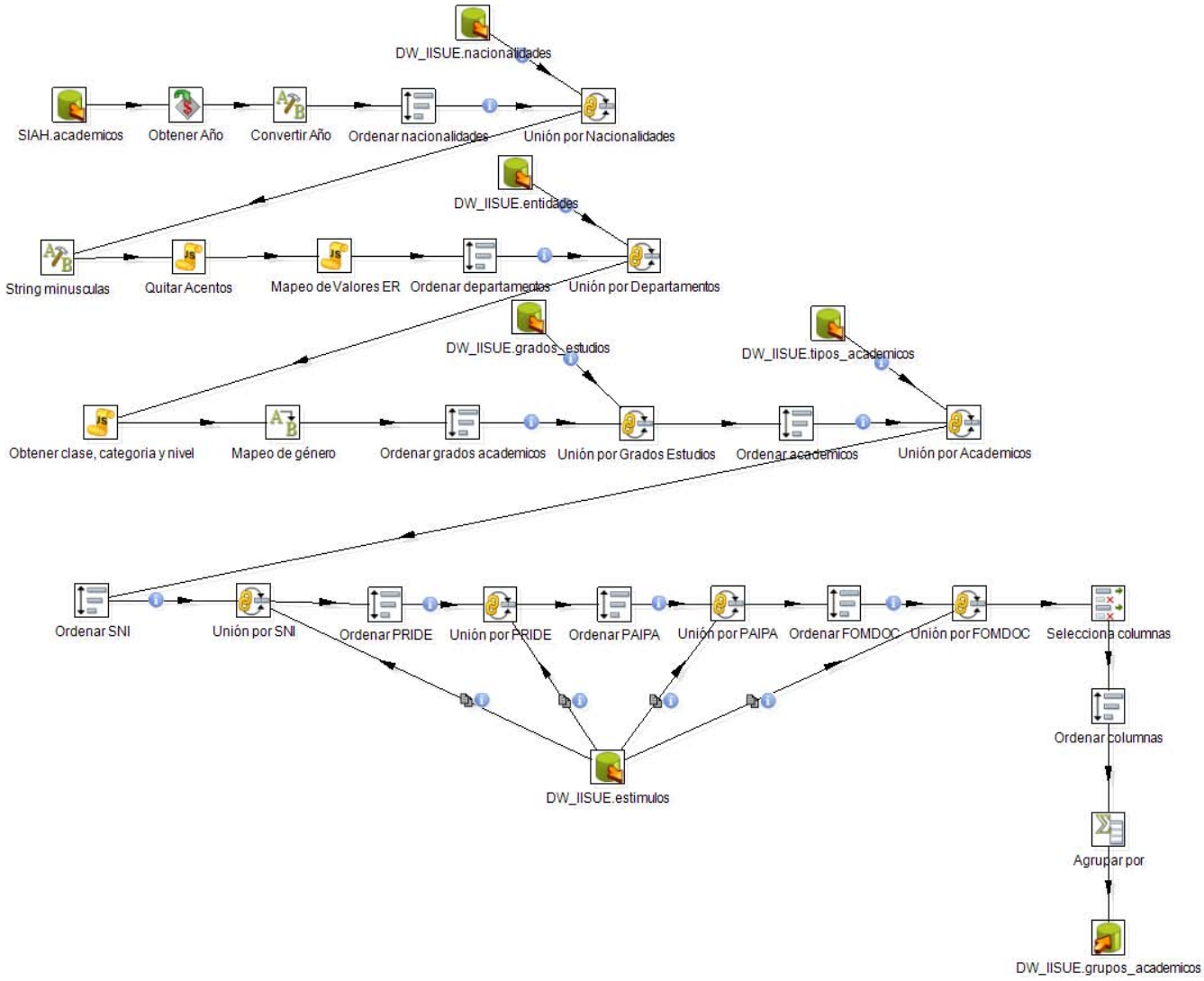


FIGURA 5.3.5.a. ETL Tabla de Hechos "KTR - FACT Grupos Academicos"

SIAH.academicos: se obtiene el tipo de académico, el sexo, el grado, la entidad, el departamento, la entidad, la abreviatura de la nacionalidad, los estímulos sni, pride, paipa y fomdoc, y el número de trabajador de la base de datos de Access.

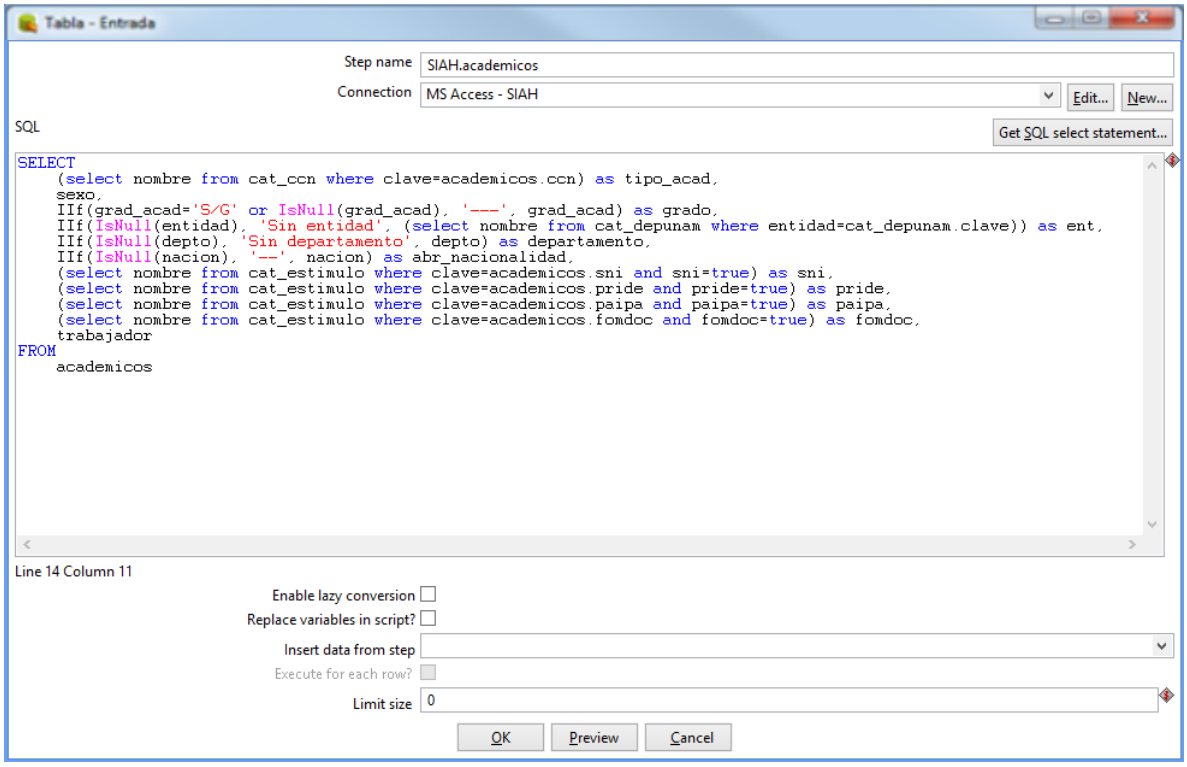


FIGURA 5.3.5.b. SIAH.academicos

Obtener Año: el campo “anio” obtiene el valor de parámetro ${ANIO}, que ya se explicó en las transformaciones anteriores de donde obtuvo el valor.

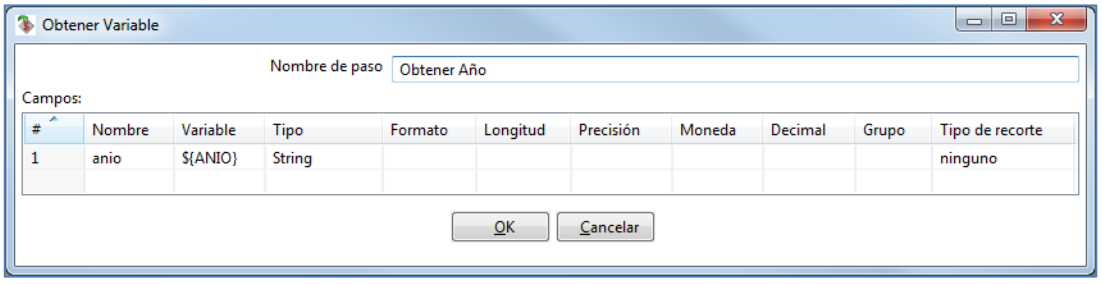


FIGURA 5.3.5.c. Obtener Año

Convertir: en el paso "String operations" se quitan los espacios en blanco que se encuentran a la izquierda, por medio de la opción Trim Type.

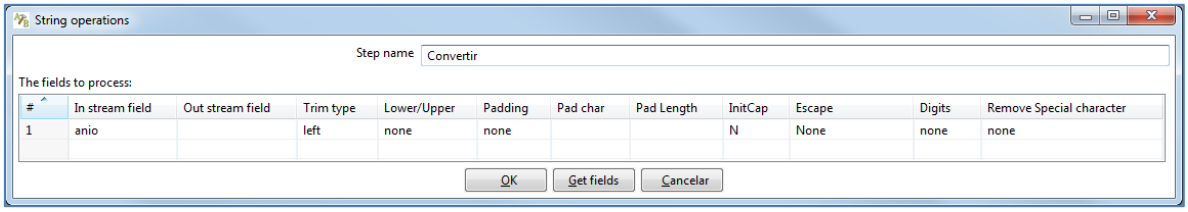


FIGURA 5.3.5.d. Convertir

Ordenar nacionalidades: Se ordenan las filas basándonos en el campo abr_nacionalidad, en orden ascendente.

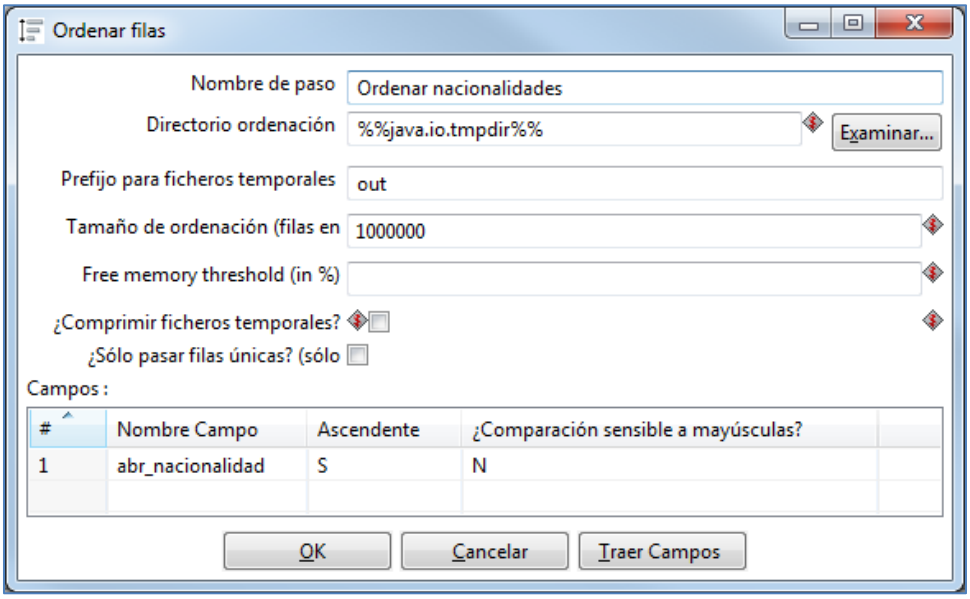


FIGURA 5.3.5.e. Ordenar nacionalidades

DW_IISUE.nacionalidades: se consulta la base de datos DW_IISUE de PostgreSQL, obteniendo los campos de "id_nacionalidad" y "abr_nacionalidad" de la tabla nacionalidades en orden ascendente.

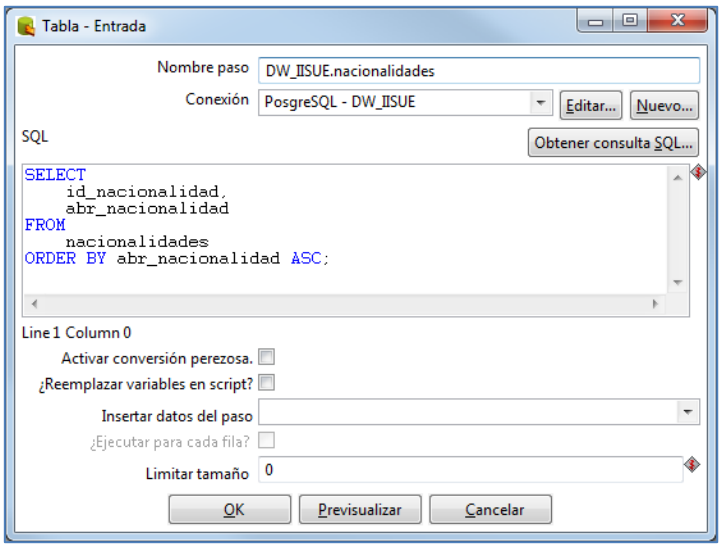


FIGURA 5.3.5.f. DW_IISUE.nacionalidades

Unión por Nacionalidades: realiza la unión entre los conjuntos de datos con los datos provenientes del paso "Ordenar nacionalidades" y del paso "DW_IISUE.nacionalidades", por medio de un campo clave en común (abr_nacionalidades).

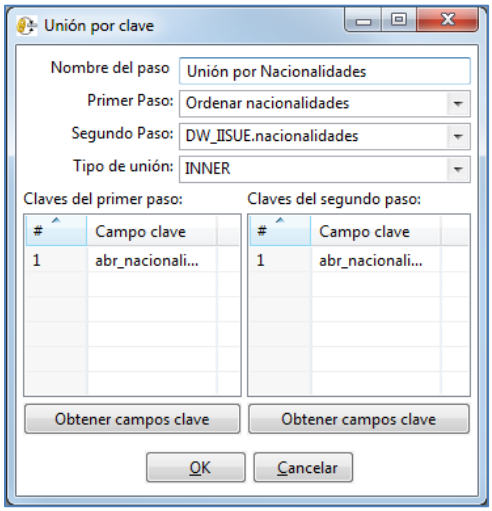


FIGURA 5.3.5.f. Unión por Nacionalidades

String minúsculas: el campo departamento se transforma en minúsculas, con la opción de Lower/Upper seleccionaremos lower.

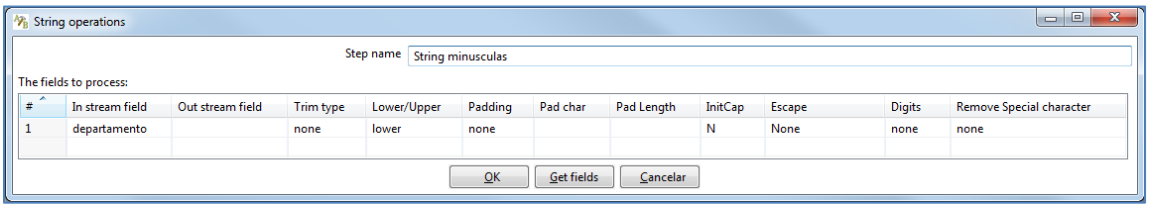


FIGURA 5.3.5.g. String minúsculas

Quitar Acentos: quita acentos (como ya se explicó en las transformaciones anteriores).

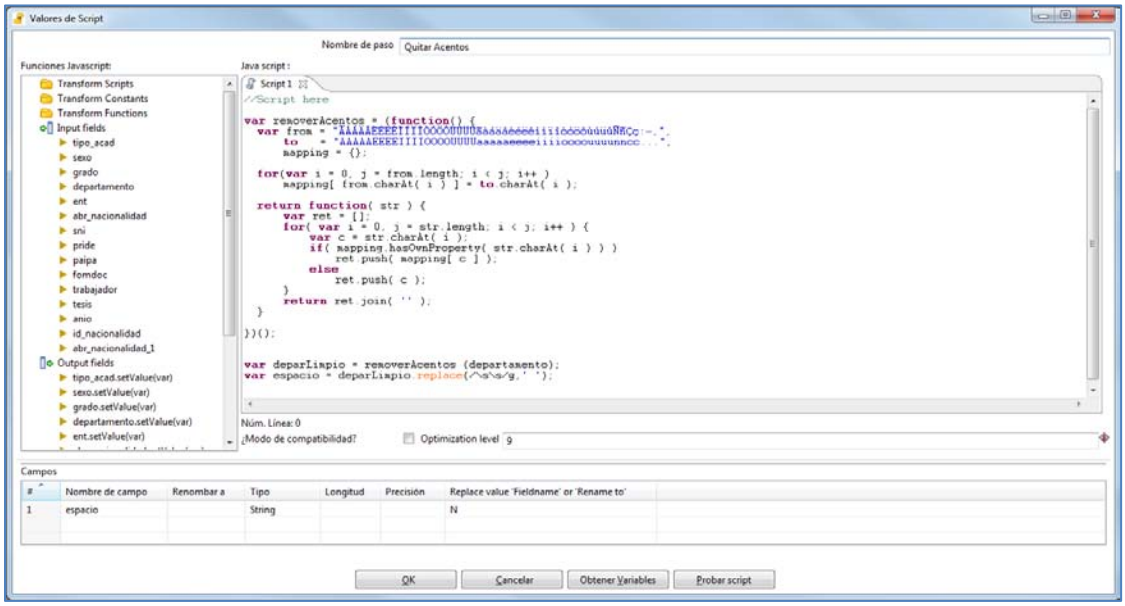


FIGURA 5.3.5.h. Quitar Acentos

Mapeo de Valores ER: como se abordó anteriormente por medio de este paso se obtiene el nombre y abreviatura de los departamentos.

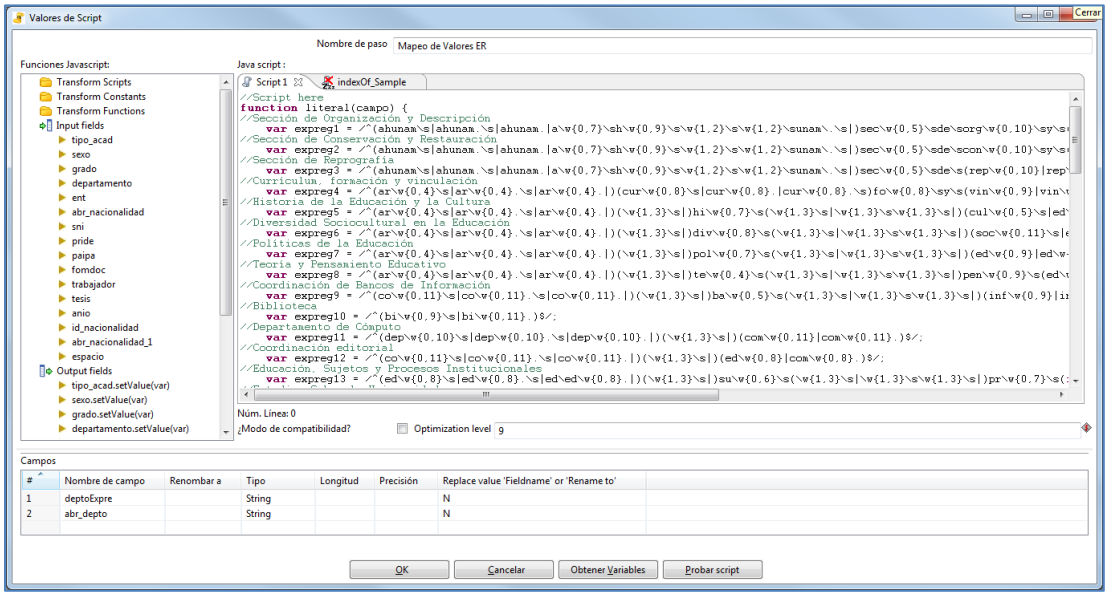


FIGURA 5.3.5.i. Mapeo de Valores ER

Ordenar departamentos: se ordenan las filas por medio del campo “ent” y “deptoExpre” de forma ascendente.

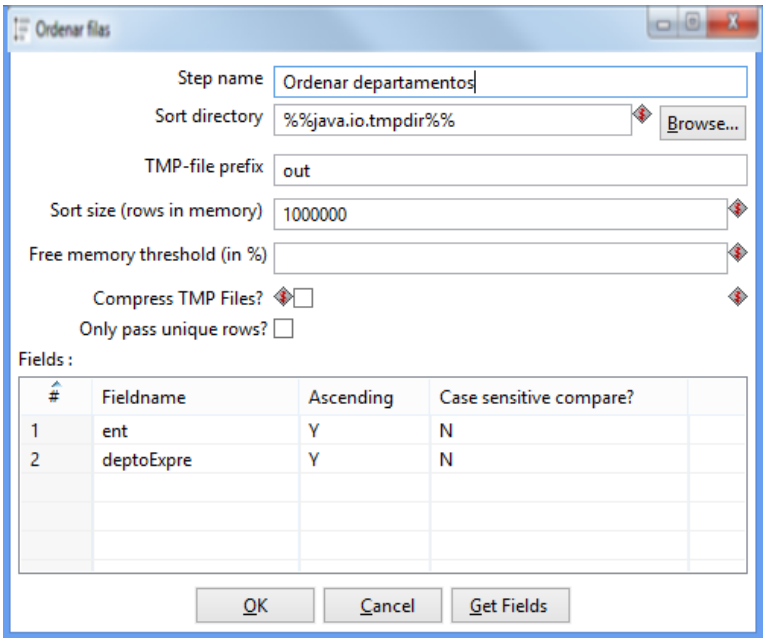


FIGURA 5.3.5.j. Ordenar departamentos

DW_IISUE.entidades: obtiene los campos “id_entidad”, “entidad” y “depto” de la base de datos DW_IISUE de PostgreSQL, ordenados por entidad y el departamento.

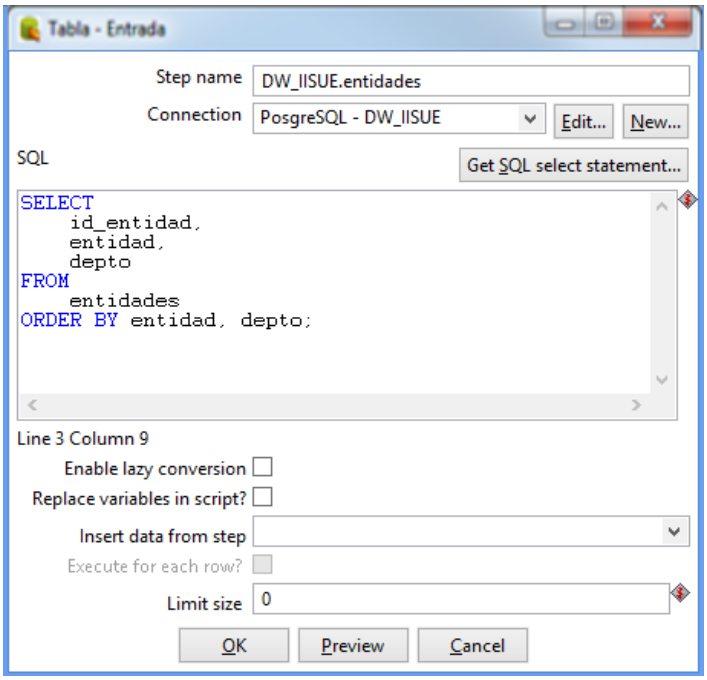


FIGURA 5.3.5.k. DW_IISUE.entidades

Unión por Departamentos: se unen los datos que provienen del paso “Ordenar departamentos” y del paso “DW_IISUE.entidades”.

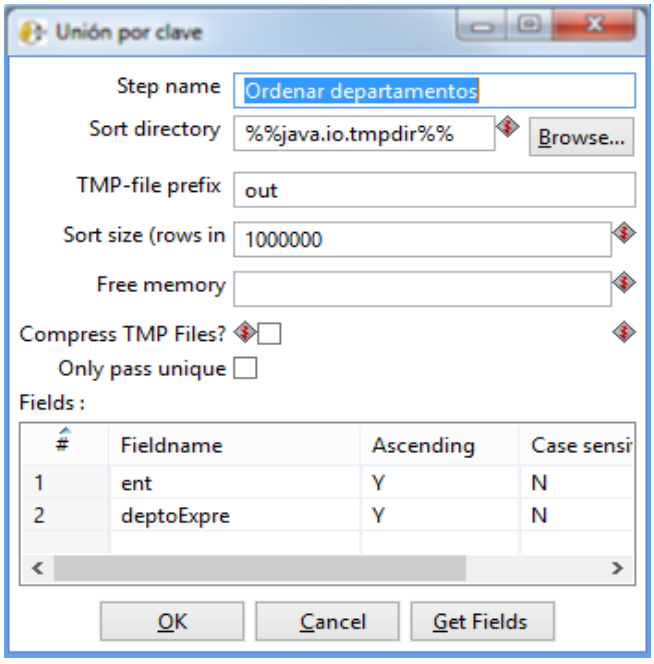


FIGURA 5.3.5.l. Unión por Departamentos

Obtener clase, categoria y nivel: en el paso “Valores de Script” se obtienen tres valores de campos (clase, categoría y nivel), que se extraerán de la columna “tipo_acad” para obtener dicha separación.

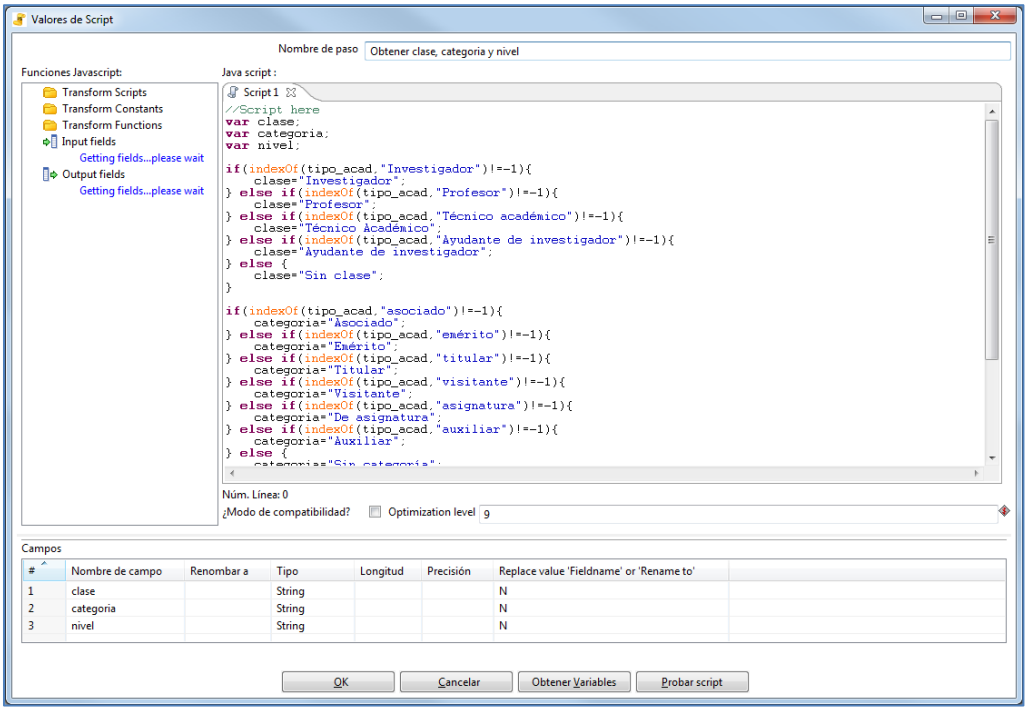


FIGURA 5.3.5.m. Obtener clase, categoria y nivel

Mapeo de género: se sustituyen el valor del campo sexo, si el valor es “M” se sustituirá por Hombre, si es “F” será igual a “Mujer” y si no trae valor será “Sin género”.

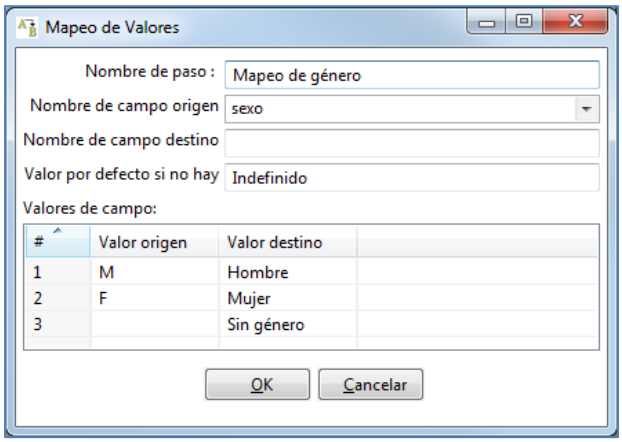


FIGURA 5.3.5.n. Mapeo de género

DW_IISUE.grados_estudios: en la consulta se seleccionan los campos de id grado de estudio y la abreviatura del grado de estudio de la tabla de grados de estudios, ordenados por el campo abreviatura del grado de estudio.

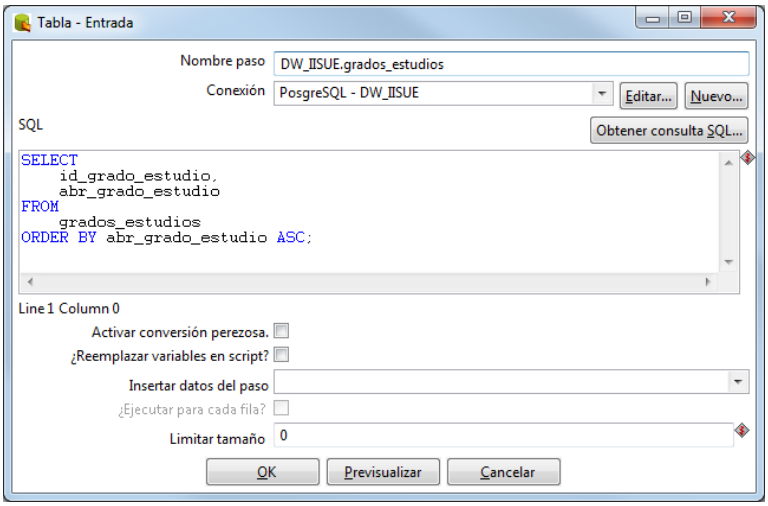


FIGURA 5.3.5.ñ. DW_IISUE.grados_estudios

Unión por Grados Estudios: se unen los datos de los pasos "Ordenar grados académicos" y de "DW_IISUE.grados_estudios".

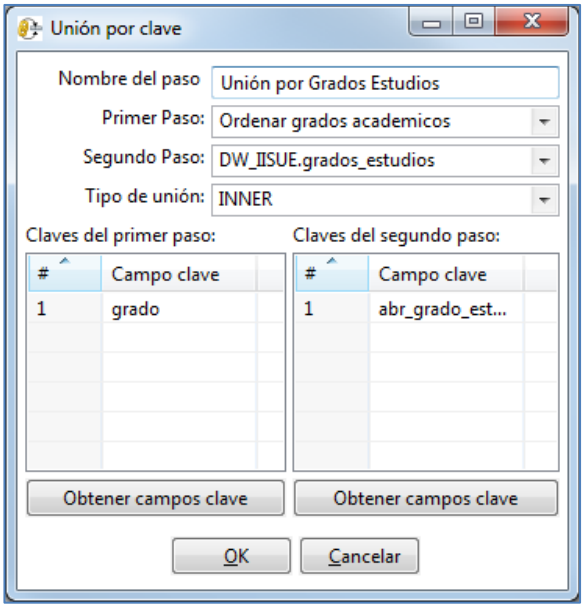


FIGURA 5.3.5.o. Unión por Grados Estudios

Ordenar académicos: ahora se ordenan de forma ascendente los campos por clase, categoría nivel y sexo.

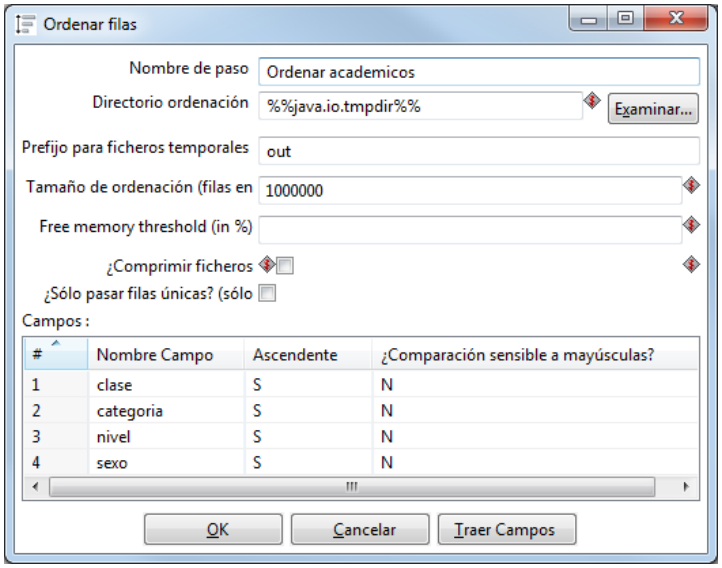


FIGURA 5.3.5.p. Ordenar académicos

DW_IISUE.tipos_academicos: se hace la consulta en donde se obtienen los campos "id_tipo_academico", "clase", "categoría", "nivel" y "genero" de la tabla de "academicos", ordenados por la clase, categoría, nivel y género de forma ascendente.

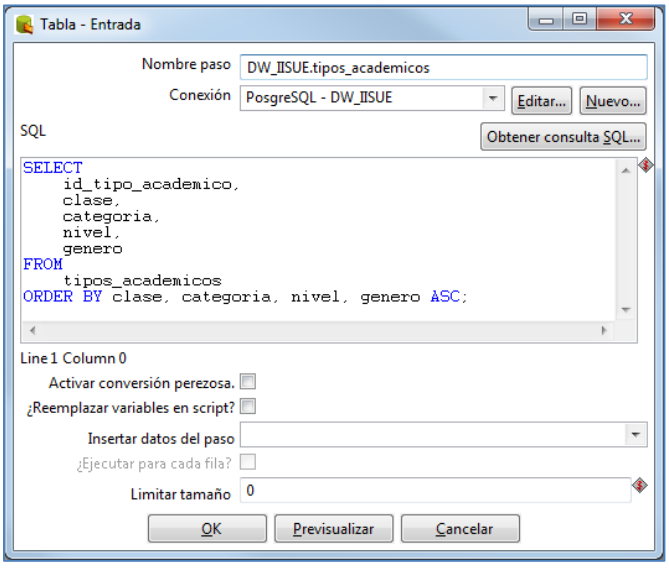


FIGURA 5.3.5.q. DW_IISUE.tipos_academicos

Unión por Academicos: ahora se unen los datos de los pasos "Ordenar académicos" y de "DW_IISUE.tipos_academicos", por medio de los campos claves que se muestran en la siguiente imagen:

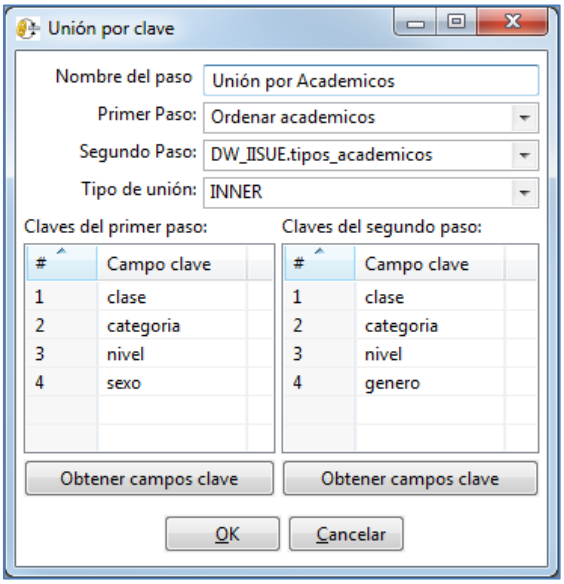


FIGURA 5.3.5.r. Unión por Academicos

Ordenar SNI: se ordena de forma ascendente los datos por el nombre del campo "sni".

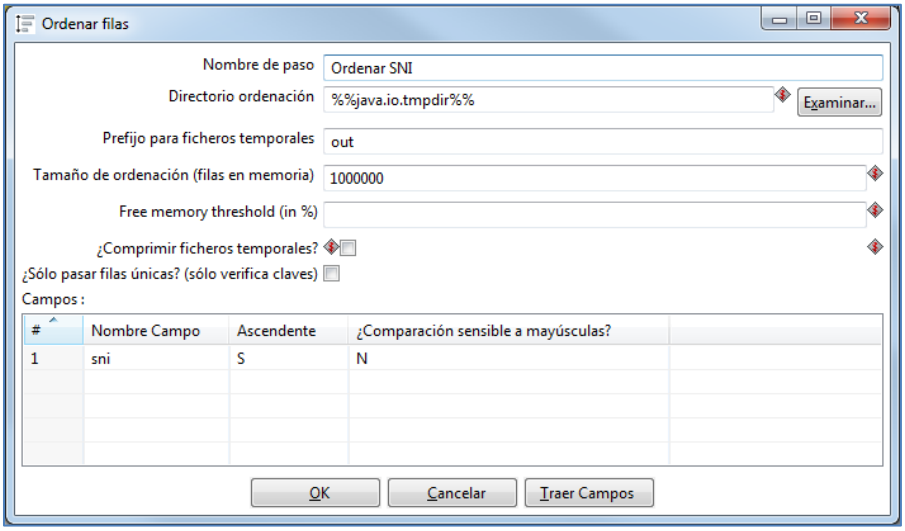


FIGURA 5.3.5.r. Ordenar SNI

DW_IISUE.estimulos: se recuperan las filas de la tabla “estimulos” de la base de datos de PostgreSQL, en forma ascendente.

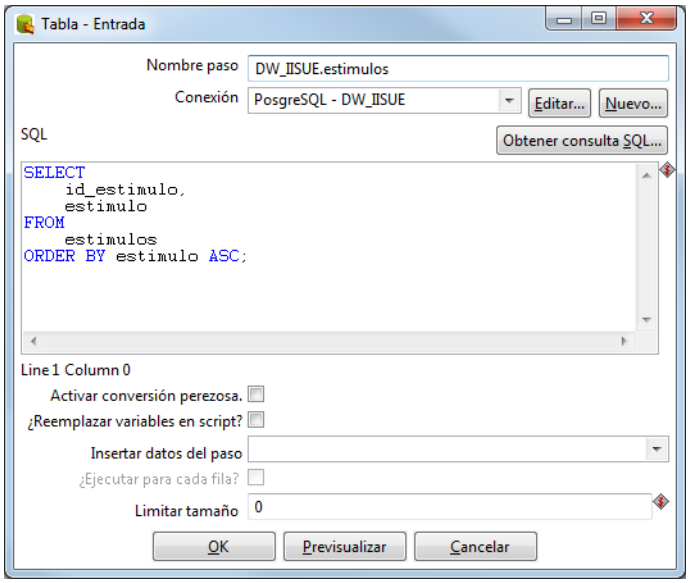


FIGURA 5.3.5.s. DW_IISUE.estimulos

Unión por SNI: se combinan los resultados de los pasos “Ordenar SIN” y “DW_IISUE.estimulos”.

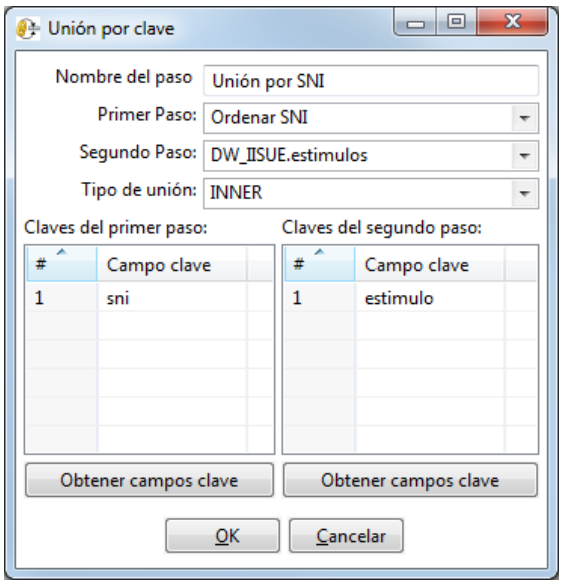


FIGURA 5.3.5.t. Unión por SNI

Ordenar PRIDE: se ordenan el conjunto de los resultados de forma ascendente del paso anterior, por medio de la columna “pride”.

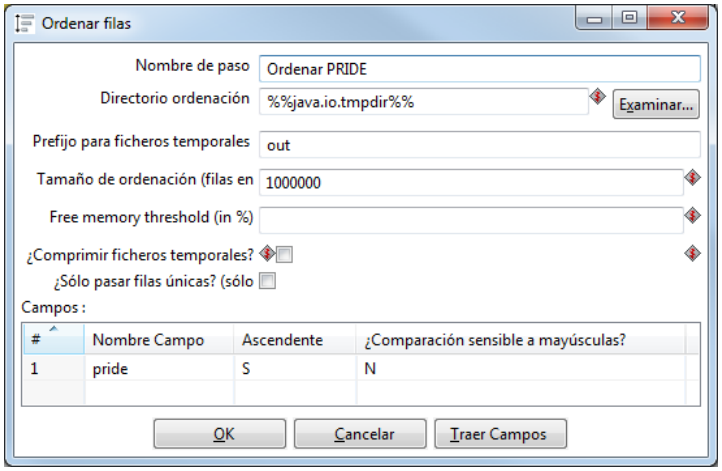


FIGURA 5.3.5.u. Ordenar PRIDE

Unión por PRIDE: se realiza la unión de los conjuntos que provienen de los pasos “Ordenar PRIDE” y de “DW_IISUE.etimulos”.

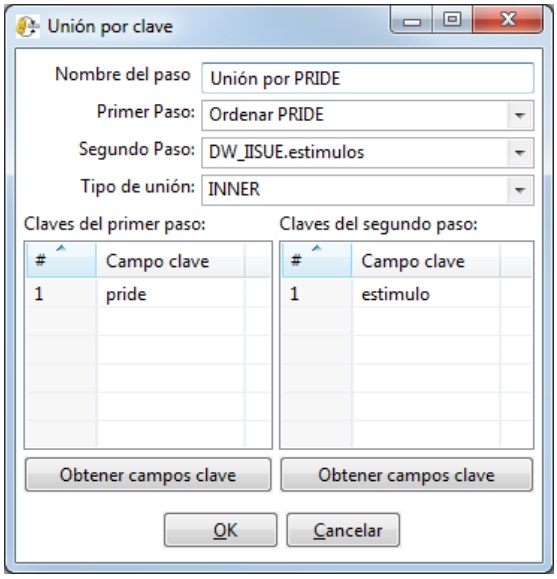


FIGURA 5.3.5.v. Unión por PRIDE

Ordenar PAIPA: en este paso se ordenan el resultado de las filas del paso anterior de forma ascendente por medio del campo “paipa”.

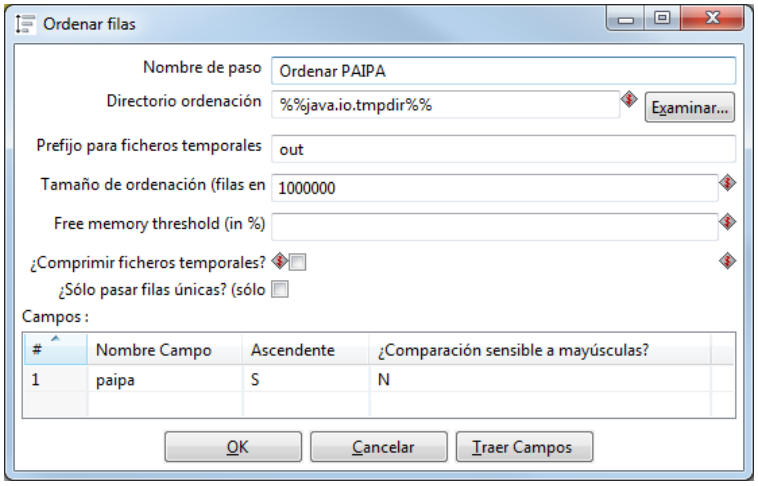


FIGURA 5.3.5.v. Unión por PRIDE

Unión por PAIPA: esta vez se unen el resultado de los dos pasos “Ordenar PAIPA” y “DW_IISUE.estimulos”.

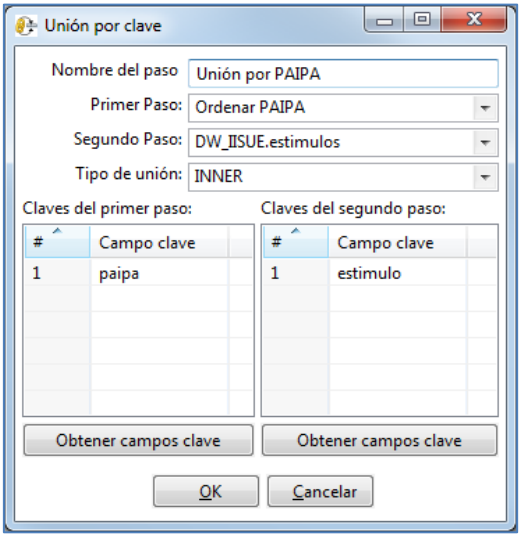


FIGURA 5.3.5.w. Unión por PAIPA

Ordenar FOMDOC: en este paso se ordenan lo datos por medio del campo "fomdoc" de forma ascendente.

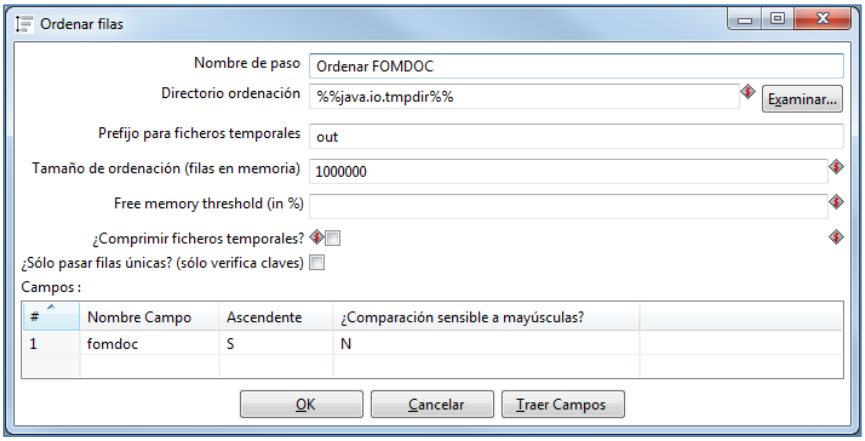


FIGURA 5.3.5.x. Ordenar FOMDOC

Unión por FOMDOC: con esta acción se unen los datos provenientes de los pasos "Ordenar FOMDOC" y "DW_IISUE.estimulos".

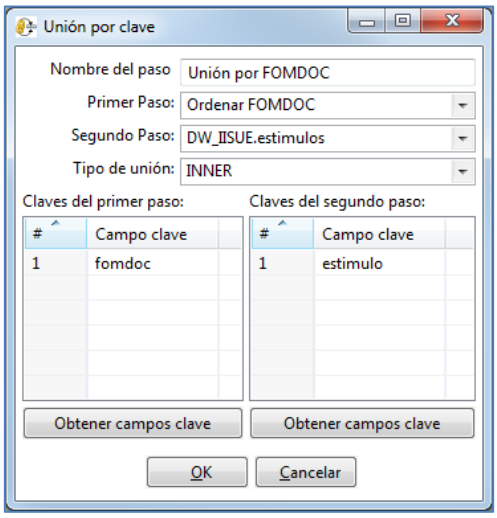


FIGURA 5.3.5.y. Unión por FOMDOC

Selecciona columnas: se hace por medio de la selección de los campos que son de nuestro interés y las renombraremos como se muestra en la siguiente imagen.

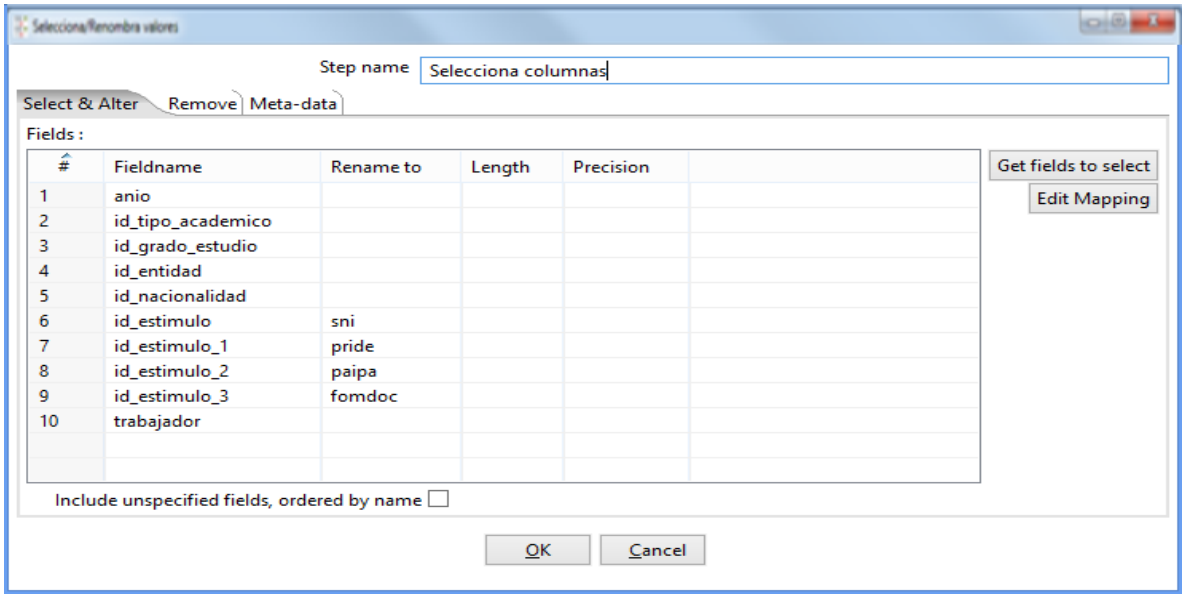


FIGURA 5.3.5.z. Selecciona columnas

Ordenar columnas: en este paso se agrupan por el orden que se muestra en la siguiente imagen.

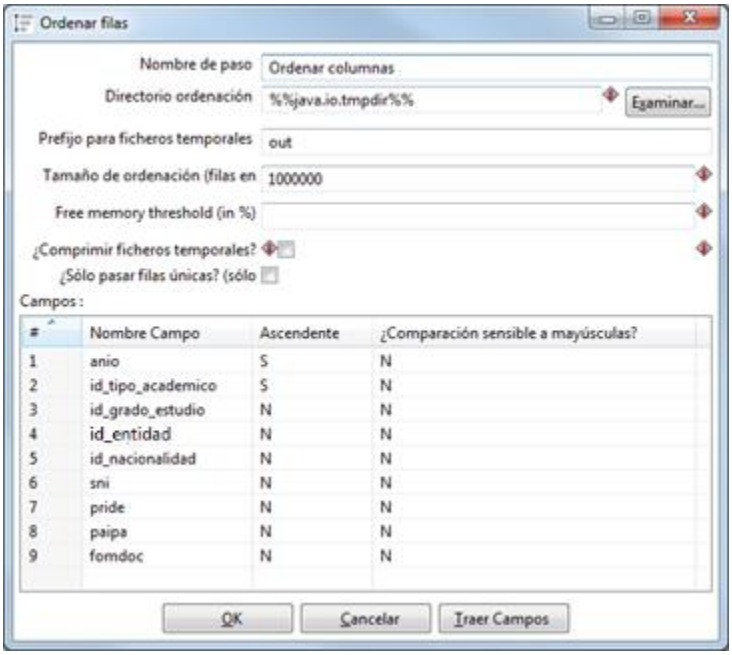


FIGURA 5.3.5.a.a. Ordenar columnas

Agrupar por: en este paso se calculan del grupo definido por los campos “anio”, “id_tipo_academico”, “id_grado_estudio”, “id_entidad”, “id_nacionalidad”, “sni”, “pride”, “paipa” y “fomdoc”.

En la sección de agregados se especifican los campos que deben de ser agregados. El total de académicos será de tipo “Número de valores distintos (N)” del campo “trabajador”, como se puede ver en la FIGURA 5.3.5.a.b.

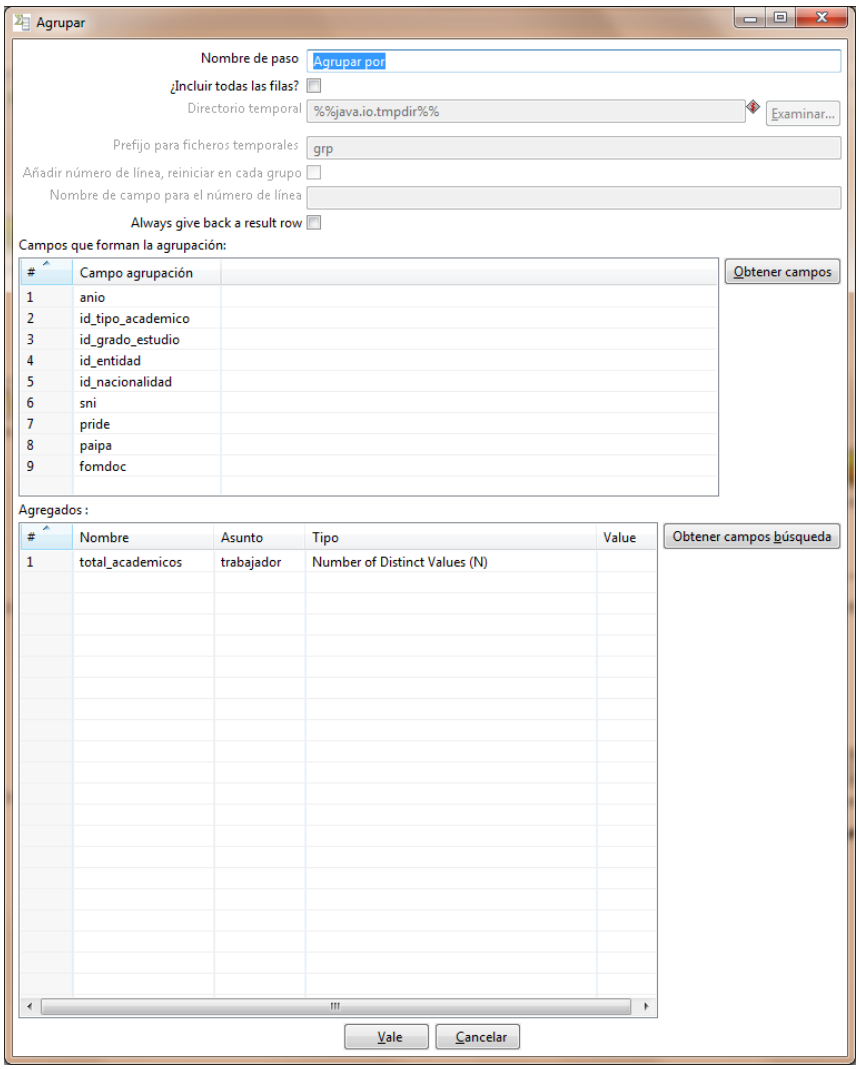


FIGURA 5.3.5.a.b. Agrupar por

DW_IISUE.grupos_academicos: en este paso se insertan las filas en la tabla "grupos_academicos", que se encuentran en la base de datos DW_IISUE de PostgresSQL.

En la pestaña "Database fields" se especifican los campos provenientes del paso anterior con los campos en dónde queremos que se inserten en la tabla, así como se muestra en la FIGURA 5.3.5.a.c.

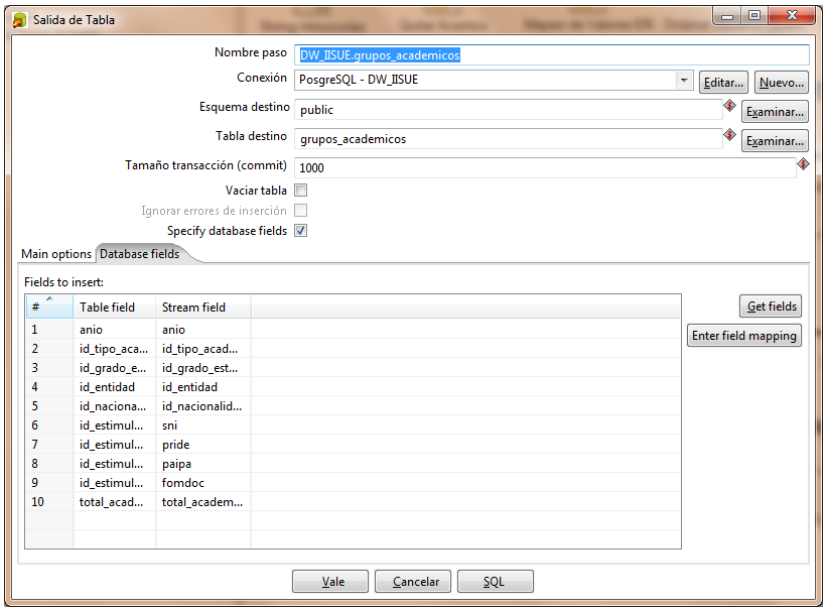


FIGURA 5.3.5.a.c. DW_IISUE.grupos_academicos

5.3.6. ETL JOB

Los Jobs son usados para un control de flujo, por lo general consiste en una serie de transformaciones. Por medio del job crearemos una secuencia de actividades que brindan un orden de ejecución automático de las transformaciones que hemos diseñado, como se observa en la FIGURA 5.3.6.a.

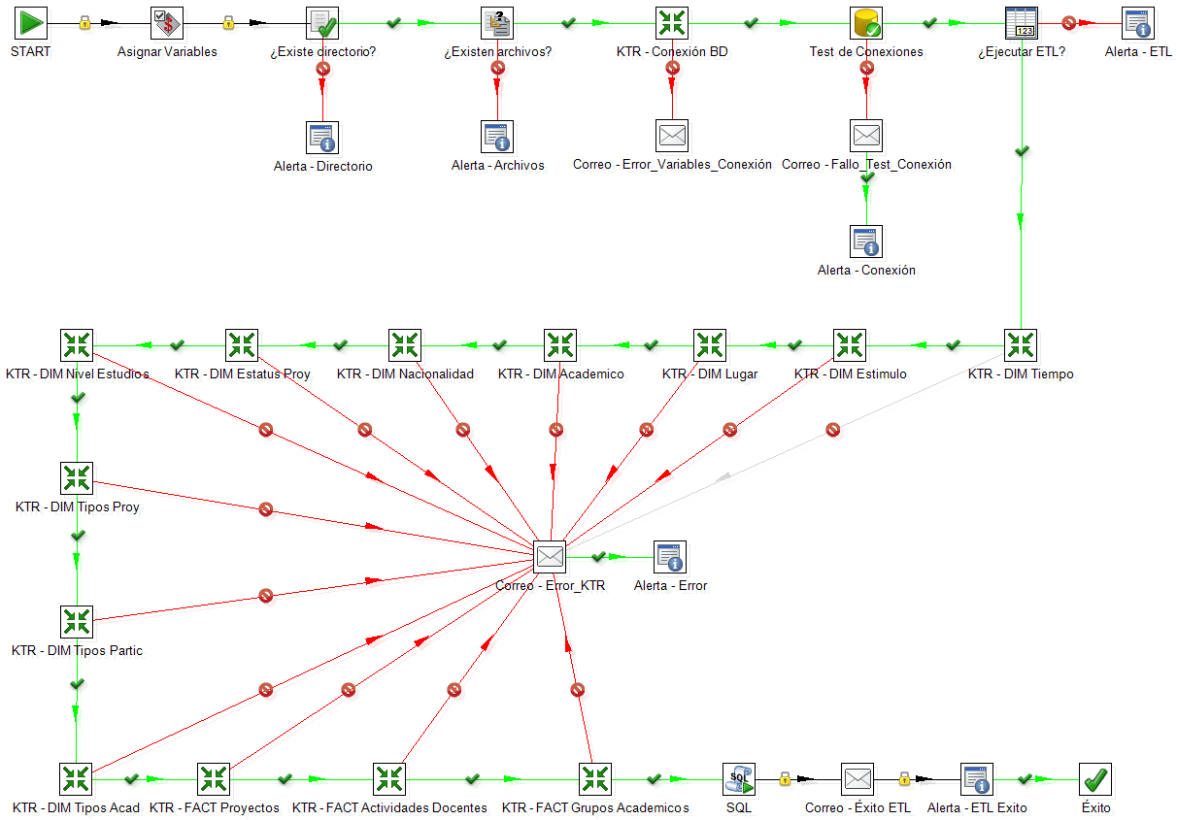


FIGURA 5.3.6.a. ETL JOB

Asignar Variables: es una entrada de trabajo de variables establecidas, en la cual especificamos el nombre de las variables y su valor.

En este caso tendremos tres variables, el de "${RUTA}" que establece la dirección del archivo de la base de datos de Access, el "${ARCHIVO}" nos servirá para obtener el nombre del archivo de la base de datos que vamos a usar para las transformaciones y por último la variable "${CADENA}" que se le asignará a la cadena de conexión de la BD. Tanto ARCHIVO como CADENA en este paso se les dará el valor por default de "-", en otros pasos se les asignará el valor que se comentó.

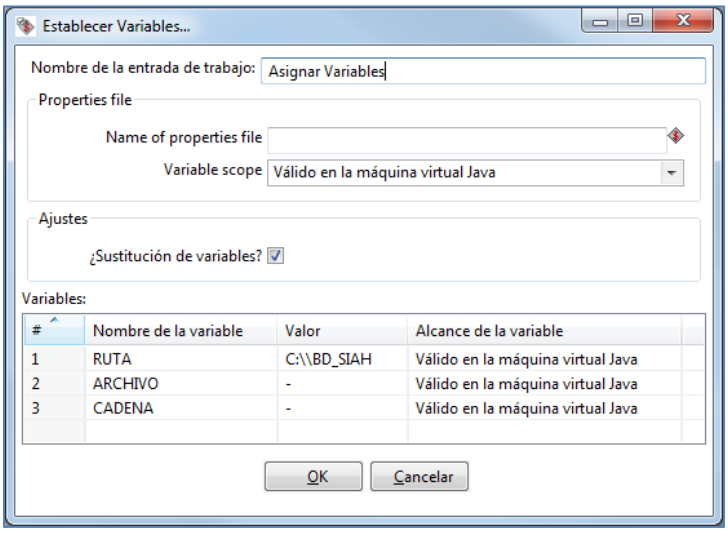


FIGURA 5.3.6.b. Asignar Variables

¿Existe directorio?: el paso "Comprobar si existen los archivos" se utiliza para comprobar si el parámetro "${RUTA}" contiene el directorio del archivo, si existe seguirá al siguiente paso "KTR - Conexión BD", sino existe pasará al paso "Mostrar Información" y se terminará el proceso del JOB.

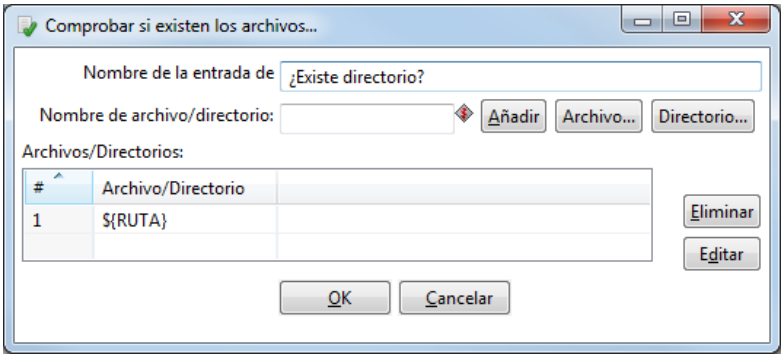


FIGURA 5.3.6.c. ¿Existe directorio?

Alerta - Directorio: esta entrada de trabajo le permite mostrar un cuadro de diálogo con un mensaje. Como se muestra en la FIGURA 5.3.6.d. se configura el nombre del paso, el título del mensaje y el Cuerpo del mensaje, que en este caso es: ¡No se encontró directorio!. Cree la carpeta "BD_SIAH" en la unidad "C:" y vuelva a ejecutar el JOB por favor.

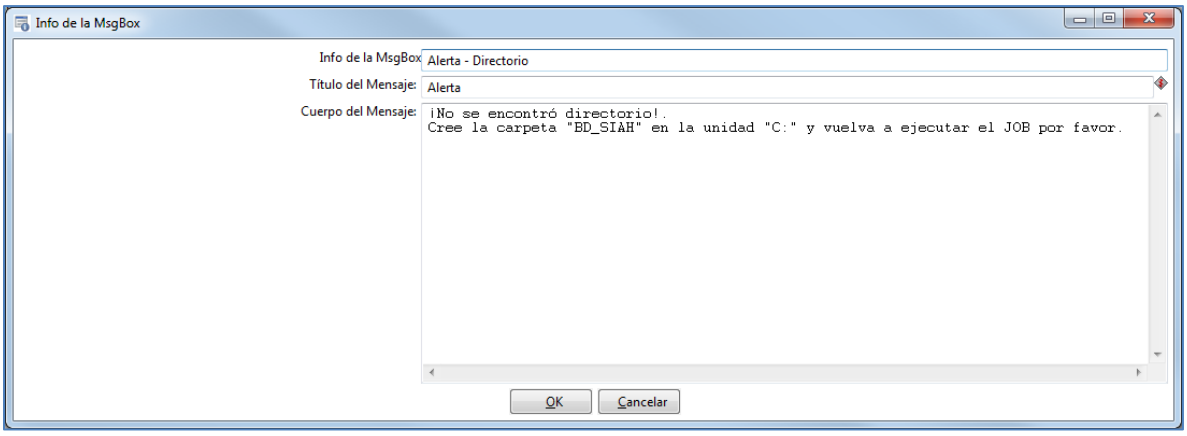


FIGURA 5.3.6.d. Alerta - Directorio

KTR - Conexión BD: en este paso se especifica la transformación “KTR - Conexión BD” que se ejecutará.

En la pestaña “Especificación de la transformación” hay que seleccionar la opción de “nombre de la transformación” y se pone en el primer campo el nombre KTR - Conexión BD, que es la transformación. En el segundo campo se pone “/ETL - dw_iisue”, que es el nombre del repositorio, en donde se encuentran todas las transformaciones y el JOB, como se muestra en la FIGURA 5.3.6.e.

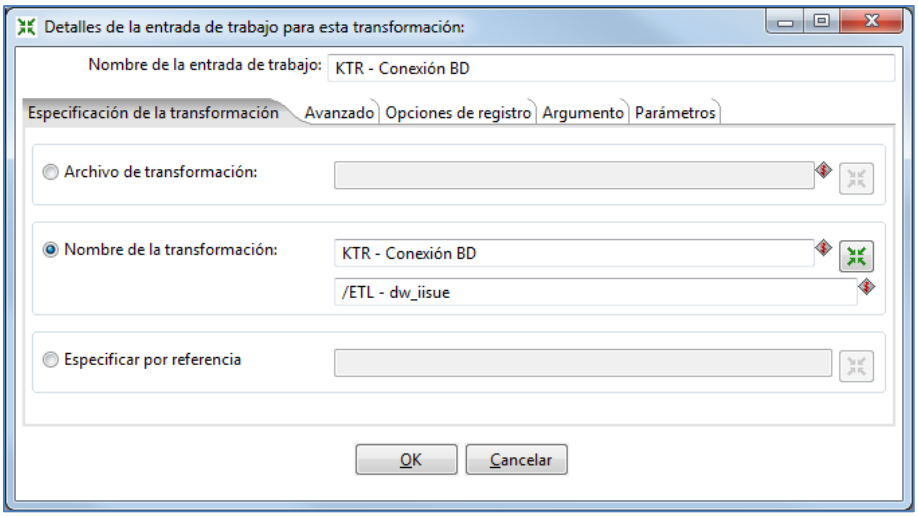


FIGURA 5.3.6.e. KTR - Conexión BD (a)

Como se muestra en la FIGURA 5.3.6.g. en la pestaña “Opciones de registro” se especifican las opciones para crear los logs, así tendremos el registro del comportamiento de las transformaciones, si funcionó correctamente o si es que falló.

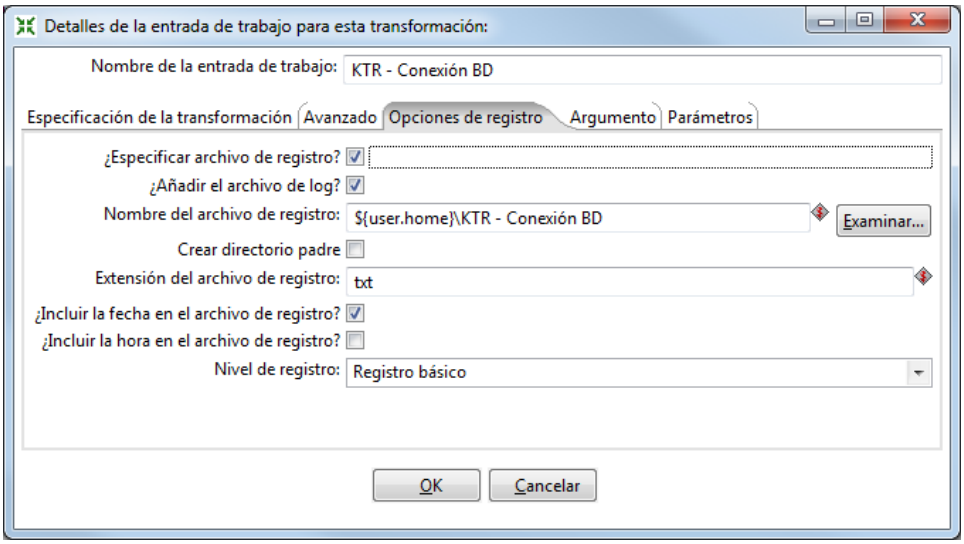


FIGURA 5.3.6.g. KTR - Conexión BD (b)

Test de Conexiones: en la entrada de trabajo “Verificar conexiones a BD…” se verifica la conectividad de las bases de datos MS Access - siah y PostgreSQL - dw_iisue con un tiempo de espera de 1 seg cada una. Si alguna de las conexiones falla se irá al paso “Correo - Error_Conexion” y terminará el proceso del JOB.

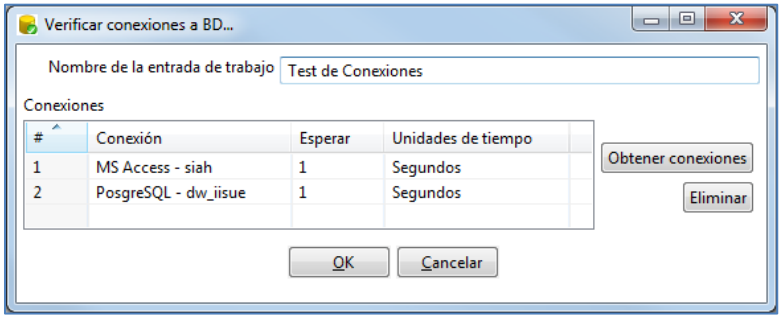


FIGURA 5.3.6.h. Test de Conexiones

Correo - Fallo_Test_Conexión: con el elemento "detalles del trabajo de correo" se envía un mail, en el caso de que se produzca un error al verificar las conexiones a las bases de datos, en donde se manda en adjunto el log. El paso lo configuraremos como se muestra en las siguientes imágenes.

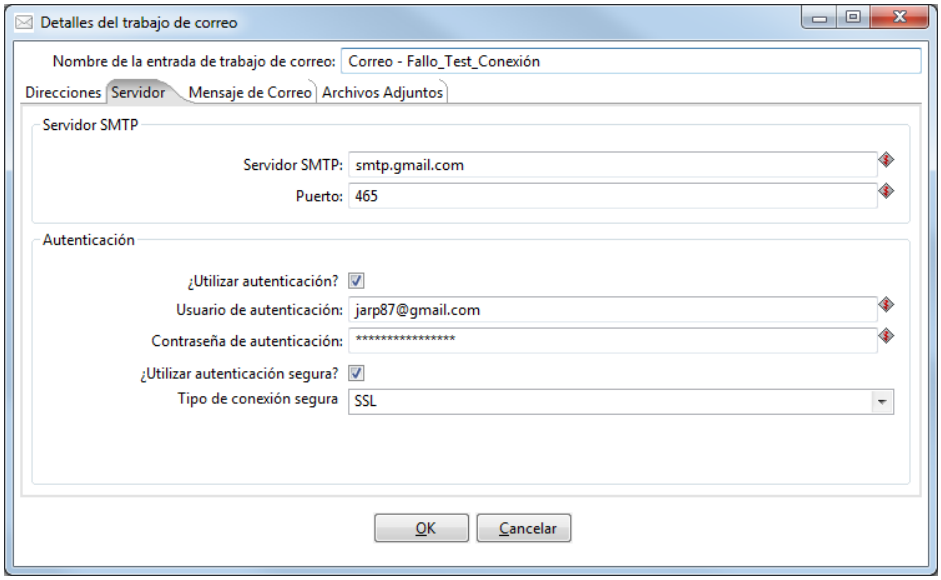


FIGURA 5.3.6.i. Correo - Fallo_Test_Conexión (a)

En la FIGURA 5.3.6.i escribimos la configuración de un servidor de correos utilizando el protocolo STMP[14] (Simple Mail Transfer Protocol) al especificar el puerto de salida y el correo que se utilizará para enviarlo.

[14] SMTP (Protocolo para la transferencia simple de correo electrónico) es un protocolo de red utilizado para el intercambio de mensajes de correo electrónico entre computadoras u otros dispositivos.

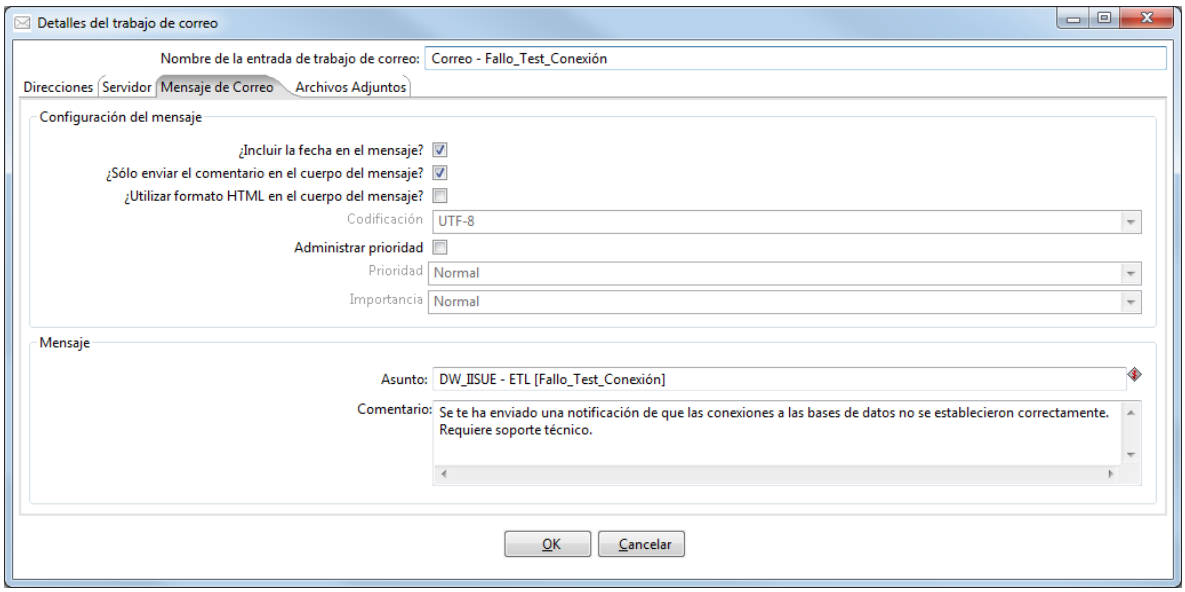


FIGURA 5.3.6.j. Correo - Fallo_Test_Conexión (b)

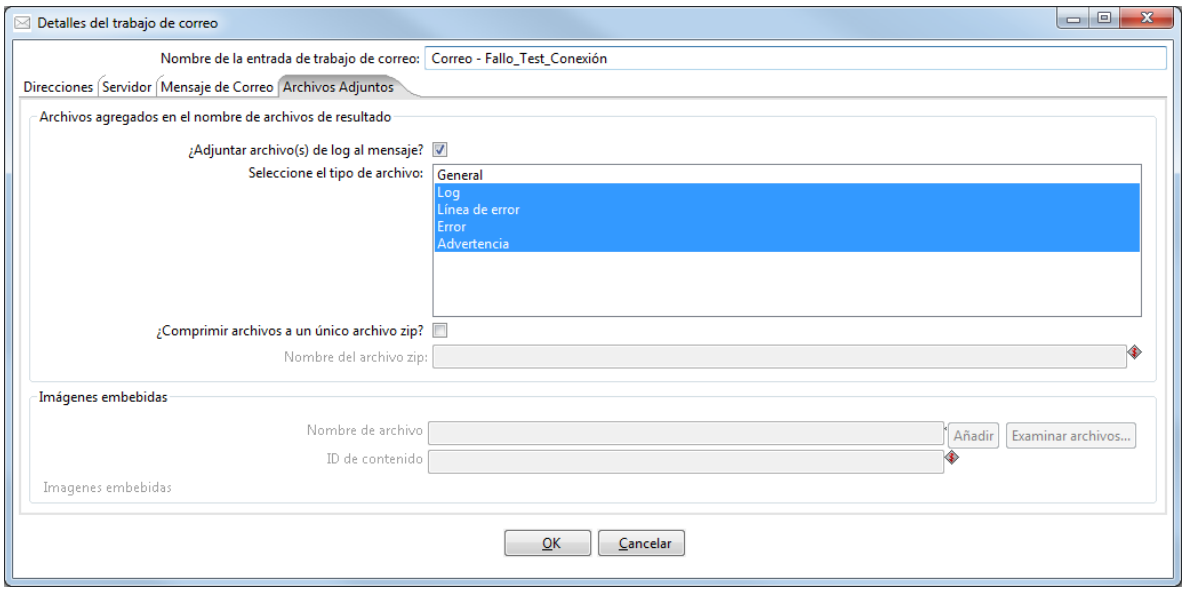


FIGURA 5.3.6.k. Correo - Fallo_Test_Conexión (c)

En la FIGURA 5.3.6.j. configuramos el asunto y el mensaje que se enviaran al administrador. Por ultimo en la FIGURA 5.3.6.k. seleccionaremos el tipo de archivo que se enviara en el correo.

Alerta – Conexión: si el Test de Conexiones tiene problemas, mandará un correo como se explicó en el paso "Correo - Fallo_Test_Conexión" y después mostrará una ventana al usuario con un mensaje.

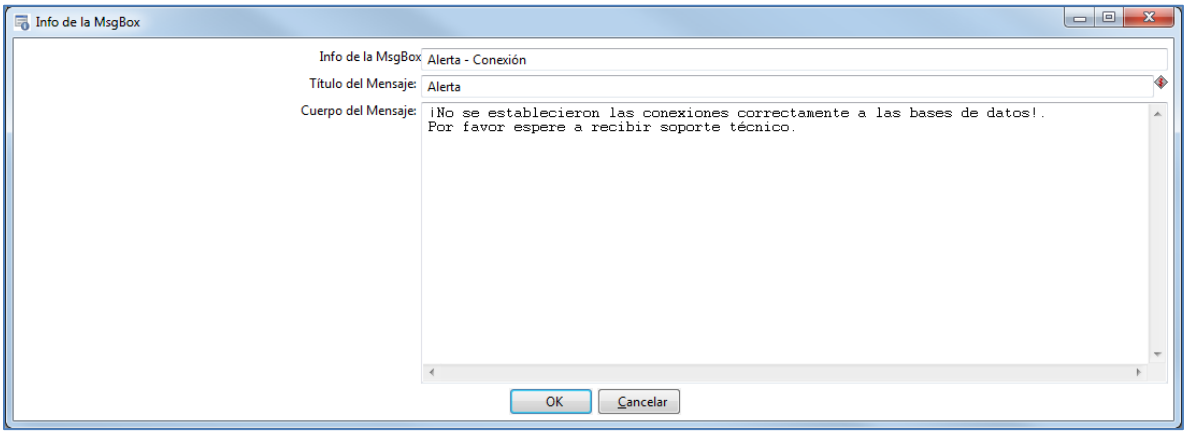


FIGURA 5.3.6.l. Alerta – Conexión

¿Ejecutar ETL?: en este paso verificaremos si el archivo que se está cargando ya fue ejecutado anteriormente y guardado.

Si el campo db_access de la tabla dw_iisue de la base de datos PostgreSQL es menor o igual a cero, cuando el campo db_access es igual a "${ARCHIVO}" (contiene el nombre del archivo que fue seleccionado), ejecutará el ETL, siguiendo por la transformación "KTR - DIM Tiempo". Si el valor de la consulta es mayor a cero, quiere decir que ese archivo ya se cargó en el DWH y ejecutará el paso "Alerta – ETL".

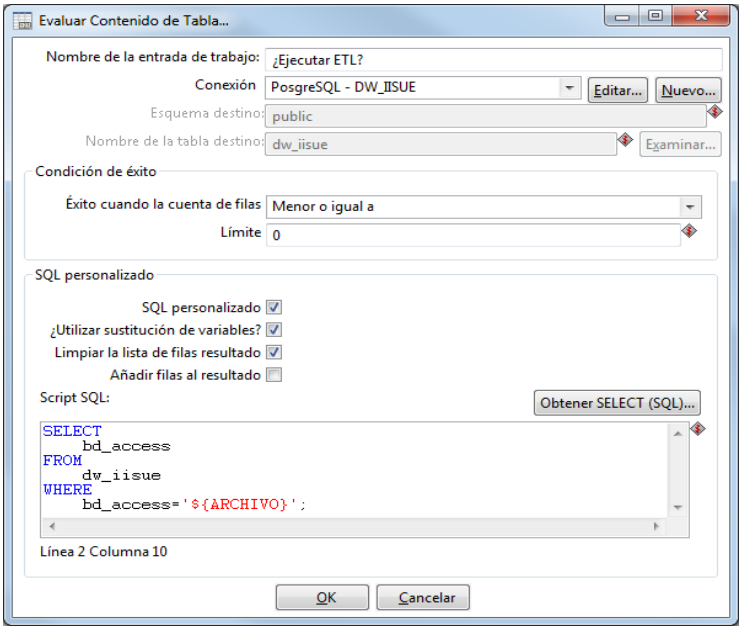


FIGURA 5.3.6.m. ¿Ejecutar ETL?

Alerta – ETL: mostrará una ventana emergente con el texto "El archivo "${ARCHIVO}" ya fue cargado en el DWH."

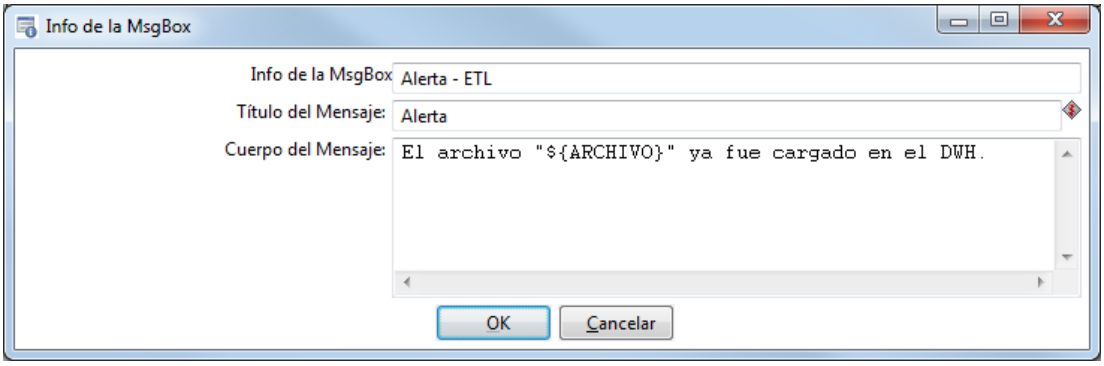


FIGURA 5.3.6.n. Alerta – ETL

SQL: en el paso "Ejecutar script SQL…" se personaliza una sentencia SQL para ejecutar la inserción de datos a la tabla "dw_iisue" con el valor que contiene el parámetro "${ARCHIVO}", como ya se explicó antes es el nombre de la base de datos de Access.

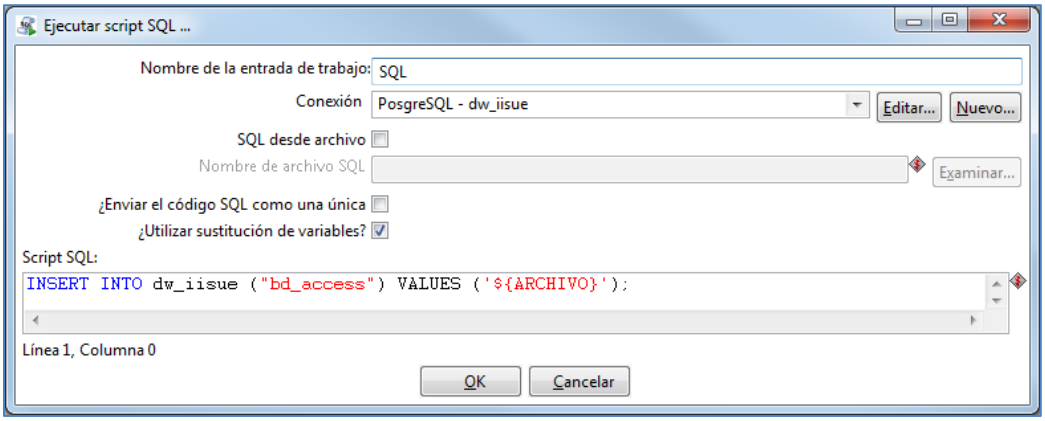


FIGURA 5.3.6.ñ. SQL

5.4. Esquemas ROLAP

Posteriormente debemos de generar el esquema ROLAP, basándonos en todo el análisis previo a los Data Marts y mediante el uso de la herramienta Pentaho Schema Worbench 3.5 es como se pueden generar los esquemas, únicamente será necesario seguir el esquema con el cual fueron diseñados (Esquema Estrella), a continuación describimos brevemente cada uno de los pasos que usamos para crear el Cubo de "grupos_academicos".

5.4.1. Conexión

Una vez que ingresamos a Pentaho Schema Workbench 3.5, lo primero que se define es la conexión a la base de datos en donde se encuentran nuestros esquemas. En nuestro caso definimos la conexión con PostgreSQL, como se muestra en la FIGURA 5.4.1.a.

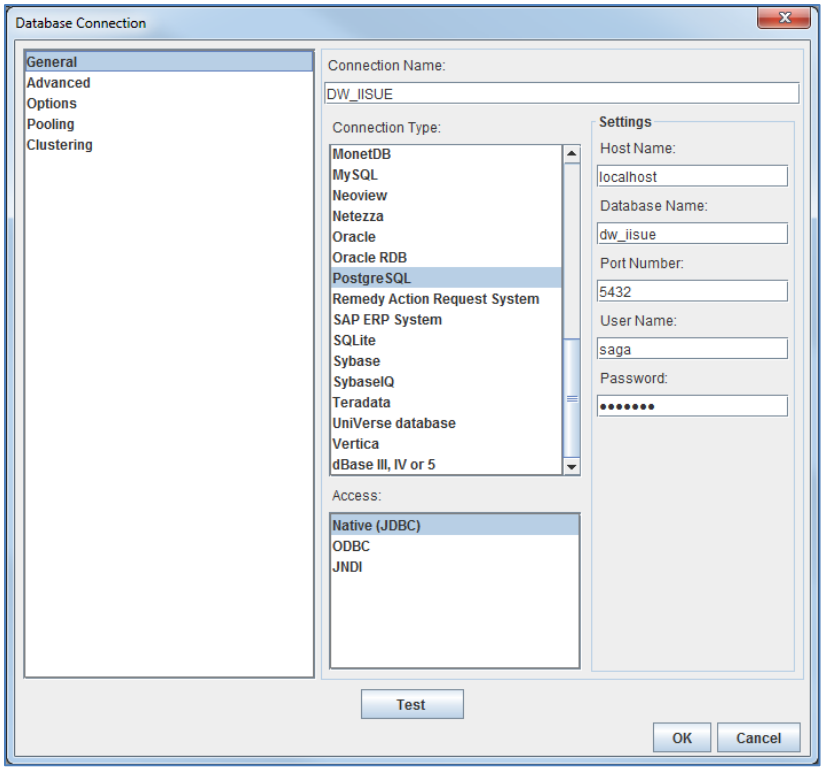


FIGURA 5.4.1.a. Conexión a la Base de Datos

5.4.2. Esquema

Abrimos un nuevo esquema y comenzamos a definir la estructura de la base de datos multidimensional, en este caso denominamos como DW_IISUE al esquema que contendrá los cubos del DWH y agregaremos una breve descripción (Data Warehouse del IISUE).

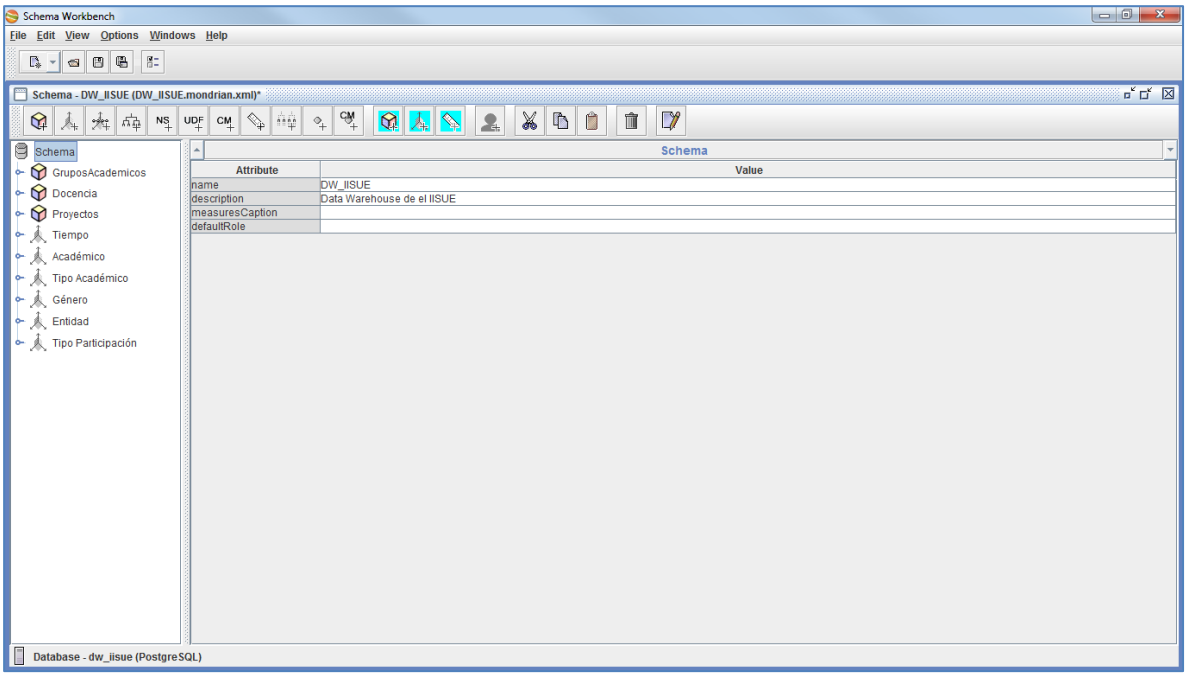


FIGURA 5.4.2.a. Esquema

5.4.3. Cubo - GruposAcademicos

En el cubo de grupos académicos es donde se define la estructura de la tabla de hechos, las medidas, los miembros calculados y las dimensiones. En la FIGURA 5.4.4.a. se muestra el esquema del cubo.

- ***Tabla de hechos:*** grupos_academicos.
- ***Dimensiones Compartidas:*** Tiempo, Tipo Académico, Género y Entidad.
- ***Dimensiones Privadas:*** Grado Estudio, Nacionalidad, SNI, PRIDE, PAIPA y FOMDOC.
- ***Medidas:*** Total de académicos.

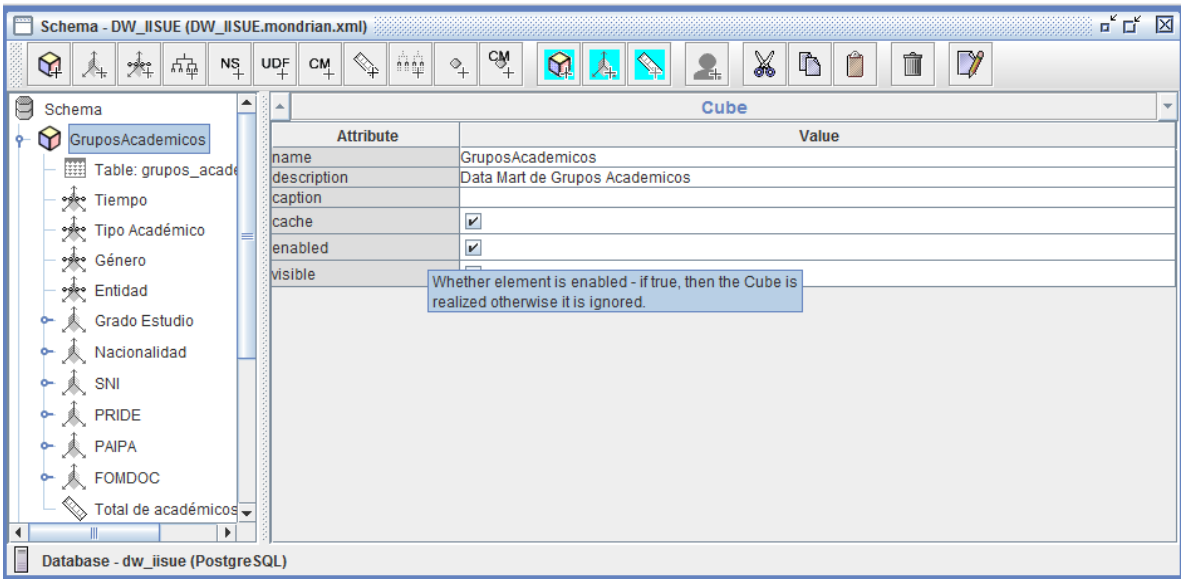


FIGURA 5.4.4.a. Cubo - GruposAcademicos

- ✓ ***name:*** es el nombre del cubo, que como se muestra en la imagen es el de "GruposAcademicos".
- ✓ ***description:*** la descripción del cubo (Data Mart de Grupos Académicos).
- ✓ ***cache:*** fue seleccionado para que ésta memoria intermedia almacene información del cubo, de forma que se pueda responder a consultas más rápidamente.
- ✓ ***enabled:*** elegimos esta opción para que Mondrian cargue el cubo.

5.4.4. Tabla de Hechos - grupos_academicos

Nuestra tabla principal del modelado multidimensional y que almacenara las medidas que nos interesan en este caso es la de "grupos_academicos".

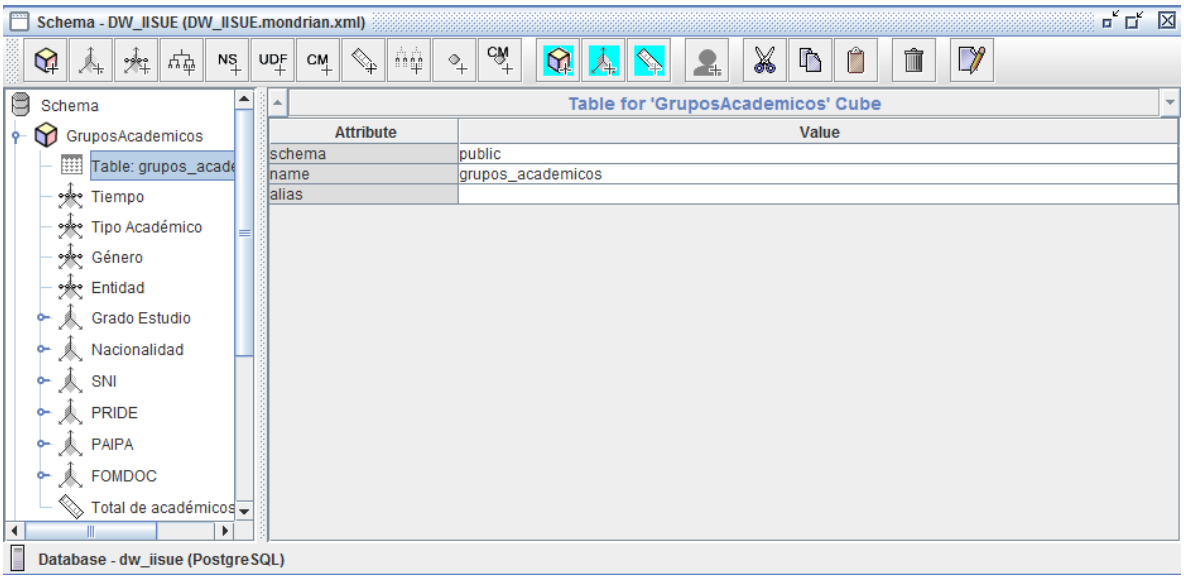


FIGURA 5.4.5.a. Tabla de Hechos - grupos_academicos

- ✓ ***schema:*** "public" es el nombre de nuestro esquema de la base de datos que contiene la tabla.
- ✓ ***name:*** el nombre de nuestra tabla de hechos "grupos_academicos".

5.4.5. Dimensiones compartidas

En nuestro modelado físico de GRUPOS_ACADEMICOS se comparten 3 dimensiones con los restantes Data Mart's y son las siguientes:

- La dimensión LUGAR.
- La dimensión TIPO DE ACADEMICO.
- La dimensión TIEMPO.

En la FIGURA 5.4.5.a. mostramos los atributos de la dimensión compartida TIEMPO. En donde tiene como nombre "Tiempo", tipo de dato StandardDimension, esto hará que sea tratada como una dimensión normal.

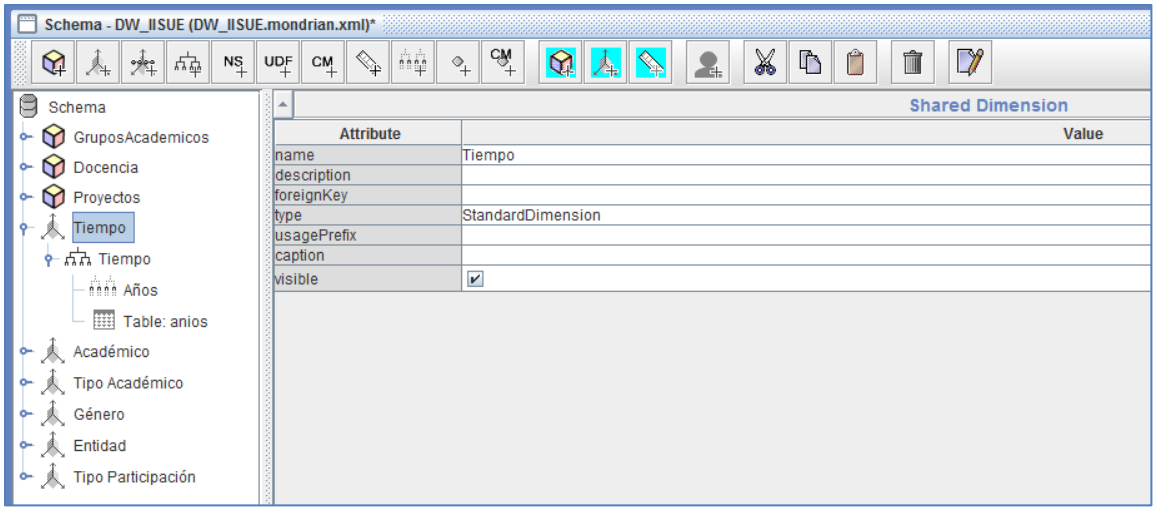


FIGURA 5.4.5.a. Dimensión compartida Tiempo

Jerarquía (Hierachy)

En la jerarquía Tiempo, se nos permite discriminar un conjunto de miembros sobre el mismo atributo en la misma tabla de hechos. En esta dimensión TIEMPO solo tiene una jerarquía y se puede observar en la FIGURA 5.4.5.b.

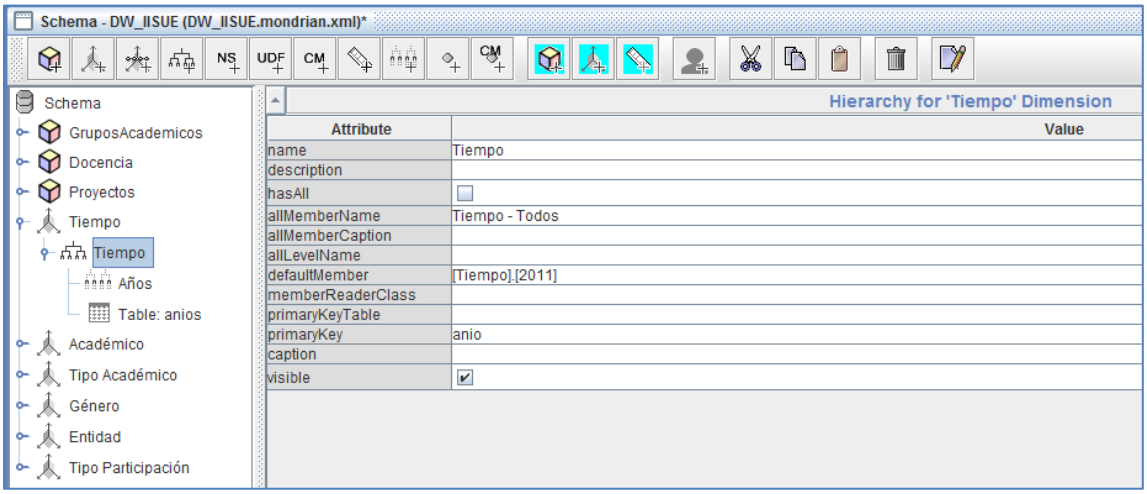


FIGURA 5.4.5.b. Jerarquía (Hierachy) Tiempo

Nivel (Level)

El nivel “Años” es el nivel de granularidad único de análisis y detalle de la información de nuestro modelo dimensional, que luego nos permitirá realizar el análisis y la navegación por los datos.

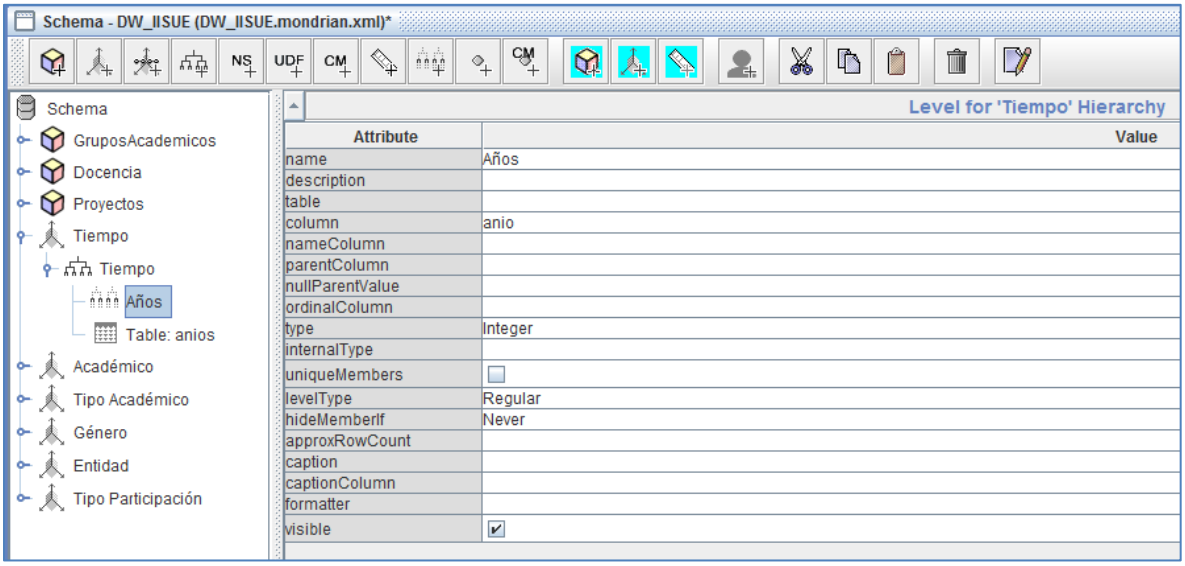


FIGURA 5.4.5.c. Nivel (Level) Años

Tabla de Dimensiones (Table)

Es la tabla dimensión que contiene los datos de la dimensión, en este caso, como es la dimensión TIEMPO la tabla correspondiente es "anios" y se encuentra en el esquema "public".

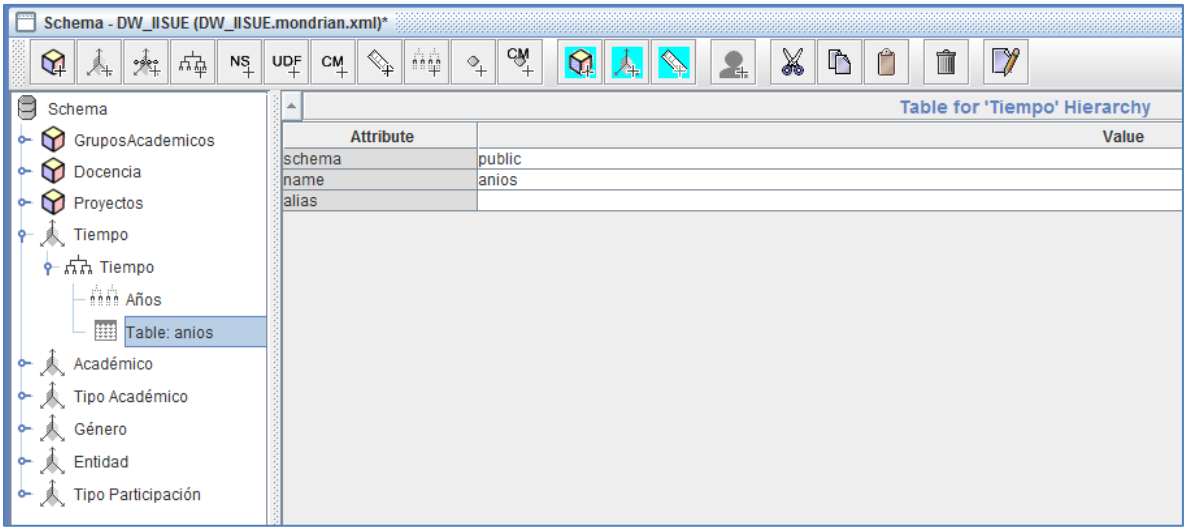


FIGURA 5.4.5.d. Tabla de Dimensiones (Table)

5.4.6. Agregar Dimensiones compartidas al cubo

En el punto anterior se mostró como crear una dimensión compartida y se dijo que nos servirá para asociarlas a múltiples cubos, así es que en la siguiente imagen se muestra como la dimensión TIEMPO se asocia al cubo de “grupos_academicos”.

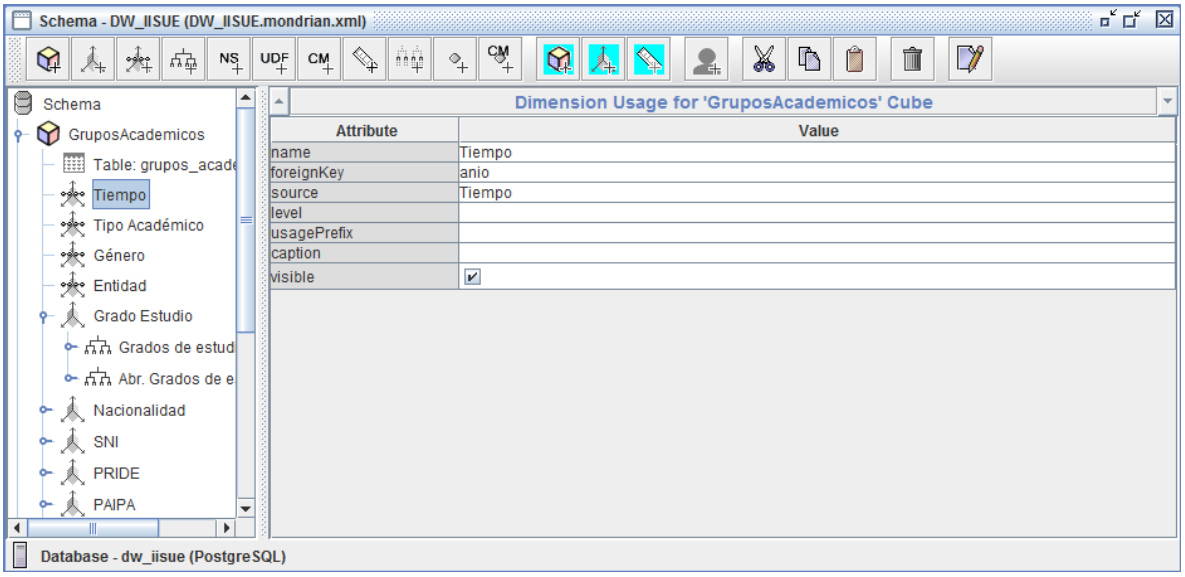


FIGURA 5.4.6.a. Agregar Dimensiones compartidas al cubo

- ✓ ***name:*** este nombre tiene que ser igual al de la dimensión compartida (TIEMPO).
- ✓ ***foreignKey:*** anio es el nombre de la columna en la tabla de hechos que hace referencia a la primary key de nuestra tabla de “anios”.
- ✓ ***source:*** es el nombre de nuestra dimensión compartida, que en este caso es la de “Tiempo”.
- ✓ ***visible:*** lo seleccionaremos para que se muestre en la interface del usuario.

5.4.7. Dimensiones privadas del cubo

Las dimensiones GRADO ESTUDIO, DEPARTAMENTO, NACIONALIDAD y ESTIMULO, son dimensiones privadas, porque son únicamente utilizadas en un solo cubo, obviamente el cubo que estamos trabajando es “grupos_academicos”.

Agregamos la dimensión GRADO ESTUDIO y configuramos los atributos como se muestra en la FIGURA 5.4.7.a.

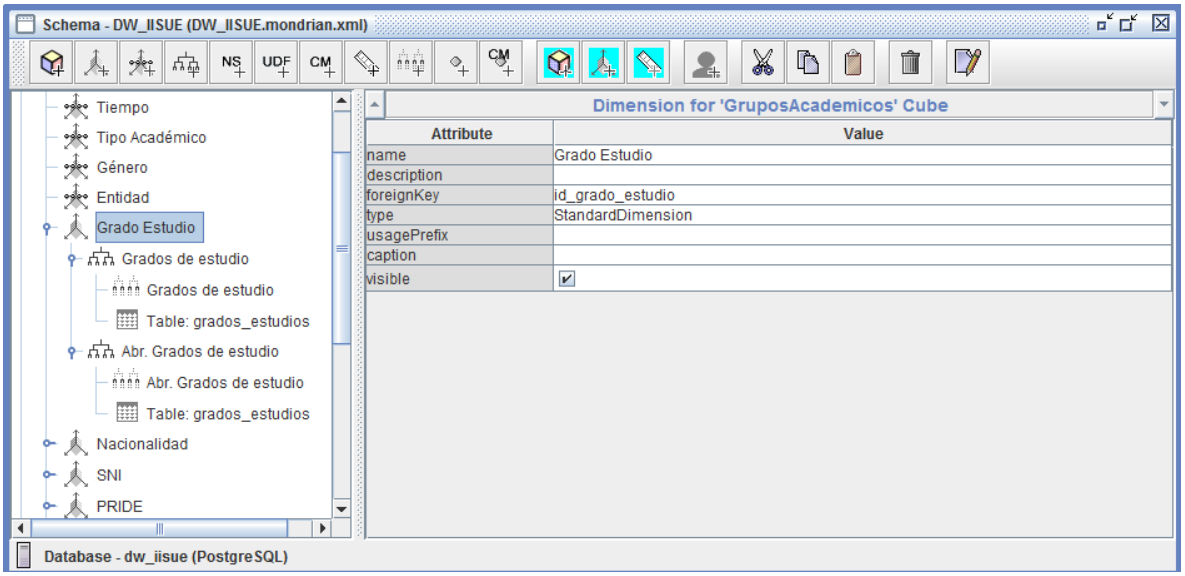


FIGURA 5.4.7.a. Dimensiones privadas del cubo

- ✓ ***name:*** “Grado Estudio” es el nombre de la dimensión.
- ✓ ***foreignKey:*** es la columna (id_grado_estudio) de la tabla de hechos que hace referencia a la clave primaria de la tabla dimensional (grados_estudios).
- ✓ ***type:*** definimos que nuestra dimensión sea de tipo Standard y no una dimensión de tiempo.

Jerarquía (Hierachy)

Como ya se mencionó anteriormente la jerarquía nos permite discriminar un conjunto de miembros en la tabla de hechos. En la dimensión grado de estudio tenemos dos jerarquías, la de “Grados de estudio” y la de “Abr. Grados de estudio”, éstas apuntan a la misma tabla.

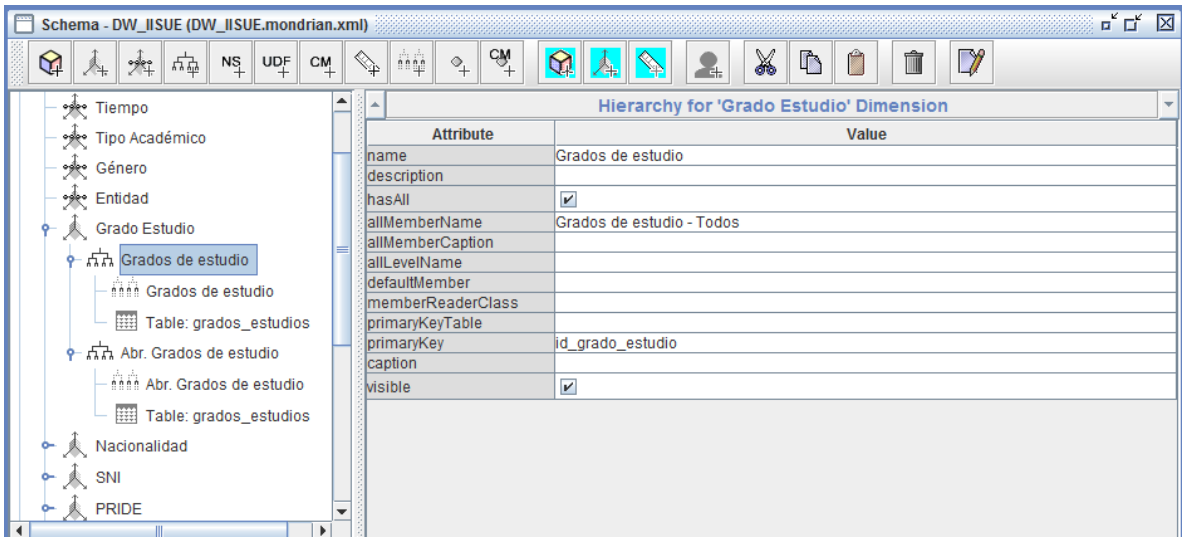


FIGURA 5.4.7.b. Jerarquía (Hierachy)

- ✓ ***name:*** nombre de la jerarquía, que en este caso es la de “Grados de estudio”.
- ✓ ***hasAll:*** nos indica si la jerarquía tiene un nivel “All” con un miembro “All”, generalmente es un indicativo de que se usaran todos los valores de la jerarquía.
- ✓ **allMemberCaption:** nombre (Abr. Grados de estudio - Todos) que se mostrará para los miembros All.

Nivel (Level)

El nivel de la jerarquía anterior es “Grados de estudio” y lo configuramos como se muestra en la siguiente imagen.

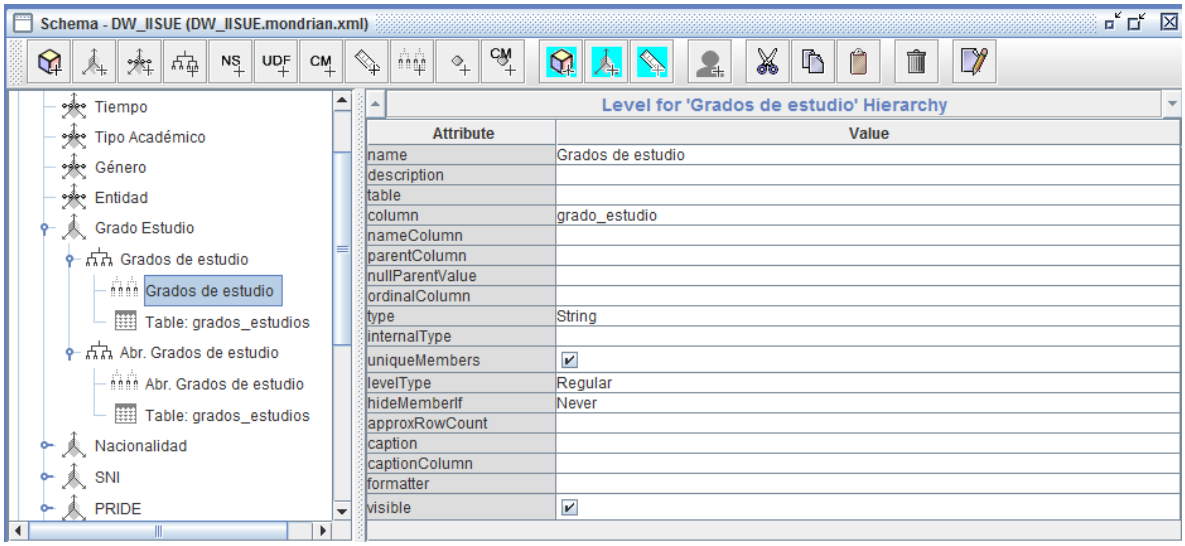


FIGURA 5.4.7.c. Nivel (Level)

- ✓ **name:** el nombre del nivel del nivel es el de Grupos de estudios.
- ✓ **column:** “grado_estudio” es la columna que identifica los miembros del nivel en la jerarquía.
- ✓ **type:** el tipo de dato del nivel es de String, ya que contendrá texto.
- ✓ **uniqueMembers:** nos indica que todos los miembros en el nivel tiene valores únicos.
- ✓ **levelType:** es un nivel “regular”.
- ✓ **hideMemberIf:** aquí determinamos que el miembro no este oculto.

Tabla de Dimensiones (Table)

Anteriormente se explicó que en este apartado seleccionaremos el nombre de la tabla de dimensión y el esquema, como se muestra en la siguiente imagen.

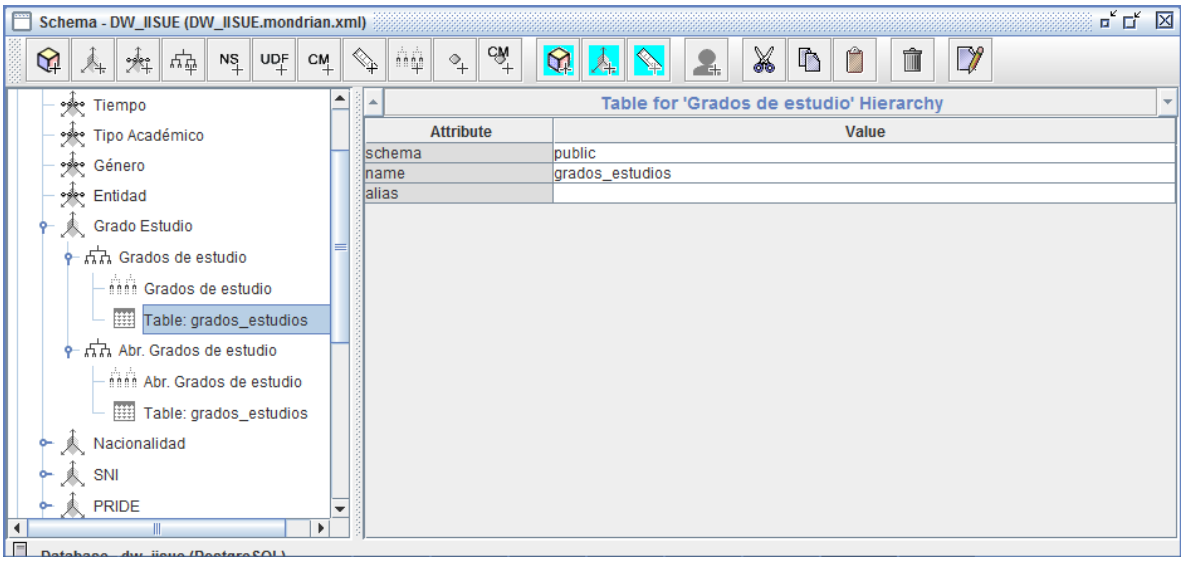


FIGURA 5.4.7.d. Tabla de Dimensiones (Table)

Medida (measure)

El "Total de académicos" es la medida que nos interesa calcular en nuestra tabla de hechos.

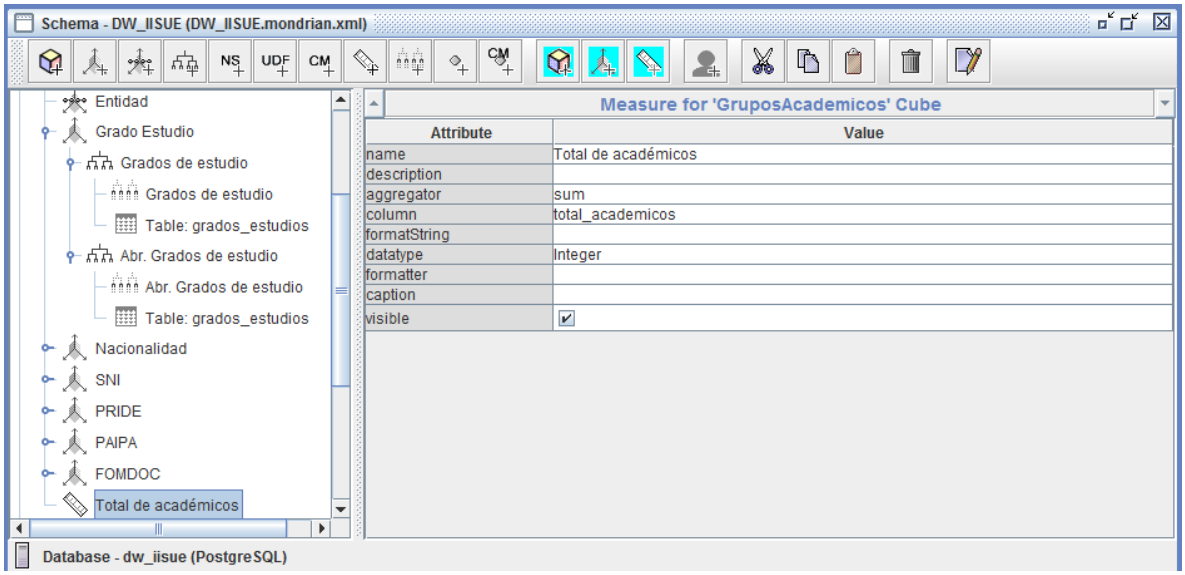


FIGURA 5.4.7.e. Medida (measure)

- ✓ **name:** nuestro nombre de la medida es "Total de académicos".
- ✓ **aggregator:** la función que llevará a cabo la agregación de las filas es de "sum". Puede ser uno de los siguientes valores sum, count, max, min, avg, distinct-count.
- ✓ **column:** "total_academicos" es la columna de la tabla de hechos correspondiente a la medida.
- ✓ **datatype:** es el tipo de dato de la columna, en este caso es de tipo "integer".

5.4.8. Publicar el cubo

Una vez terminado el modelado del esquema completo, el siguiente y último paso para poder utilizarlos en los análisis de BI de Pentaho es el publicarlo.

Salvamos el cubo y seleccionamos la opción de menú File –> Publish.

Se abrirá una nueva ventana, como la que se muestra en la FIGURA 5.4.8.a. y se pone la dirección de publicación de nuestro servidor, la contraseña de publicación, el usuario y el password. Se realiza la conexión con el servidor y el esquema ya está disponible para ser utilizado.

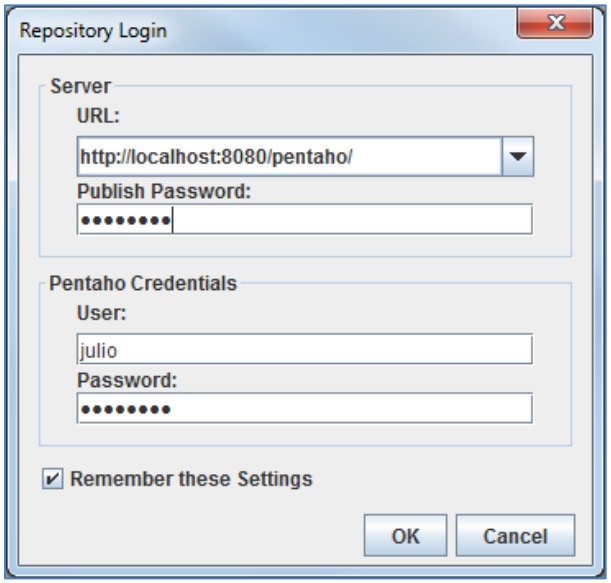


FIGURA 5.4.8.a. Repository Login

En la ventana Publish Schema (FIGURA 5.4.8.b) guardamos el esquema que creamos en la carpeta dw_iisue, que se encuentra en el repositorio y damos clic en el botón “Publish”.

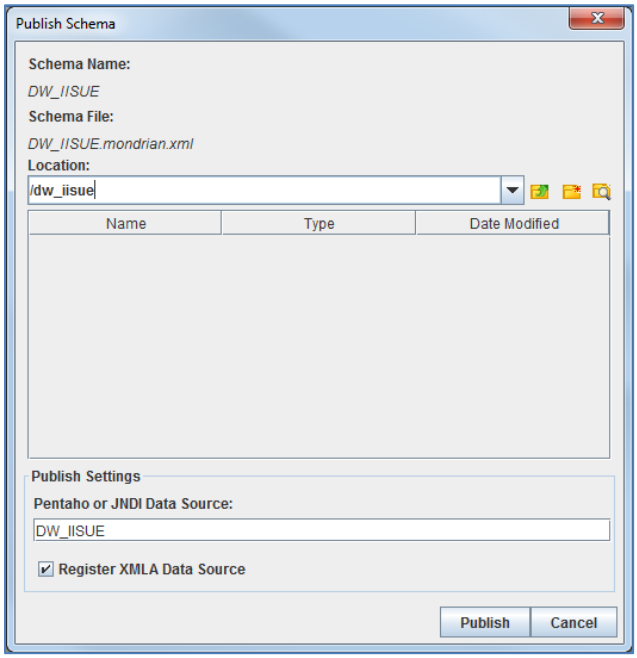


FIGURA 5.4.8.b. Publish Schema

Después de haber publicado el esquema de nuestros cubos satisfactoriamente en el repositorio de soluciones de pentaho, ya se puede utilizar el servidor de Pentaho BI para construir vistas de análisis.

5.5. Exploración de Datos y Análisis

Finalmente después de realizar todo el proceso de Data Warehousing hemos logrado llegar a la exploración y análisis de los datos, es decir, ahora podemos ejecutar consultas ad-hoc en tiempo real y totalmente personalizadas, y estadísticas de los Cubos que han sido creados

5.5.1. Análisis Interactivos

Para poder ver los cubos que creamos tendremos que entrar a la interfaz de Pentaho BI Server, por medio del navegador web ingresamos a la siguiente dirección http://localhost:8080/pentaho/ (esta dirección debe ser la URL del servidor) e iniciamos sesión en el sistema, como se muestra en la FIGURA 5.5.1.a.

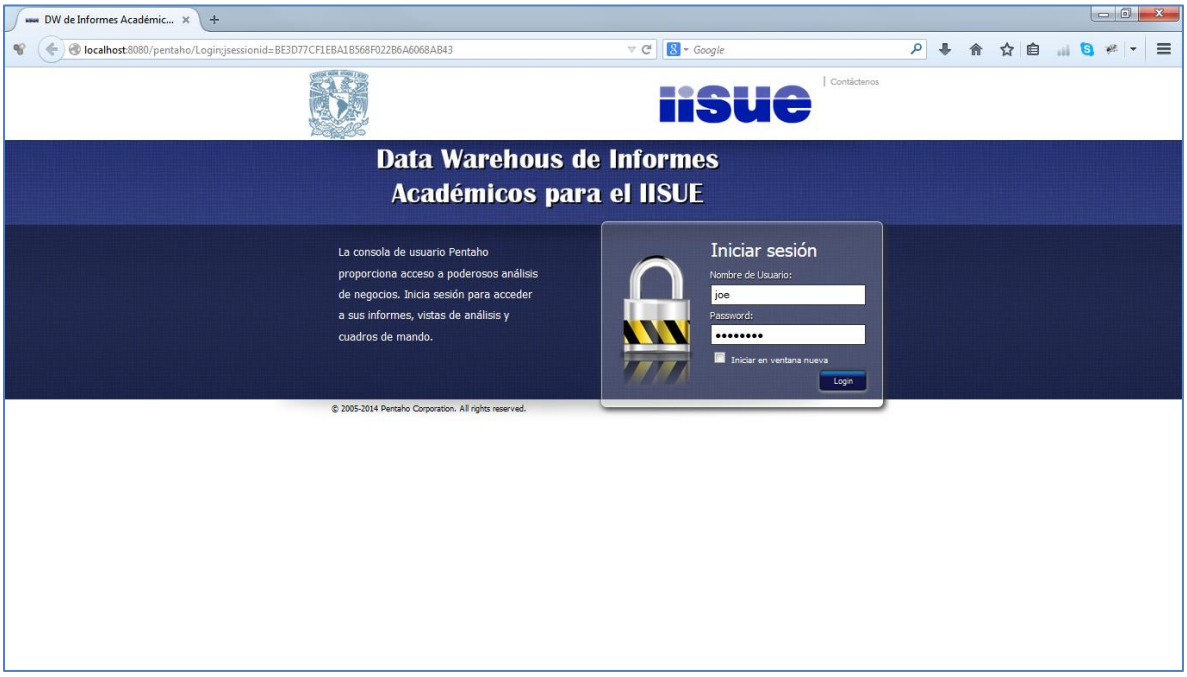


FIGURA 5.5.1.a. Interfaz de Pentaho BI Server

Nos aparece una pestaña que dice “New Saiku Analytics” y damos clic.

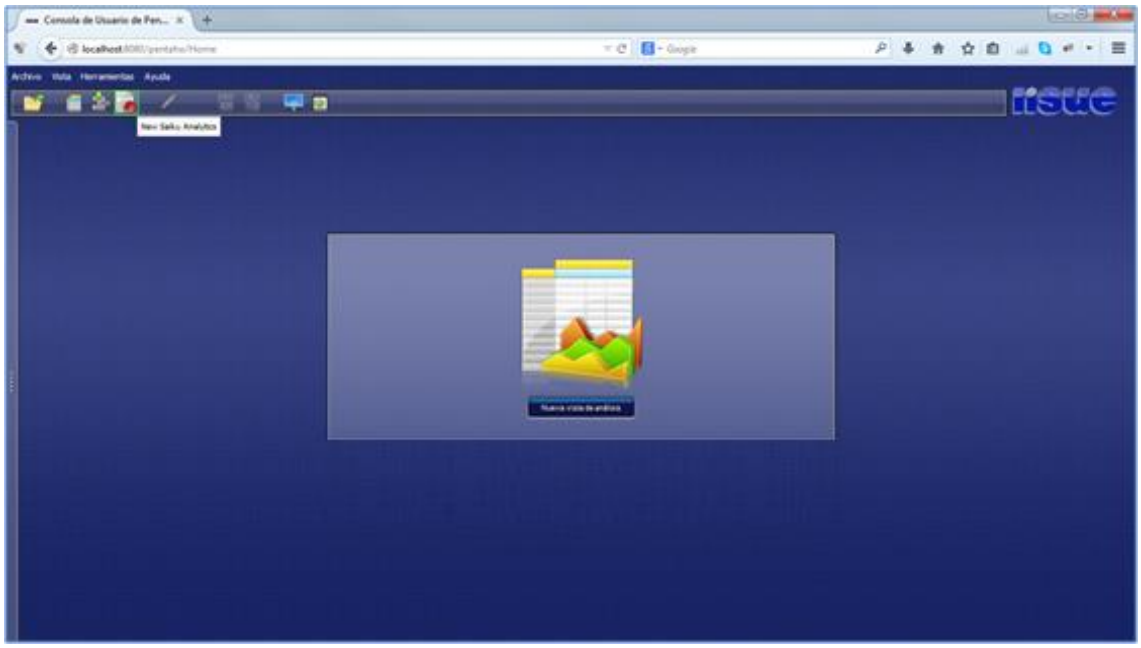


FIGURA 5.5.1.b. New Saiku Analytics

En la nueva ventana o área de trabajo nos aparece en la parte izquierda un panel para estar controlando las vistas de análisis pertenecientes a los Cubos que hemos creado.

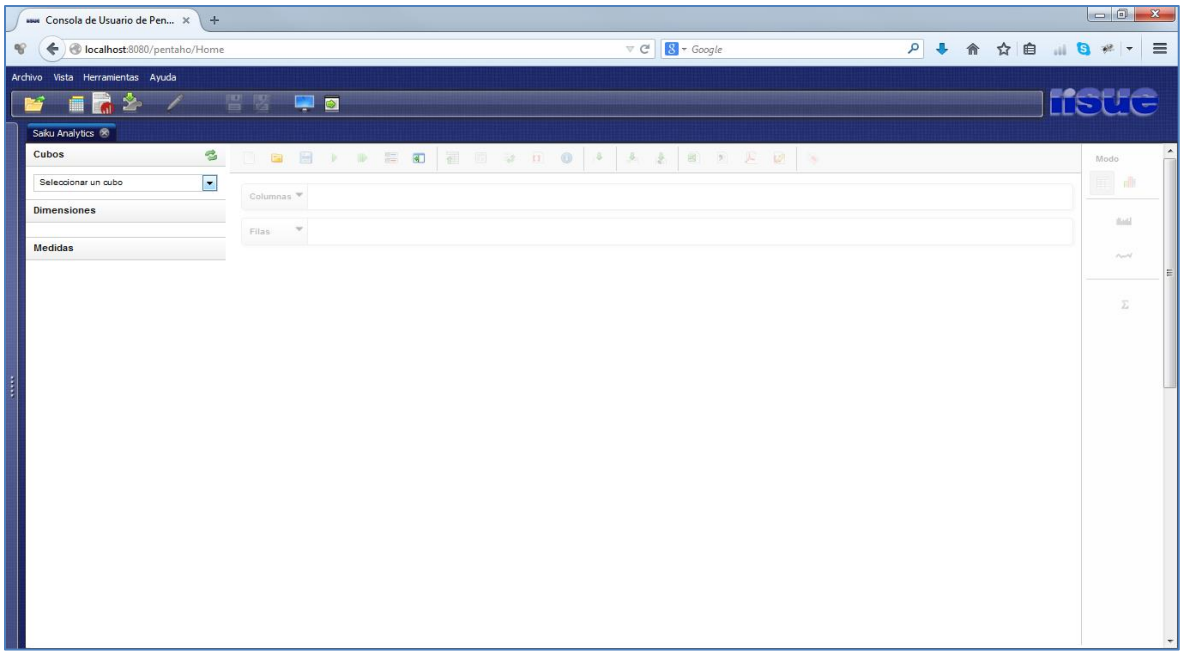


FIGURA 5.5.1.c. Saiku Analytics

Para determinar si el análisis, diseño y el desarrollo del DWH logró cumplir los objetivos y requerimientos será necesario realizar pruebas que justifiquen y comprueben el logro de tales requerimientos.

Requerimiento 1

"El IISUE tiene la necesidad de saber el total de académicos en base al tipo de académico (Investigador, Profesor o Técnico académico) anualmente. "

Como se muestra en la FIGURA 5.5.1.d, seleccionamos años y clase en columnas, el Total de académicos, que son nuestras medidas y estarán en las filas.

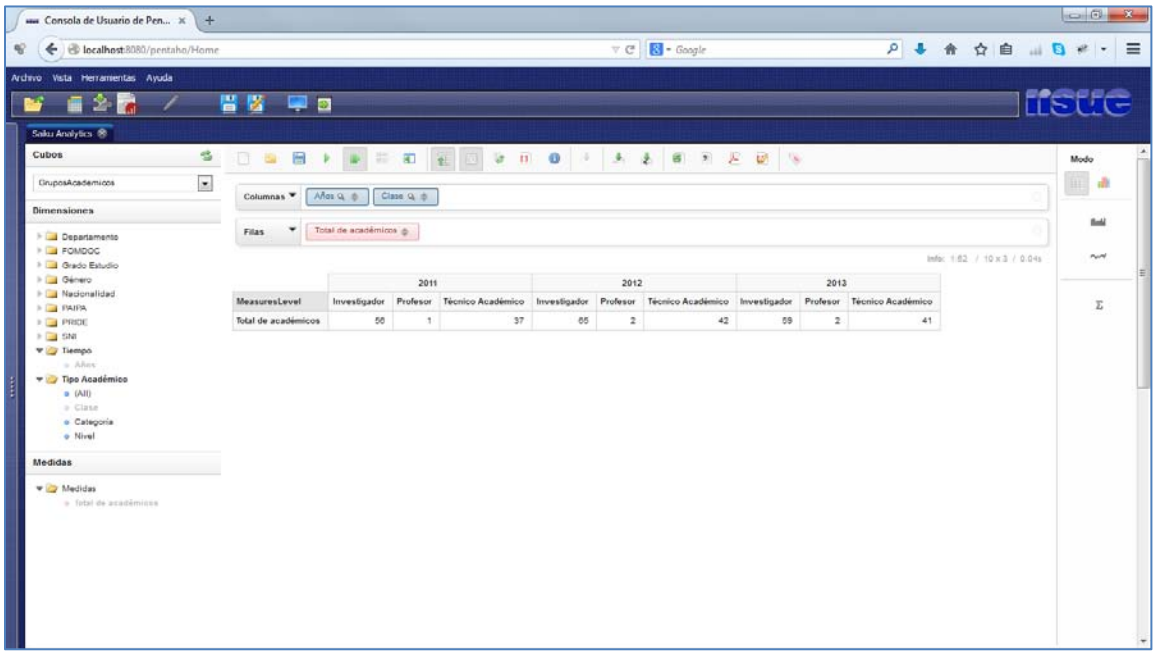


FIGURA 5.5.1.d. Requerimiento 1 (a)

Con la visualización de datos de saiku se nos permite analizar los resultados por medio de gráficas, esto nos facilita la interpretación de la información.

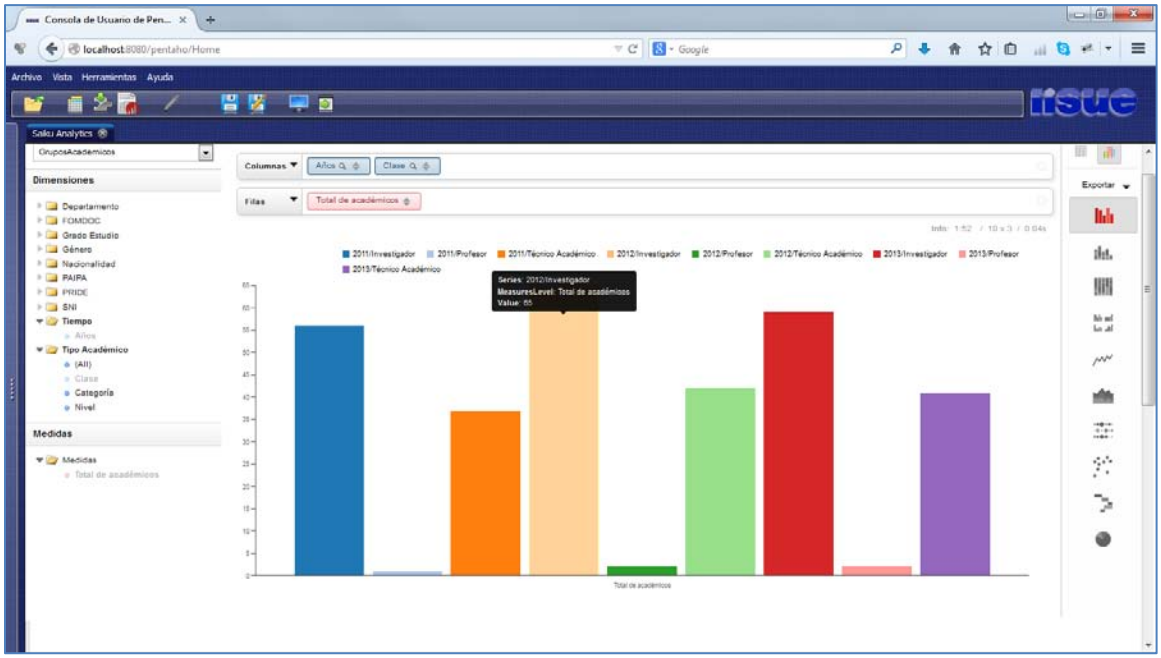


FIGURA 5.5.1.e. Requerimiento 1 (b)

Por medio de la visualización de datos y con la gráfica (FIGURA 5.5.1.e) podemos observar que en el 2012 hay mayor número de Académicos Investigadores.

Requerimiento 2

"El IISUE necesita saber el total de académicos en base al nivel de estudios que poseen sus académicos anualmente."

En columnas arrastramos años y Grados de estudio, en filas Total de académicos.

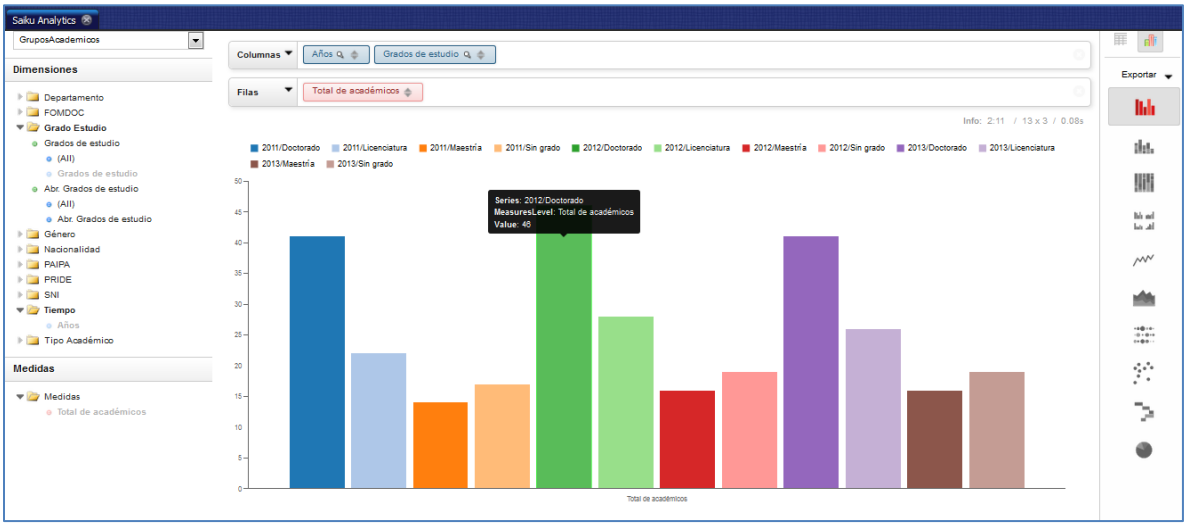
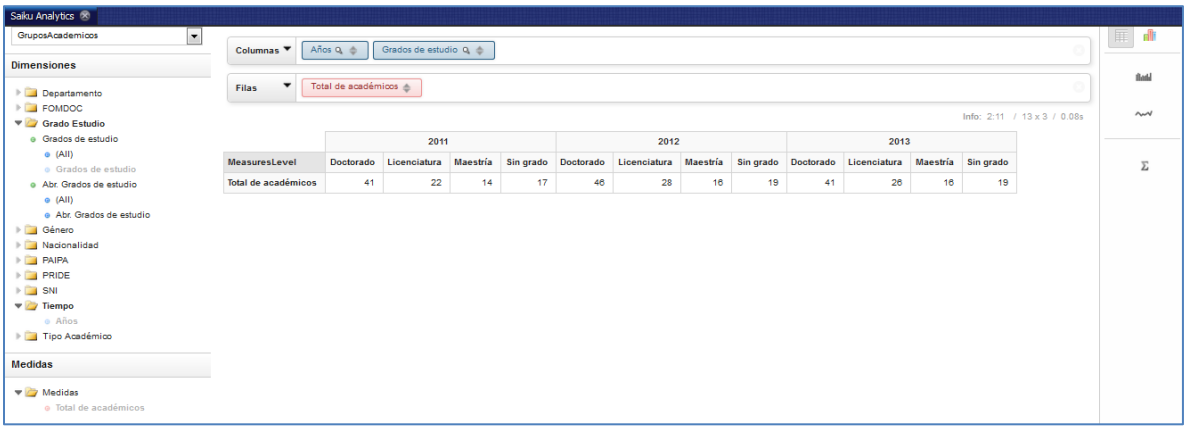


FIGURA 5.5.1.f. Requerimiento 2 (b)



	2011				2012				2013			
MeasuresLevel	Doctorado	Licenciatura	Maestría	Sin grado	Doctorado	Licenciatura	Maestría	Sin grado	Doctorado	Licenciatura	Maestría	Sin grado
Total de académicos	41	22	14	17	46	28	16	19	41	26	16	19

FIGURA 5.5.1.g. Requerimiento 2 (a)

Por medio de las dos imágenes anteriores, podemos analizar que en el 2012 tiene el mayor número de académicos que además cuentan con un Doctorado.

Requerimiento 3

"El IISUE solicitó saber el total de académicos en base al departamento en el que están adscritos y al tipo de nacionalidad que tienen."

En columnas se eligió el año con filtro de 2011, nacionalidades y el departamento, en filas se agregó el total de académicos.

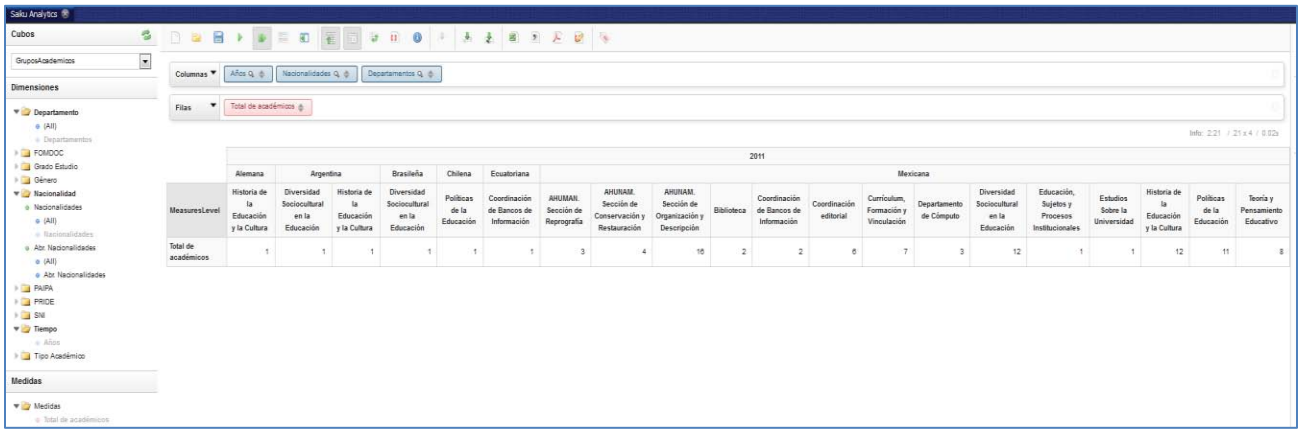


FIGURA 5.5.1.h. Requerimiento 3 (a)

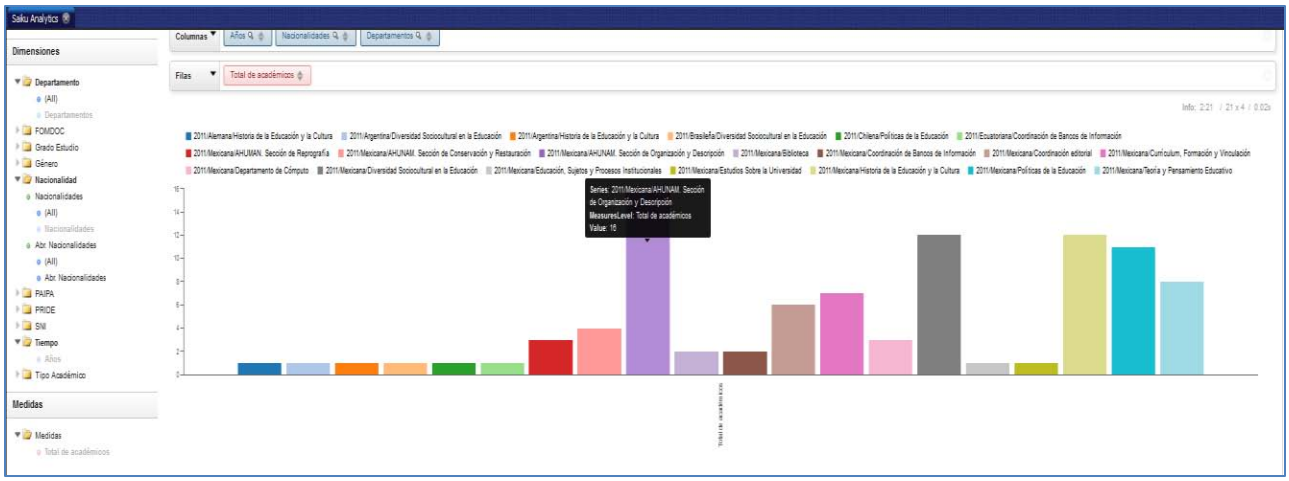


FIGURA 5.5.1.i. Requerimiento 3 (b)

Podemos observar que el departamento "AHUNAM. Sección de Organización y Descripción" es el que tiene más académicos con nacionalidad Mexicana en el 2011.

Requerimiento 4

"El IISUE tiene la necesidad de saber los tipos de estímulos (SNI, PRIDE, FOMDOC, PAIPA) que se les brindan a sus académicos."

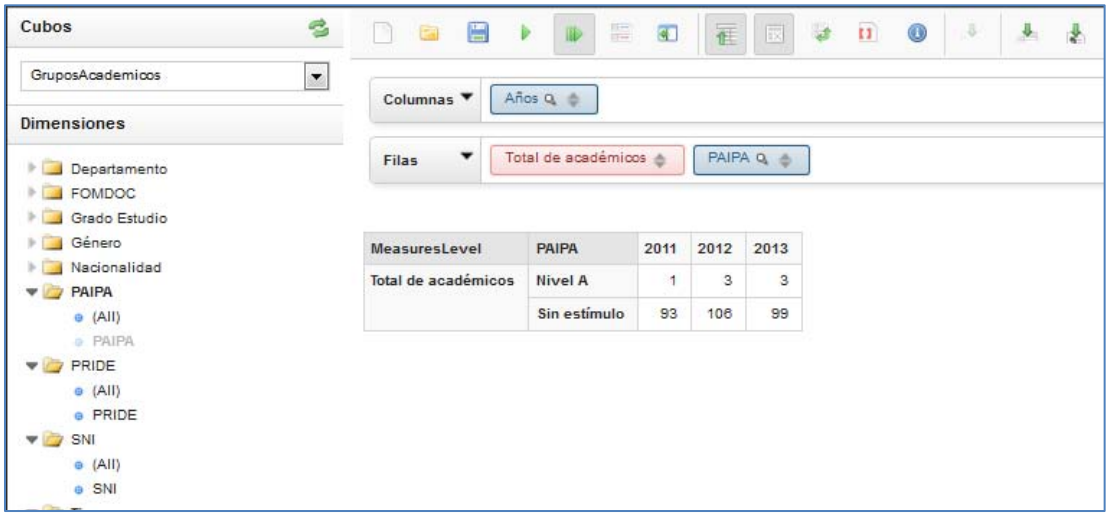


FIGURA 5.5.1.j. Requerimiento 4 (a)

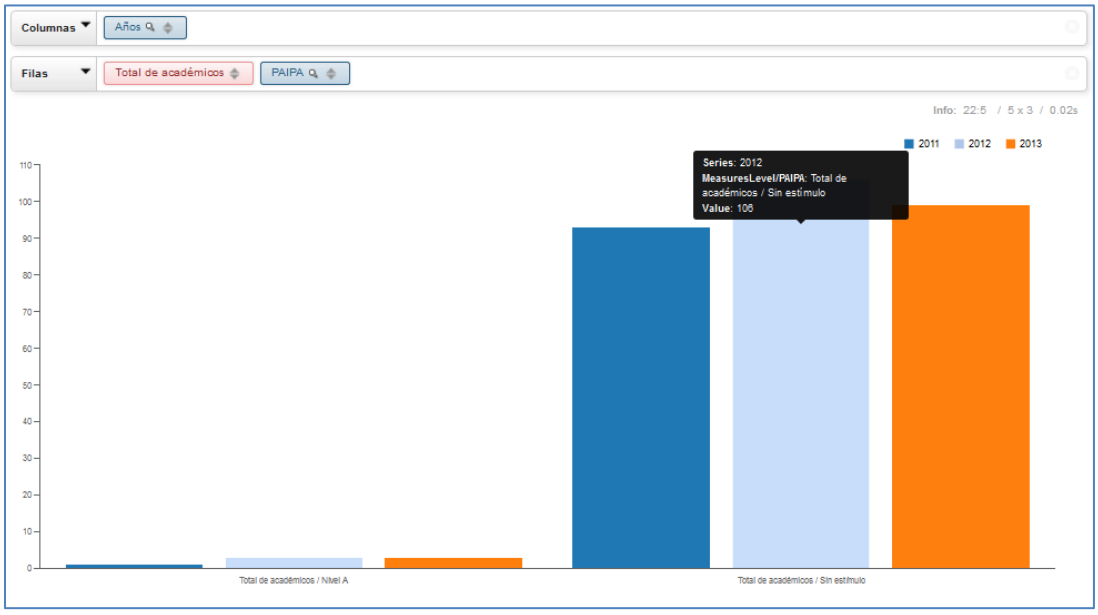
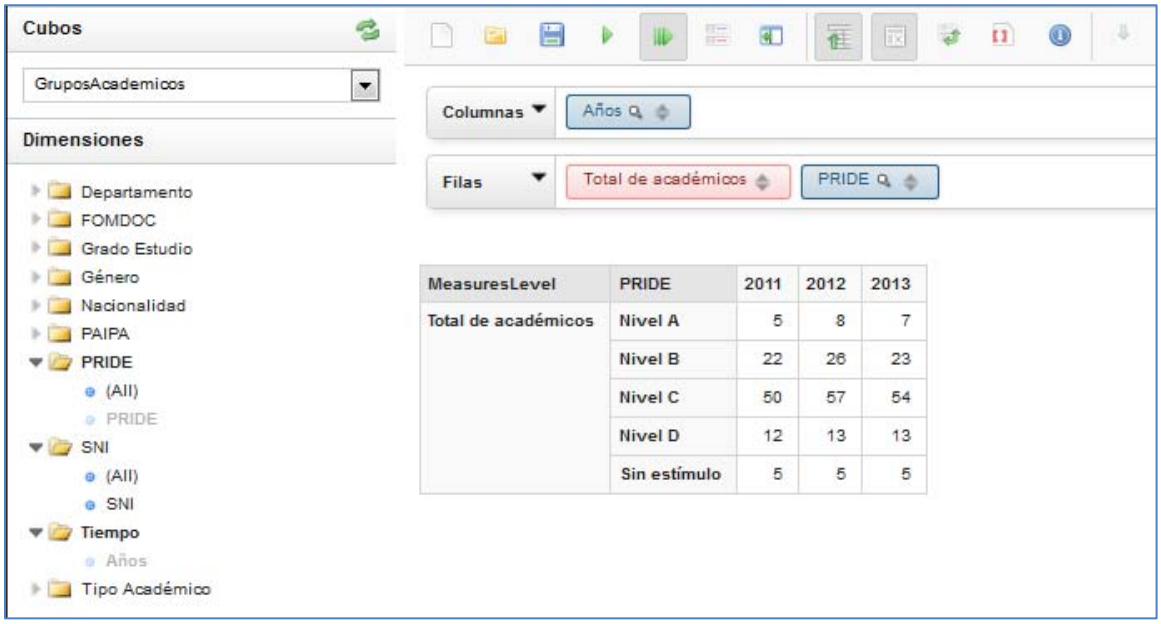


FIGURA 5.5.1.k. Requerimiento 4 (b)



MeasuresLevel	PRIDE	2011	2012	2013
Total de académicos	Nivel A	5	8	7
	Nivel B	22	26	23
	Nivel C	50	57	54
	Nivel D	12	13	13
	Sin estímulo	5	5	5

FIGURA 5.5.1.l. Requerimiento 4 (c)

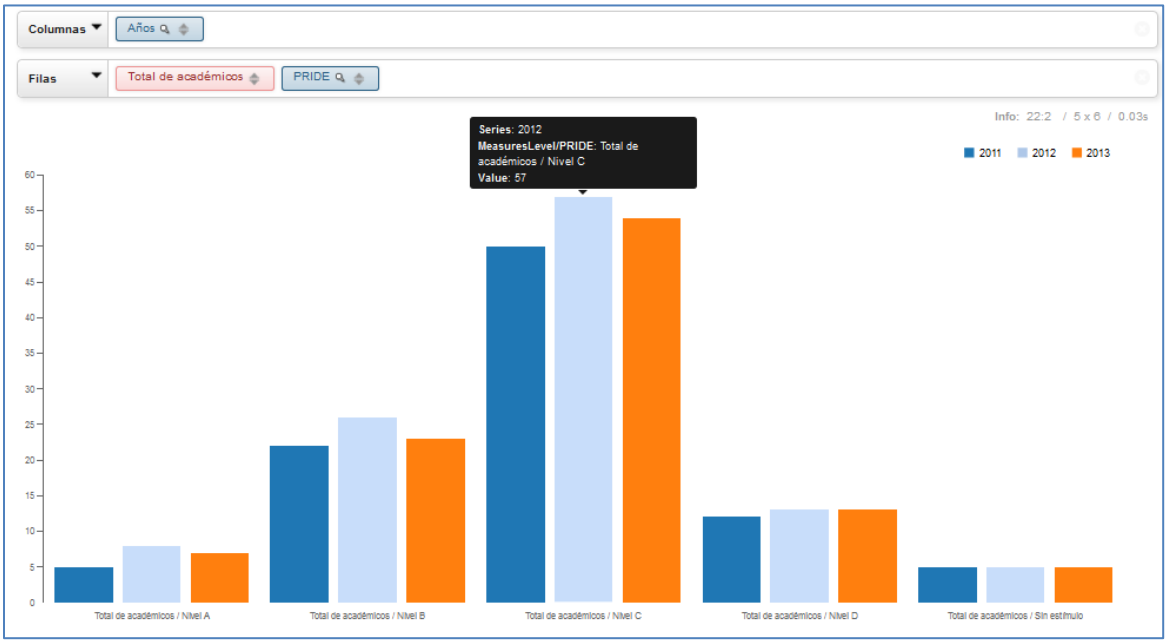


FIGURA 5.5.1.m. Requerimiento 4 (d)

MeasuresLevel	SNI	2011	2012	2013
Total de académicos	Candidato		2	2
	I	11	13	12
	II	16	19	16
	III	5	6	6
	Sin estímulo	62	69	66

FIGURA 5.5.1.n. Requerimiento 4 (e)

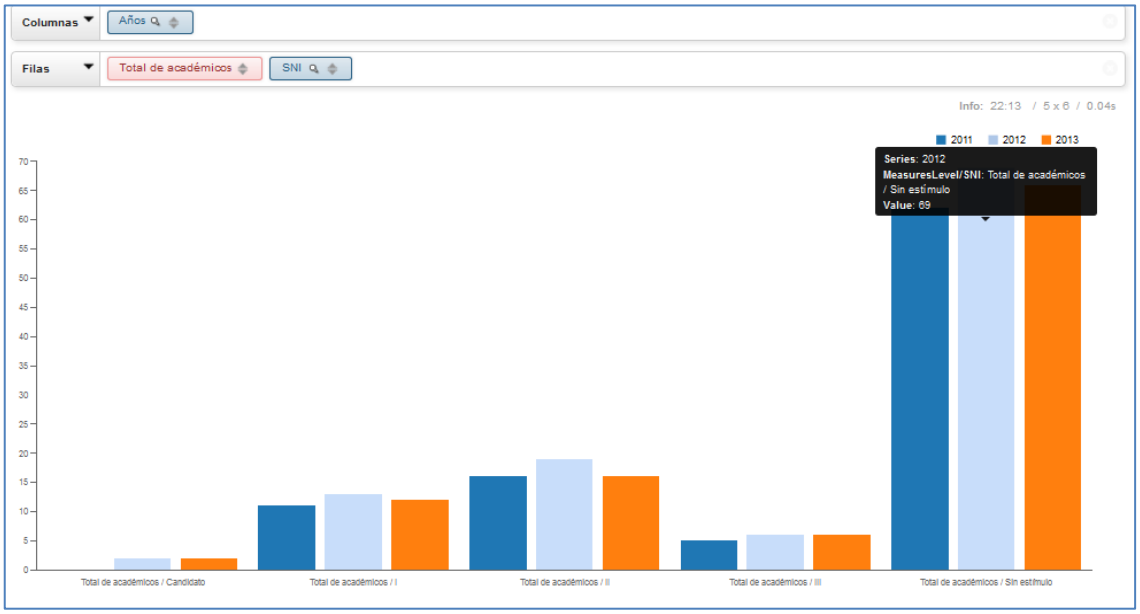
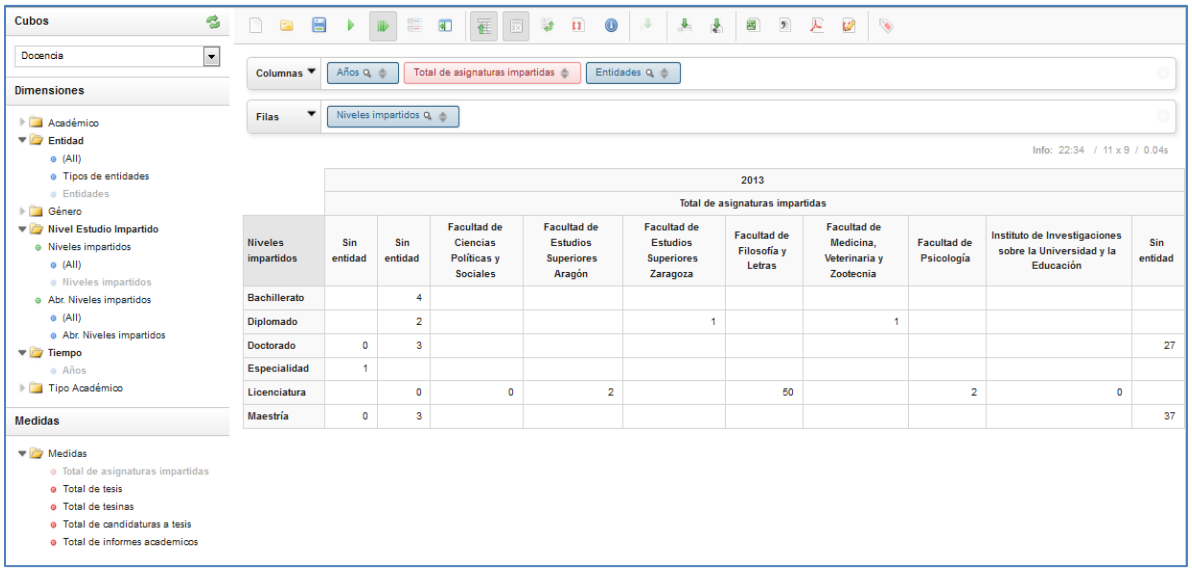


FIGURA 5.5.1.ñ. Requerimiento 4 (f)

En las imágenes que se mostraron anteriormente, se puede observar que cumple el requerimiento para el IISUE de mostrar los diferentes estímulos que se les brindan a sus académicos, por año.

Requerimiento 5

"El IISUE pretende conocer el total de asignaturas impartidas por sus académicos, en base al grado de estudio impartido y a la entidad adscripta."



	2013									
	Total de asignaturas impartidas									
Niveles impartidos	Sin entidad	Sin entidad	Facultad de Ciencias Políticas y Sociales	Facultad de Estudios Superiores Aragón	Facultad de Estudios Superiores Zaragoza	Facultad de Filosofía y Letras	Facultad de Medicina, Veterinaria y Zootecnia	Facultad de Psicología	Instituto de Investigaciones sobre la Universidad y la Educación	Sin entidad
Bachillerato		4								
Diplomado		2			1		1			
Doctorado	0	3								27
Especialidad	1									
Licenciatura		0	0	2		50		2	0	
Maestría	0	3								37

FIGURA 5.5.1.o. Requerimiento 5 (a)

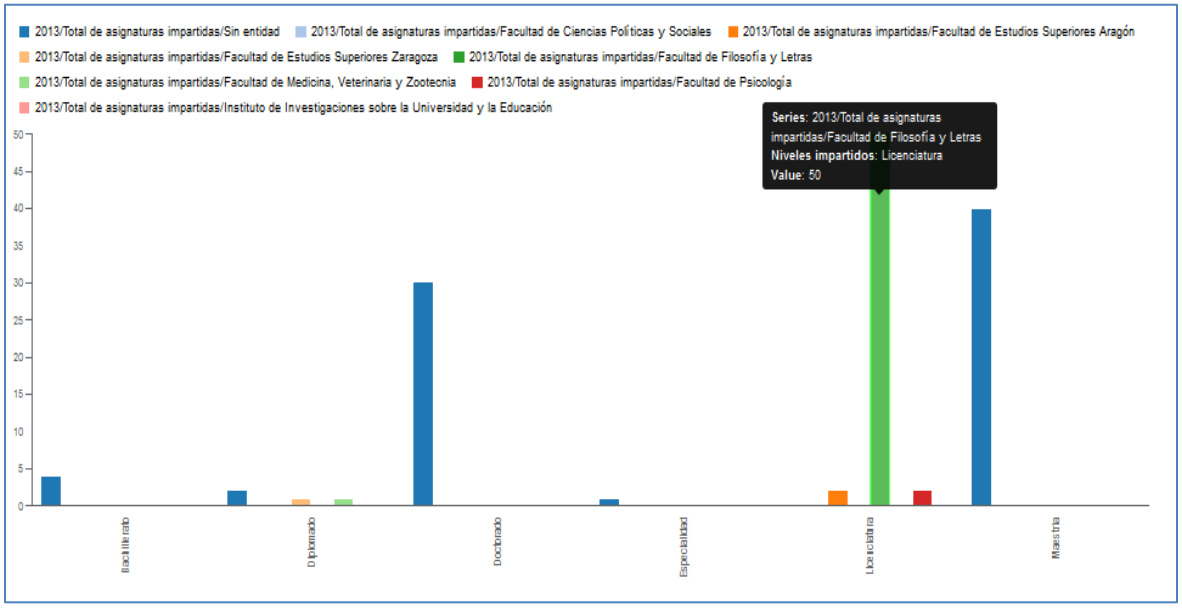
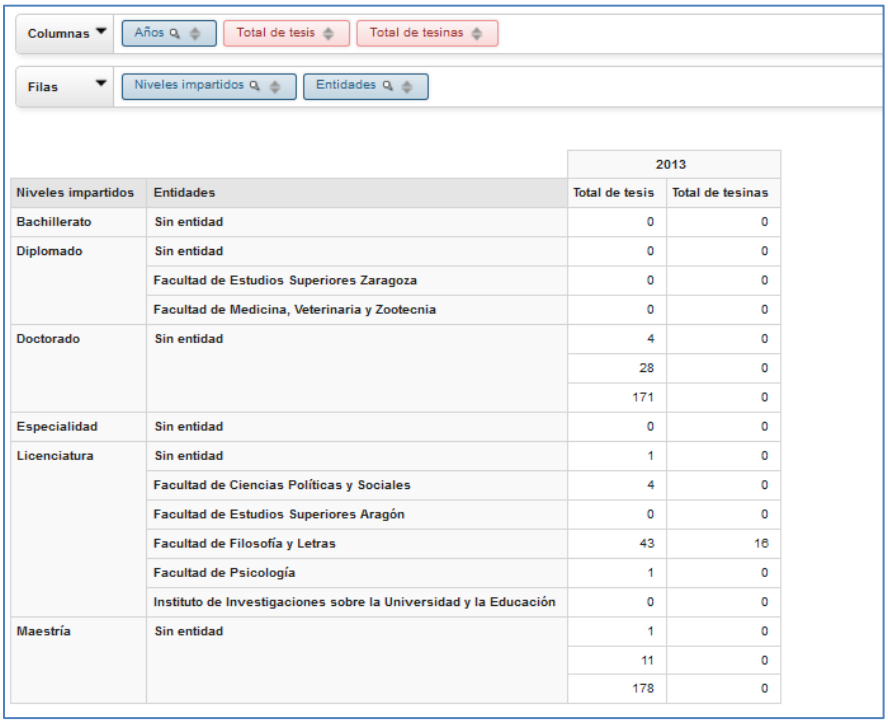


FIGURA 5.5.1.p. Requerimiento 5 (b)

Observamos que los académicos con licenciatura tienen el mayor número de asignaturas impartidas en la Facultad de Filosofía y Letras en el 2013.

Requerimiento 6

"El IISUE pidió conocer el total de tesis y tesinas dirigidas por sus académicos, en base al nivel de estudio impartido y a la entidad adscripta."



		2013	
Niveles impartidos	Entidades	Total de tesis	Total de tesinas
Bachillerato	Sin entidad	0	0
Diplomado	Sin entidad	0	0
	Facultad de Estudios Superiores Zaragoza	0	0
	Facultad de Medicina, Veterinaria y Zootecnia	0	0
Doctorado	Sin entidad	4	0
		28	0
		171	0
Especialidad	Sin entidad	0	0
Licenciatura	Sin entidad	1	0
	Facultad de Ciencias Políticas y Sociales	4	0
	Facultad de Estudios Superiores Aragón	0	0
	Facultad de Filosofía y Letras	43	16
	Facultad de Psicología	1	0
	Instituto de Investigaciones sobre la Universidad y la Educación	0	0
Maestría	Sin entidad	1	0
		11	0
		178	0

FIGURA 5.5.1.q. Requerimiento 6 (b)

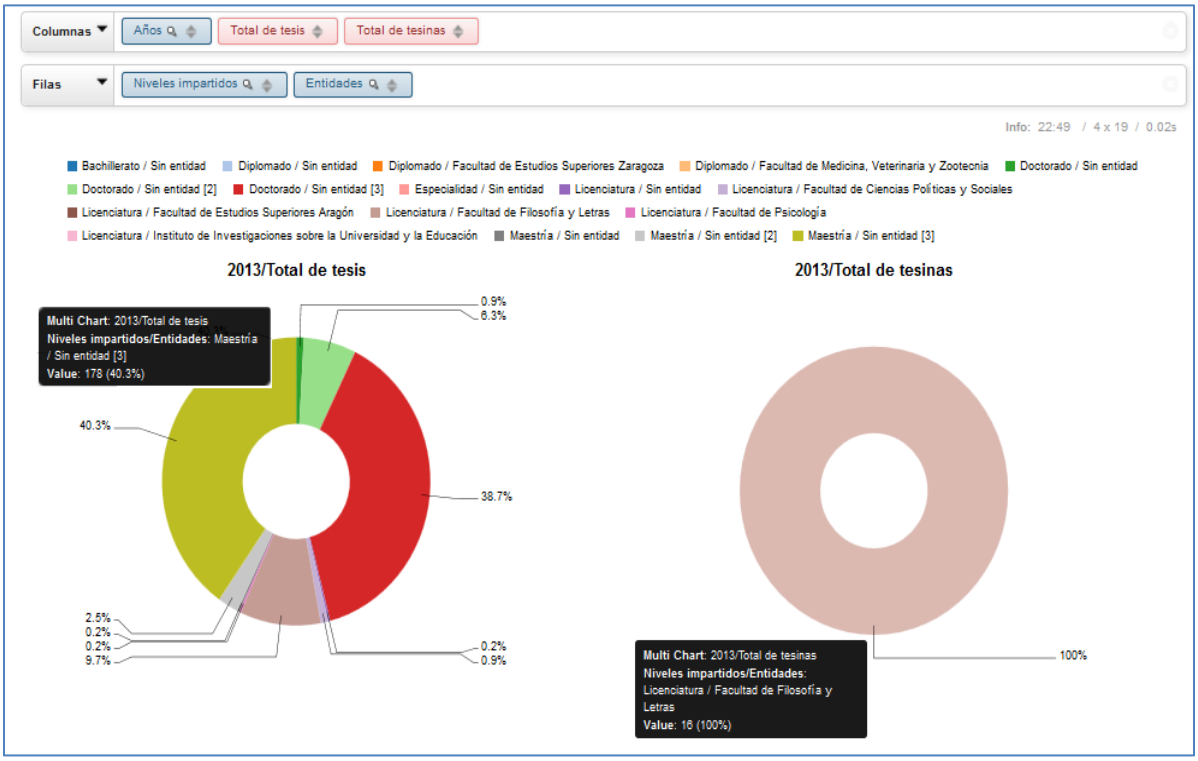
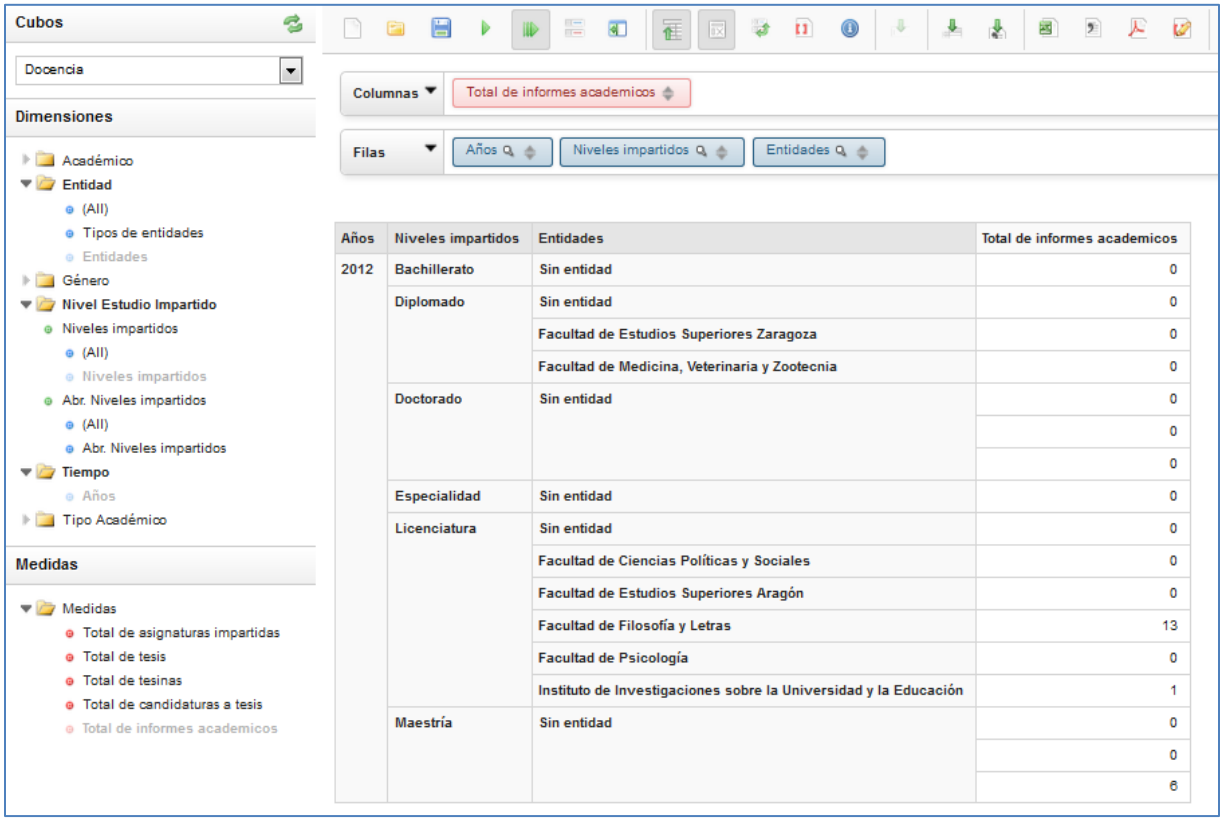


FIGURA 5.5.1.r. Requerimiento 6 (b)

Analizando las FIGURAS 5.5.1.q. y 5.5.1.r. anteriores vemos que, el mayor número de tesis es dirigida por académicos, con un nivel de estudio impartido de Maestría y no cuentan con una entidad adscripta en el 2013. También se observa que sólo 16 tesinas son dirigidas por académicos, con un nivel de estudio de licenciatura en este año.

Requerimiento 7

"El IISUE solicitó conocer el total de informes emitidos por sus académicos, en base al nivel de estudio impartido y a la entidad adscripta."



Años	Niveles impartidos	Entidades	Total de informes academicos
2012	Bachillerato	Sin entidad	0
	Diplomado	Sin entidad	0
		Facultad de Estudios Superiores Zaragoza	0
		Facultad de Medicina, Veterinaria y Zootecnia	0
	Doctorado	Sin entidad	0
			0
			0
	Especialidad	Sin entidad	0
	Licenciatura	Sin entidad	0
		Facultad de Ciencias Políticas y Sociales	0
		Facultad de Estudios Superiores Aragón	0
		Facultad de Filosofía y Letras	13
		Facultad de Psicología	0
		Instituto de Investigaciones sobre la Universidad y la Educación	1
	Maestría	Sin entidad	0
			0
			6

FIGURA 5.5.1.s. Requerimiento 7 (a)

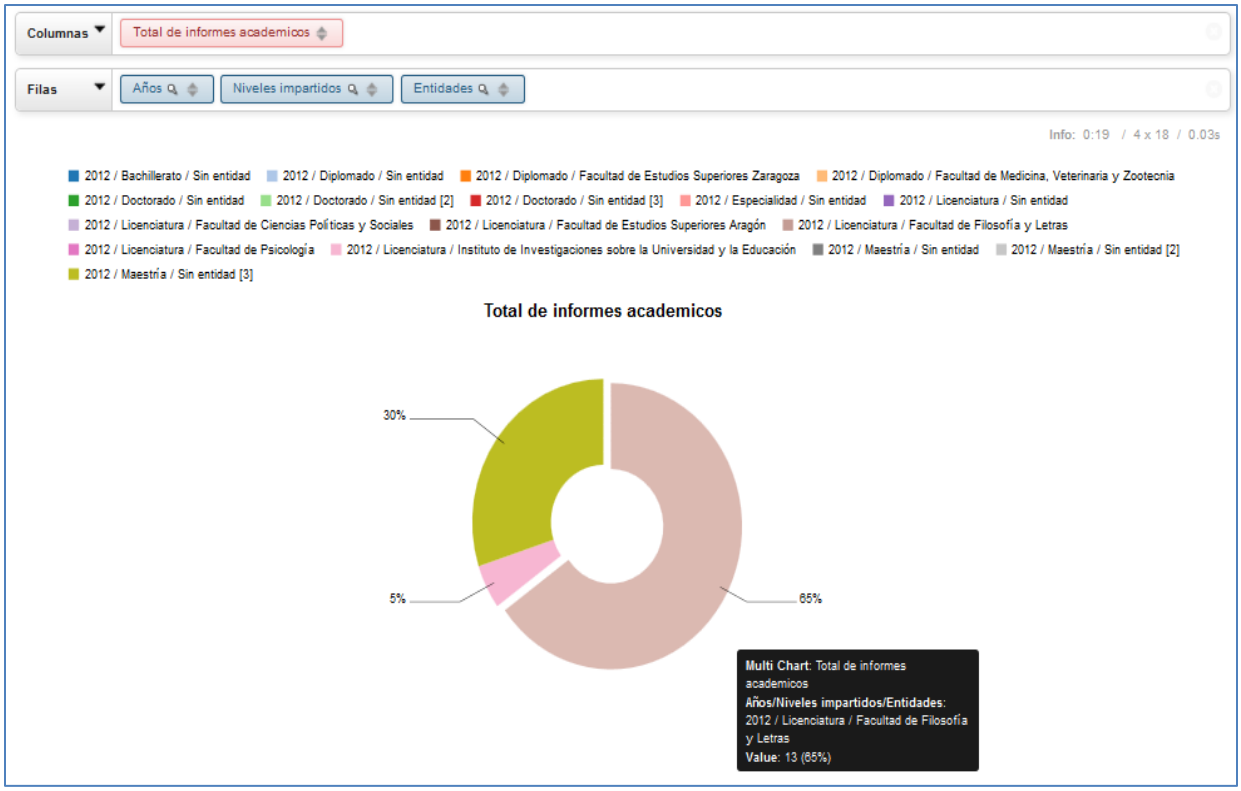
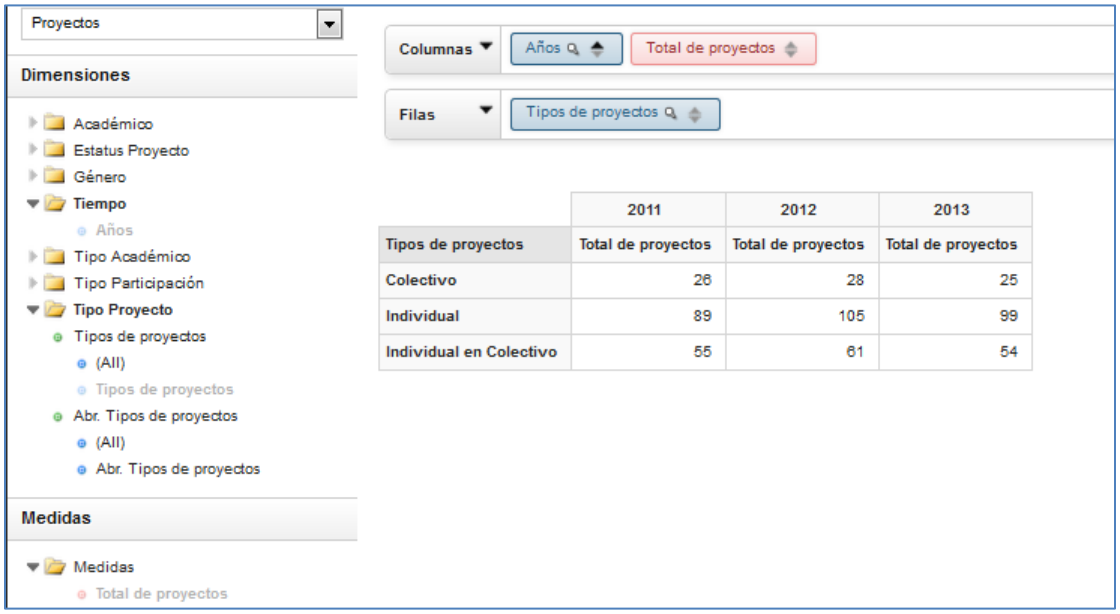


FIGURA 5.5.1.t. Requerimiento 7 (b)

Observamos que en el 2012, la entidad con mayores informes emitidos por académicos con un nivel de estudios de licenciatura, es la Facultad de Filosofía y Letras.

Requerimiento 8

"El IISUE requiere la información de sus diferentes tipos de proyectos (colectivo e individual) por cada año."



	2011	2012	2013
Tipos de proyectos	Total de proyectos	Total de proyectos	Total de proyectos
Colectivo	26	28	25
Individual	89	105	99
Individual en Colectivo	55	61	54

FIGURA 5.5.1.u. Requerimiento 8 (a)

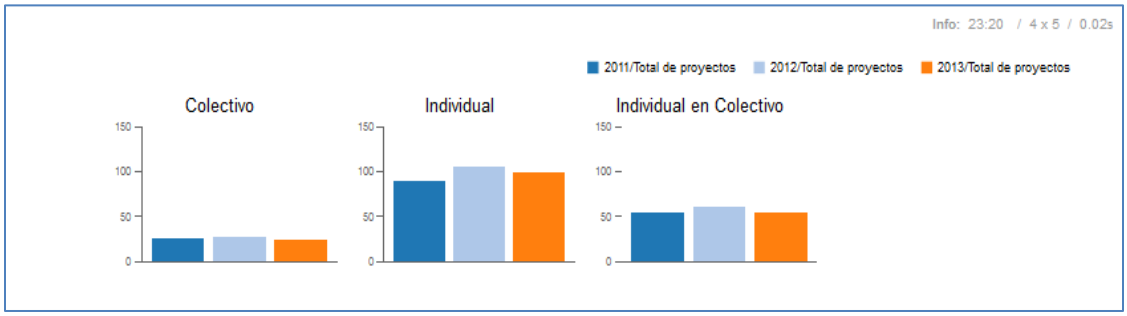
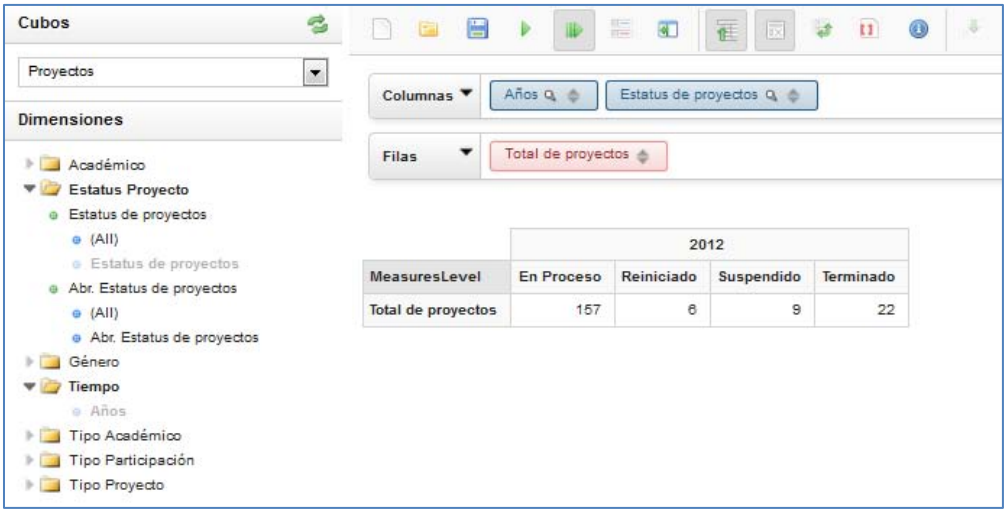


FIGURA 5.5.1.v. Requerimiento 8 (b)

En el 2011, 2012 y el 2013 hay mayor número de proyectos individuales que colectivos.

Requerimiento 9

"El IISUE solicitó conocer el total de proyecto en base a su estatus (en proceso, reiniciado, suspendido y terminado), por año."



	2012			
MeasuresLevel	En Proceso	Reiniciado	Suspendido	Terminado
Total de proyectos	157	6	9	22

FIGURA 5.5.1.w. Requerimiento 9 (a)

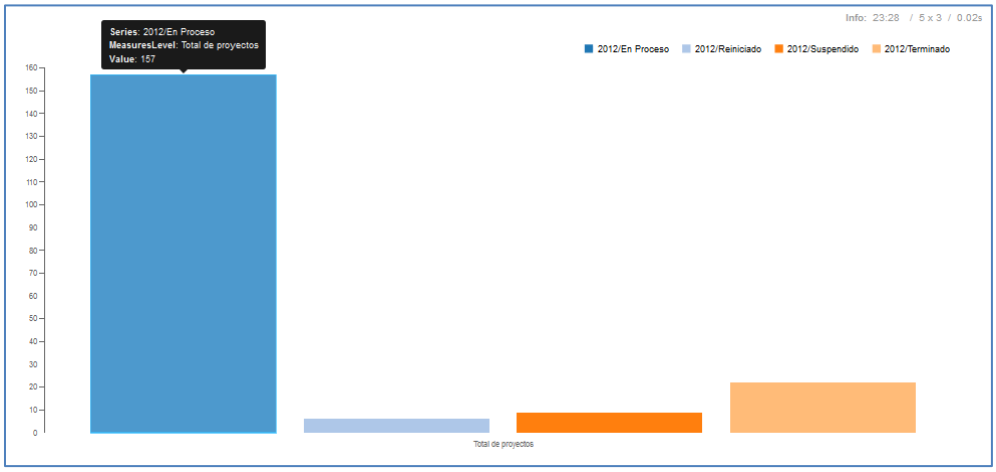


FIGURA 5.5.1.x. Requerimiento 9 (b)

En el 2012, la mayoría de los proyectos en los que se encuentran trabajando los académicos están en proceso, seguido de los proyectos terminados y solo una pequeña parte lo han suspendido o reiniciado.

Requerimiento 10

"El IISUE necesita conocer el total de proyectos en base al tipo de participación de cada uno de sus académicos, por año."

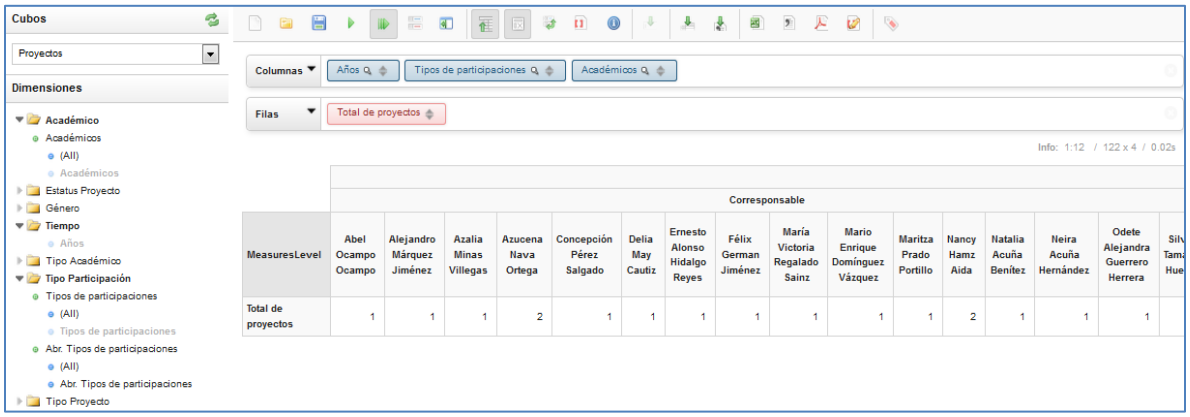


FIGURA 5.5.1.y. Requerimiento 10 (a)

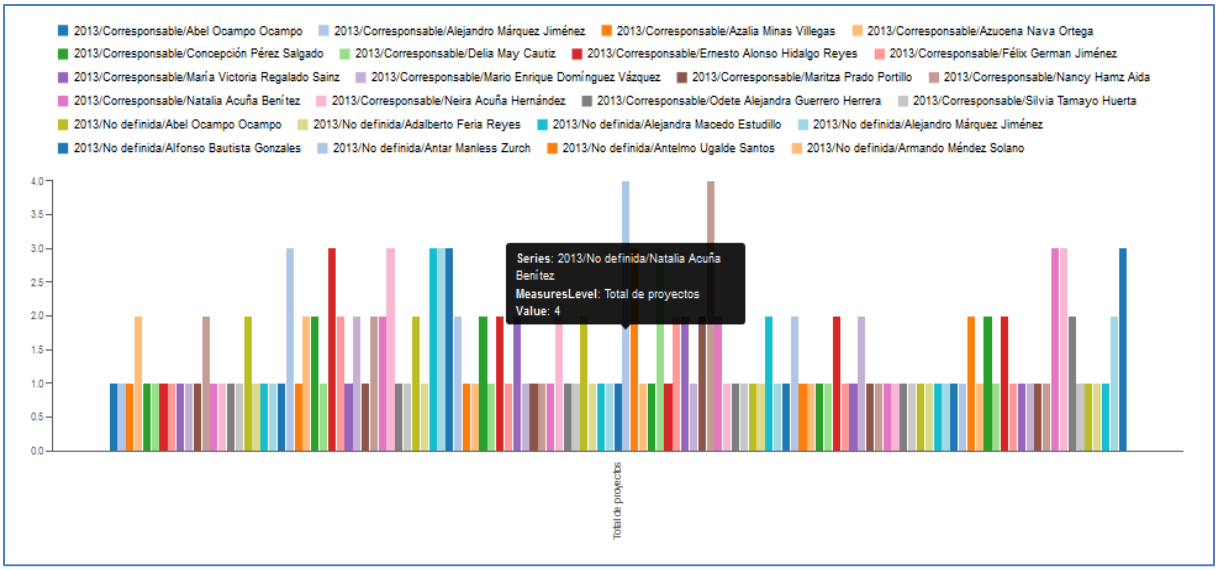


FIGURA 5.5.1.z. Requerimiento 10 (b)

Por medio de las imágenes vemos que los académicos Natalia Acuña Benítez y Verónica Pulido Rodríguez con una participación no definida, son las que están en más proyectos en el 2013. El IISUE solicitó conocer el total de proyectos en base al tipo de participación de cada uno de sus académicos, por año.

CAPÍTULO 6. Conclusiones

El desarrollo de un DWH suele ser una tarea compleja, debido a los procesos que se deben de llevar a cabo para su elaboración, en este sentido se destaca el poder integrar distintas fuentes de información con diferentes nomenclaturas, formatos, reglas de negocio, etc. Sin embargo, un DWH planeado y diseñado en base a una metodología resuelve varios de estos conflictos, ya que te indican el camino que se debe seguir para llegar a los objetivos principales que tiene un DWH y establece ciertos tiempos.

La metodología de Kimball proporciona una base empírica y metodológica adecuada para las implementaciones de almacén de datos pequeños y medianos, dada su gran versatilidad y su enfoque ascendente, que permite construir los DWH en forma escalonada. Además presenta una serie de herramientas, tales como documentos de alcance, plan de proyecto y documentos anexos (matriz de procesos, diagramas de flujo, etc.), que proporcionan una gran ayuda para iniciarse en el ámbito de la construcción de un Data Warehouse.

Por otra parte el DWH es una de las tecnologías que están orientadas a los usuarios de nivel ejecutivo, que tengan la facultad de tomar decisiones, por ende suele ser también uno de los sistemas de gran impacto sobre la organización o empresa, ya que se convierte en la base de toda la información analítica y detallada que brindará todo una amplia perspectiva de los aspectos más importantes para las empresas (producción, ventas, pagos, etc.). Resolviendo así la parte en donde los usuarios que lo requieran, puedan conseguir y elaborar reportes de información específica, de manera intuitiva, sencilla y gráfica.

Los desarrollos de DWH actualmente ya son comunes en varias organizaciones o empresas, al igual que las herramientas que permiten su desarrollo y manejo, están en una fase óptima porque permite dar solución a los distintos problemas con los que se puede llegar a encontrar en la creación de un DWH, esto es porque los procesos de ETL pueden ser realizados con herramientas totalmente gráficas, sencillas de utilizar, robustas en cuanto a seguridad y multiplataforma, que nos da el plus de no depender tanto del uso de scripts programados que suelen ser bastante extensos y no permiten integrar distintas fuentes de información.

Un punto a favor de este proyecto es que el constante desarrollo, lo volverá un sistema de suma importancia que incrementará sus alcances siendo de gran ayuda en establecer, modificar o redirigir los rumbos estratégicos que las empresas requieren para apuntalarse como líderes en su rubro.

Como objetivos cumplidos en esta tesis, fue que se elaboró un análisis, un diseño y una implementación de la tecnología de Data Warehouse para el IISUE, y sobre ello determinamos que es una tecnología que aún rebasa demasiado los alcances que se presentaron en esta tesis, es decir, el Data Warehouse puede soportar la integración y almacenamiento de nuevas fuentes de información que puedan dar un vista mucho más amplia de lo que originalmente se construyó. Otros objetivos que se lograron fue el de explicar las metodologías de desarrollo de un DW: Kimball e Inmon, llegando a la conclusión de que ambas metodologías son buenas y que para adoptar una depende básicamente de las necesidades de la organización ya sea privada o pública, inclusive puede llegar a suceder que se haga una mezcla de ambas.

El proyecto como tal, resolvió las necesidades expuestas por el Personal Administrativo del IISUE, cabe destacar que entre sus necesidades era el procesar su fuente de información denominada SIAH, es por ello que se construyó un proceso de ETL seguro, robusto y de calidad, ya que su principal ventaja es la de emitir correos electrónicos a los administradores si llegara a presentar alguna falla o anomalía dentro de la ejecución del mismo. A parte se logró rediseñar la presentación de la vista que nos presenta el software de Pentaho (Pentaho BI Server 4.8 Community Edition), el hacer uso de este software nos otorga inmediatamente un sistema totalmente robusto, multiplataforma, seguro y de alto rendimiento, y por supuesto también brinda a los usuarios un sistema de análisis de datos de fácil acceso e intuitivo y sencillo de usar (elaborar reportes ad-hoc, generar gráficas y exportar los datos a distintos formatos).

Es síntesis las necesidades y requerimientos principales fueron cubiertas por este proyecto, sin embargo aún sigue manteniéndose como sistema piloto, debido a los costos que implicaría mantener un sistema en producción y también el hecho de que no se cuenta con el acceso a otras fuentes de información que pudieran enriquecer el contenido de nuestro DWH. Aun así, el uso de esta tecnología mejorará, automatizara y permitirá conseguir resultados positivos dentro de la toma de decisiones de la planeación y administración de recursos del IISUE, porque ofrecerá a los usuarios conocimiento cuantitativo del cómo se han ido desarrollando las actividades, proyectos e investigaciones elaboradas por el Personal Académico a lo largo de un año, y proveerá una base sobre la cual generar reportes personalizados que den a conocer tales resultados.

Determinamos con base a una pequeña investigación y también en base al conocimiento que poseemos, que las herramientas de software libre sobre el ámbito de Data Warehouse actualmente son bastantes estables, seguras y maduras para dar soporte a este tipo de desarrollos y permiten conseguir los resultados necesarios como lo harían las herramientas de software privado. Este último punto lo dejamos abierto para aquel o aquellos que quieran profundizar más en el uso de software privado.

Finalmente este proyecto permitió que nos empapáramos de información que será de gran utilidad para nuestra vida profesional, pues nos ofreció la oportunidad de tener contacto con personas que requerían de la creación de una propuesta útil, que a su vez nos proporcionó una perspectiva clara y real de cómo resolver alguna carencia que tiene una empresa. Es de mencionar que el IISUE y personal de esta institución amablemente nos apoyó con sus puntos de vista y fueron parte importante en la definición de este trabajo.

CAPÍTULO 7. Trabajo a futuro

Consideramos necesario plantear cuestiones a futuro que pueden fortalecer este desarrollo de DWH:

- Realizar el análisis de otras fuentes de información y añadir fuentes externas que mejoren la calidad de los datos sobre el DWH.

- Diseñar nuevos Data Marts que den soporte en la toma de decisiones con respecto a diferentes aspectos.

- Proporcionar y configurar un ambiente de producción para poner en marcha la plataforma.

- Realizar mejoras en el proceso de ETL (incluya nuevas reglas para mantener integridad sobre los datos).

- Realizar minería de datos con el fin de obtener conocimientos acerca de los datos que se encuentran en el DWH.

- Elaborar reportes predeterminados que puedan ser impresos en formato PDF y enriquezcan más la exploración de los datos.

- Mejorar la metodología utilizada en este proyecto (Kimball Top-Down) para lograr objetivos a corto plazo.

Glosario

Aplicación web. Interfaz gráfica que permite al usuario efectuar operaciones sobre los datos que en ella contiene, mediante el uso de un navegador y la red.

Base de Datos. Conjunto de información organizada perteneciente a un determinado contexto.

ETL (Extraer, transformar y cargar). Proceso que permite a las organizaciones mover datos desde múltiples fuentes, reformatearlos, limpiarlos y cargarlos en otra base de datos.

Java. Lenguaje de programación de propósito general, concurrente, multiplataforma, orientado a objetos y basado en clases.

Javascript. Lenguaje de programación interpretado, definido como orientado a objetos y es ejecutado principalmente sobre los navegadores web.

MDX (expresiones multidimensionales). Lenguaje de consulta para bases de datos multidimensionales sobre cubos OLAP, se utiliza en Business Intelligence para generar reportes para la toma de decisiones basados en datos históricos, con la posibilidad de cambiar la estructura o rotación del cubo.

Mondrian. Servidor OLAP escrito en Java. Permite analizar grandes cantidades de datos almacenados en bases de datos SQL de una forma interactiva sin necesidad de escribir las sentencias que serían necesarias para ello en SQL. Entre otras características destaca la compatibilidad con el MDX (expresiones multidimensionales) y el lenguaje de consulta XML para análisis.

OLAP (procesamiento analítico en línea). Bases de datos orientadas al procesamiento analítico. Este análisis suele implicar, generalmente, la lectura de grandes cantidades de datos para llegar a extraer algún tipo de información útil: tendencias de ventas, patrones de comportamiento de los consumidores, elaboración de informes complejos… etc. Este sistema es típico de los Data Marts.

OLTP (Procesamiento de Transacciones En Línea). Bases de datos orientadas al procesamiento de transacciones. Una transacción genera un proceso atómico (que debe ser validado con un *commit*, o invalidado con un *rollbac*k), que puede involucrar operaciones de inserción, modificación y borrado de datos. El proceso transaccional es típico de las bases de datos operacionales.

Saiku. Es una herramienta OLAP destinada a usuarios finales de Pentaho, que permite visualizar y realizar análisis de datos de forma fácil e intuitiva.

SGBD Relacional. Sistema de Gestión de Bases de Datos Relacionales. Es un conjunto de programas que permiten el almacenamiento, modificación y extracción de la información de una base de datos.

Sistema informático. Conjunto de partes (hardware, software y recurso humano) que funcionan interrelacionándose entre sí para poder así procesar información de un determinado ámbito.

Software Privado. Es aquel software en el que su uso, redistribución o modificación está prohibida, o requiere de un permiso expreso por parte del titular del software.

SQL (lenguaje de consulta estructurado). Lenguaje declarativo de acceso a bases de datos relacionales que permite especificar diversos tipos de operaciones en ellas. Una de sus características es el manejo del álgebra y el cálculo relacional que permiten efectuar consultas con el fin de recuperar de forma sencilla información de interés de bases de datos, así como hacer cambios en ella.

XML (lenguaje de marcas extensible). Lenguaje de marcas desarrollado por el World Wide Web Consortium (W3C) utilizado para almacenar datos en forma legible. A diferencia de otros lenguajes, XML da soporte a bases de datos, siendo útil cuando varias aplicaciones deben comunicarse entre sí o integrar información.

Bibliografía y mesografía

[1]. Página web del Instituto de Investigaciones sobre la Universidad y la Educación. [Fecha de consulta: Enero 2014]. Disponible en: http://www.iisue.unam.mx

[2]. Cuenta de Facebook del Instituto de Investigaciones sobre la Universidad y la Educación. [Fecha de consulta: Enero de 2014]. Disponible en: http://www.facebook.com/pages/Iisue-UNAM/114818071927586

[3]. Willian H. Inmon. (2005). "Building the Data Warehouse". Indiana: Wiley Publishing, Inc.

[4]. Ralph Kimball, Joe Caserta. (2004). "The Data Warehouse ETL Toolkit". Indiana: Wiley Publishing, Inc.

[5]. Salvador Ramos. (2011). "Microsoft Business Intelligence: vea el cubo medio lleno". España: SolidQ™ Press.

[6]. Surajit Chaudhuri, Umeshwar Dayal. "An Overview of Data Warehousing and OLAP Technology". Página web de la Universidad de Lahore de Ciencias de la Gestión (LUMS). [Fecha de consulta: Diciembre 2014]. Disponible en: http://suraj.lums.edu.pk/~cs544w03/chaudhuri.pdf

[7]. Josep Lluís Cano. "Business Intelligence: Competir con Información" Página web de La Escuela Superior de Administración y Dirección de Empresas (ESADE). [Fecha de consulta: Diciembre 2014]. Disponible en: http://www.esade.edu/profesorado/joseplluis.cano

[8]. Ana Buigues. "Arquitectura de un Data Warehouse", Blog de Ana Buigues. [Fecha de consulta: Diciembre 2014]. Disponible en: http://anabuigues.com/2010/03/05/arquitectura-de-un-data-warehouse/

[9]. Bernabeu R. Dario. “HEFESTO”, Blog TGX - HEFESTO.
[Fecha de consulta: Diciembre 2014]. Disponible en:
http://sourceforge.net/projects/bihefesto/files/Hefesto/HEFESTO.gz/download

[10]. Página web de la Free Software Foundation, Inc.
[Fecha de consulta: Diciembre 2014]. Disponible en:
http://www.gnu.org/philosophy/free-sw.es.html

[11]. Roland Bouman, Jos van Dongen. (2009). “Pentaho Solutions: Business Intelligence and DataWarehousing with Pentaho and MySQL”. Indianapolis: Wiley Publishing, Inc.

[12]. María Carina Roldán. (2010). “Pentaho 3.2 Data Integration”. Birmingham: Packt Publishing, Inc.

[13]. Página web de PostgreSQL, “PostgreSQL.org.es”
[Fecha de consulta: Diciembre 2014]. Disponible en:
http://www.postgresql.org.es/sobre_postgresql#historia

[14]. Página web de PostgreSQL. “Programmer’s Guide”
[Fecha de consulta: Diciembre 2014]. Disponible en:
http://www.postgresql.org/files/documentation/pdf/7.3/programmer-7.3.2-A4.pdf

[15]. Página web de Pentaho. “Documentación del Usuario de Pan”
[Fecha de consulta: Junio 2014]. Disponible en:
http://wiki.pentaho.com/pages/viewpage.action?pageId=11869063

[16]. Página web de Pentaho. “Documentación del Usuario de Kitchen”
[Fecha de consulta: Junio 2014]. Disponible en:
http://wiki.pentaho.com/pages/viewpage.action?pageId=11869458

[17]. Página web de Datalytics. “Encontradole la vuelta a JPivot (2da parte)” [Fecha de consulta: Diciembre 2013]. Disponible en: http://www.datalytics.com/blog/2011/09/encontrandole-la-vuelta-a-jpivot-2da-parte/

[18]. Página web de 2ndquadrant. “PostgreSQL – La historia hasta ahora” [Fecha de consulta: Diciembre 2013]. Disponible en: http://2ndquadrant.com/es/postgresql/postgresql-la-historia-hasta-ahora/

[19]. Página web de Daniel Pecos Martínez. “PostGreSQL vs. MySQL” [Fecha de consulta: Febrero 2014]. Disponible en: http://danielpecos.com/documents/postgresql-vs-mysql/

[20]. Página web de Enterprisedb. “Postgres Plus 8.4 vs. MySQL 5.5” [Fecha de consulta: Febrero 2014]. Disponible en: http://get.enterprisedb.com/whitepapers/Postgres_Plus_8.4_vs_MySQL_5.5.pdf

[21]. Ing. Gabriela Betzabé Lizárraga Ramírez. “Apuntes de la materia de Depósitos de Datos”

[22]. Página web de inteligenciadenegociosval. “Metodología Kimball” [Fecha de consulta: Febrero 2014]. Disponible en: http://inteligenciadenegociosval.blogspot.mx/2014/01/metodologia-de-kimball.html

[23]. Página web de wikipedia. “Pentaho” [Fecha de consulta: Diciembre 2013]. Disponible en: http://es.wikipedia.org/wiki/Pentaho

Anexos

Anexo I. Manual de instalación y configuración de Pentaho BI Server 4.8 Community Edition

Este manual tiene por objetivo guiar al usuario para la instalación y configuración de **Pentaho BI Suite 4.8 Community Edition**, sobre la plataforma **Centos 5.8** (previamente instalado en una Máquina Virtual con Oracle VM VirtualBox) y utilizando como RDBMS **PostgreSQL 9.1**.

Requerimientos mínimos

Para realizar una instalación básica de Pentaho BI Server 4.8 recomendamos contar con al menos el siguiente hardware y software especificado:

- Hardware
  - ✓ RAM: 1 GB o más.
  - ✓ Disco Duro: 100 GB o más.
  - ✓ Procesador: Doble núcleo o superior.

- Software
  - ✓ Sistema Operativo: Centos 5.8 o superior (Instalado previamente).
  - ✓ Pentaho: BI Server 4.8 Community Edition.
  - ✓ RDBMS: PostgreSQL 9.1.
  - ✓ Java: JDK (Java Development Kit) version 1.6.X.

Configuración de Java

Antes de poder iniciar con la instalación y configuración del servidor de Pentaho hay que verificar que la JVM (Java Virtual Machine) esté instalada y que las variables de entorno estén configuradas correctamente.

Para hacer la verificación de que la JVM está instalada correctamente, ingrese a modo consola y conéctese como usuario root y ejecute el comando *java -versión*. Si la JVM está instalada aparecerá algo como lo siguiente:

```
[Saga@localhost /]$ su root
Contraseña:
[root@localhost /]# java -version
java version "1.6.0_22"
OpenJDK Runtime Environment (IcedTea6 1.10.4) (rhel-1.24.1.10.4.el5-i386)
OpenJDK Client VM (build 20.0-b11, mixed mode)
[root@localhost /]#
```

Si le aparece un mensaje parecido al de la imagen anterior o no está instalada ninguna versión de Java, debe actualizar a la versión oficial, en este manual se trabajará con la versión de **Java 1.6.u45**.

Actualización de Java

Primero se descarga el paquete de instalación del jdk de acuerdo a la arquitectura que tenga nuestro S.O., en nuestro caso es de 32 bits entonces se descarga el i586 desde la siguiente URL:

JDK 1.6.u45: http://www.oracle.com/technetwork/java/javasebusiness/downloads/java-archive-downloads-javase6-419409.html#jdk-6u45-oth-JPR

Se cambia a la ruta */usr/java/* y se le asignan privilegios de ejecución al paquete de instalación mediante el comando *chmod +x*. Posteriormente se ejecuta el paquete jdk-6u45-linux-i586.bin mediante el comando *sh*, como se muestra en la imagen.

```
[root@localhost saga]# cd /usr/java/
[root@localhost java]# chmod +x /media/JARP/Java/jdk-6u45-linux-i586.bin
[root@localhost java]# sh /media/JARP/Java/jdk-6u45-linux-i586.bin
```

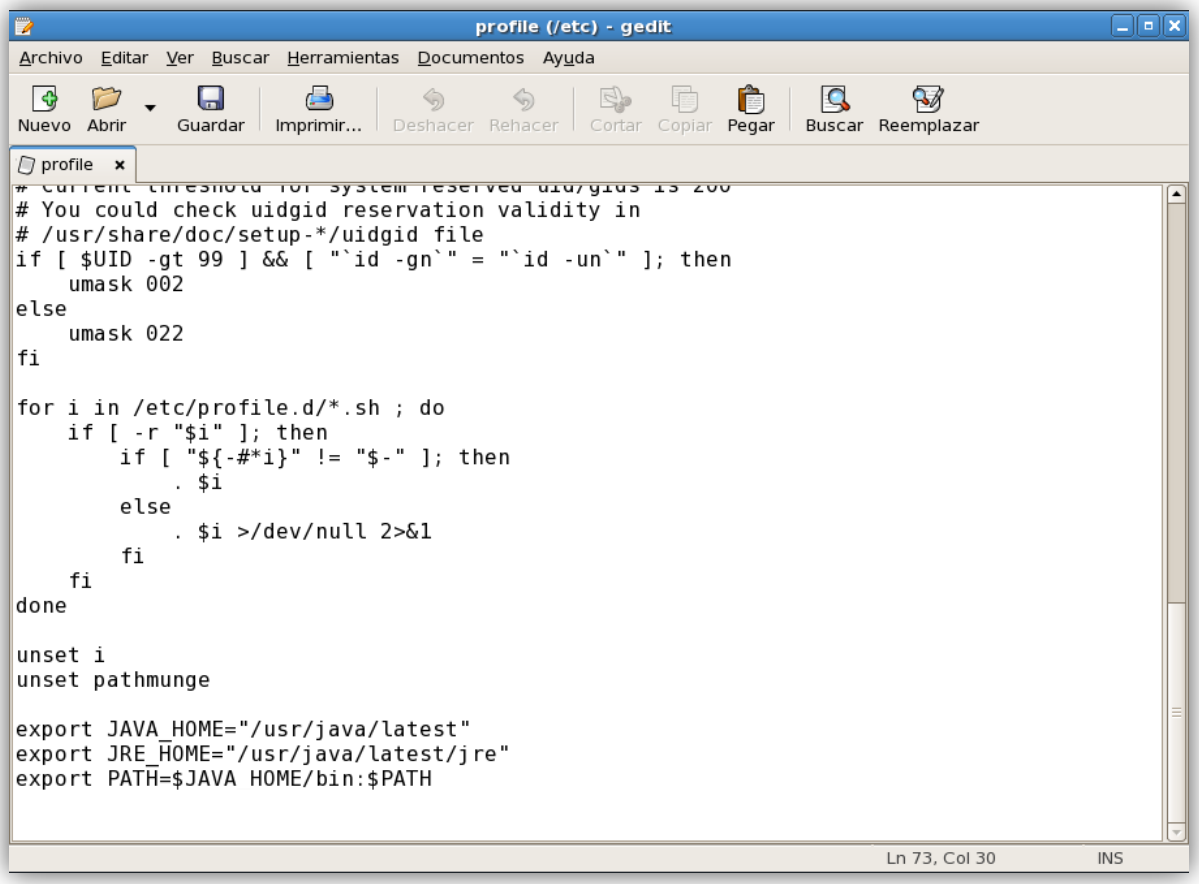
Después de ejecutar el paquete de instalación, se crea un link simbólico, esto nos sirve por si tenemos instaladas varias versiones del mismo JDK y por lo tanto se ligan únicamente la versión que deseamos utilizar. Esto se realiza mediante el comando *ln –s*.

```
[root@localhost java]# ln -s /usr/java/jdk1.6.0_45/ latest
[root@localhost java]#
```

Para la ejecución de los nuevos comandos de Java sólo tienen que configurarse utilizando el comando *alternatives --install*.

```
[root@localhost saga]# /usr/sbin/alternatives --install /usr/bin/java java /usr/
java/latest/jre/bin/java 200000
[root@localhost saga]# /usr/sbin/alternatives --install /usr/bin/javaws javaws /
usr/java/latest/jre/bin/javaws 200000
[root@localhost saga]# /usr/sbin/alternatives --install /usr/bin/javac javac /us
r/java/latest/bin/javac 200000
[root@localhost saga]# /usr/sbin/alternatives --install /usr/bin/jar jar /usr/ja
va/latest/bin/jar 200000
[root@localhost saga]#
```

Finalmente después de haber hecho la instalación se deben agregar las nuevas variables de entorno. Se abre una terminal y se ejecuta el comando *gedit /etc/profile*.



El agregar estas variables de entorno nos permite tener un correcto funcionamiento del software de Pentaho.

Una vez realizado todo el proceso ya se debería de contar con la versión oficial de Java y para verificarlo volvemos a ejecutar el comando *java –version*.

```
[root@localhost saga]# java -version
java version "1.6.0_45"
Java(TM) SE Runtime Environment (build 1.6.0_45-b06)
Java HotSpot(TM) Client VM (build 20.45-b01, mixed mode, sharing)
[root@localhost saga]#
```

Instalación de Pentaho BI Server 4.8 Community Edition

Primero se descarga el paquete BI Server Pentaho 4.8:

http://sourceforge.net/projects/pentaho/files/Business%20Intelligence%20Server/4.8.0-stable/

Después de descargar, se debe crear el directorio */opt/pentaho/server_bi_4.8* para almacenar el servidor de Pentaho.

```
[root@localhost ~]# cd /opt/
[root@localhost opt]# mkdir pentaho
[root@localhost opt]# cd pentaho/
[root@localhost pentaho]# mkdir server_bi_4.8
```

Mover o copiar el archivo en donde se descargó, al directorio creado.

```
[root@localhost pentaho]# cp /media/JARP/Pentaho/BI\ Server/Pentaho\ BI\ Server\
tar.gz server_bi_4.8/
[root@localhost pentaho]# cd server_bi_4.8/
[root@localhost server_bi_4.8]# ls
biserver-ce-4.8.0-stable.tar.gz
[root@localhost server_bi_4.8]#
```

Para descomprimir el archivo se utiliza el comando *tar –xzf*.

```
[root@localhost server_bi_4.8]# tar -xzf biserver-ce-4.8.0-stable.tar.gz
[root@localhost server_bi_4.8]# ls
administration-console  biserver-ce  biserver-ce-4.8.0-stable.tar.gz
[root@localhost server_bi_4.8]# rm biserver-ce-4.8.0-stable.tar.gz
rm: ¿borrar el fichero regular «biserver-ce-4.8.0-stable.tar.gz»? (s/n) s
[root@localhost server_bi_4.8]# ls
administration-console  biserver-ce
[root@localhost server_bi_4.8]#
```

Esto creará dos carpetas, una llamada *biserver-ce* y la otra *administration-console*. Se asignan los permisos de ejecución a los archivos de procesamiento (*.sh).

```
[root@localhost server_bi_4.8]# cd biserver-ce/
[root@localhost biserver-ce]# chmod +x *.sh
[root@localhost biserver-ce]# cd ..
[root@localhost server_bi_4.8]# cd administration-console/
[root@localhost administration-console]# chmod +x *.sh
[root@localhost administration-console]# cd ..
[root@localhost server_bi_4.8]#
```

Realizadas las tareas anteriores, podemos arrancar por primera vez el servidor y la consola de administración de Pentaho, para ello hay que colocarse en el directorio “biserver-ce” y ejecutar el comando *./start-pentaho.sh*.

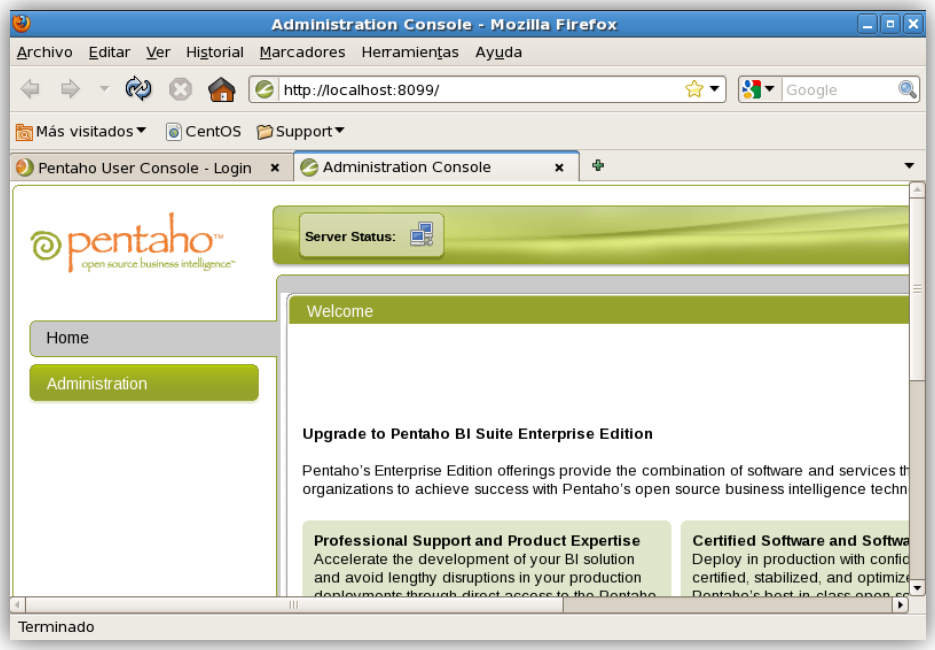
```
[root@localhost server_bi_4.8]# ls
administration-console  biserver-ce
[root@localhost server_bi_4.8]# cd biserver-ce/
[root@localhost biserver-ce]# ./start-pentaho.sh
/opt/pentaho/server_bi_4.8/biserver-ce
/opt/pentaho/server_bi_4.8/biserver-ce
DEBUG: Using JAVA_HOME
DEBUG: _PENTAHO_JAVA_HOME=/usr/java/latest
DEBUG: _PENTAHO_JAVA=/usr/java/latest/bin/java
------------------------------------------------------------------------------
------------
The Pentaho BI Platform now contains a version checker that will notify you
when newer versions of the software are available. The version checker is enable
d by default.
For information on what the version checker does, why it is beneficial, and how
it works see:
http://wiki.pentaho.com/display/ServerDoc2x/Version+Checker
Press Enter to continue, or type cancel or Ctrl-C to prevent the server from sta
rting.
You will only be prompted once with this question.
------------------------------------------------------------------------------
------------
[OK]:
Using CATALINA_BASE:   /opt/pentaho/server_bi_4.8/biserver-ce/tomcat
Using CATALINA_HOME:   /opt/pentaho/server_bi_4.8/biserver-ce/tomcat
Using CATALINA_TMPDIR: /opt/pentaho/server_bi_4.8/biserver-ce/tomcat/temp
Using JRE_HOME:        /usr/java/latest/jre
Using CLASSPATH:       /opt/pentaho/server_bi_4.8/biserver-ce/tomcat/bin/bootstr
ap.jar
[root@localhost biserver-ce]#
```

Posteriormente arranca la consola de administración de pentaho, para ello uno se coloca en el directorio “administration-console” y se ejecuta el comando *./start-pac-sh*.

```
[root@localhost server_bi_4.8]# ls
administration-console  biserver-ce
[root@localhost server_bi_4.8]# cd administration-console/
[root@localhost administration-console]# ./start-pac.sh
/opt/pentaho/server_bi_4.8/administration-console
/opt/pentaho/server_bi_4.8/administration-console
DEBUG: Using JAVA_HOME
DEBUG: _PENTAHO_JAVA_HOME=/usr/java/latest
DEBUG: _PENTAHO_JAVA=/usr/java/latest/bin/java
06:31:42,874 INFO  [JettyServer] Console is starting
06:31:43,082 INFO  [/] org.pentaho.pac.server.BrowserLocaleServlet-26613447: init
06:31:43,087 INFO  [/] org.pentaho.pac.server.DefaultConsoleServlet-21171036: init
06:31:43,089 INFO  [/] org.pentaho.pac.server.PacServiceImpl-21061094: init
06:31:43,089 INFO  [/] org.pentaho.pac.server.SchedulerServiceImpl-4115088: init
06:31:43,089 INFO  [/] org.pentaho.pac.server.SolutionRepositoryServiceImpl-26715004: in
it
06:31:43,089 INFO  [/] org.pentaho.pac.server.SubscriptionServiceImpl-24529889: init
06:31:43,089 INFO  [/] org.pentaho.pac.server.common.HibernateConfigurationServiceImpl-1
2067688: init
06:31:43,089 INFO  [/] org.pentaho.pac.server.common.JdbcDriverDiscoveryServiceImpl-2625
0401: init
06:31:43,130 INFO  [JettyServer] Console is now started. It can be accessed using http:/
/localhost.localdomain:8099 or http://127.0.0.1:8099
```

Se abre el navegador y colocamos la dirección http://localhost:8080/pentaho para poder ingresar a la aplicación de Pentaho BI Server 4.8 Community Edition.

Se abre otra pestaña en el navegador y se coloca la siguiente dirección http://localhost:8099. Para ingresar a la consola de administración de de Pentaho BI Server 4.8 Community Edition hay que especificar el usuario y contraseña que por defecto son "***admin***" y "***password***". Una vez logueado correctamente se verá la siguiente pantalla:



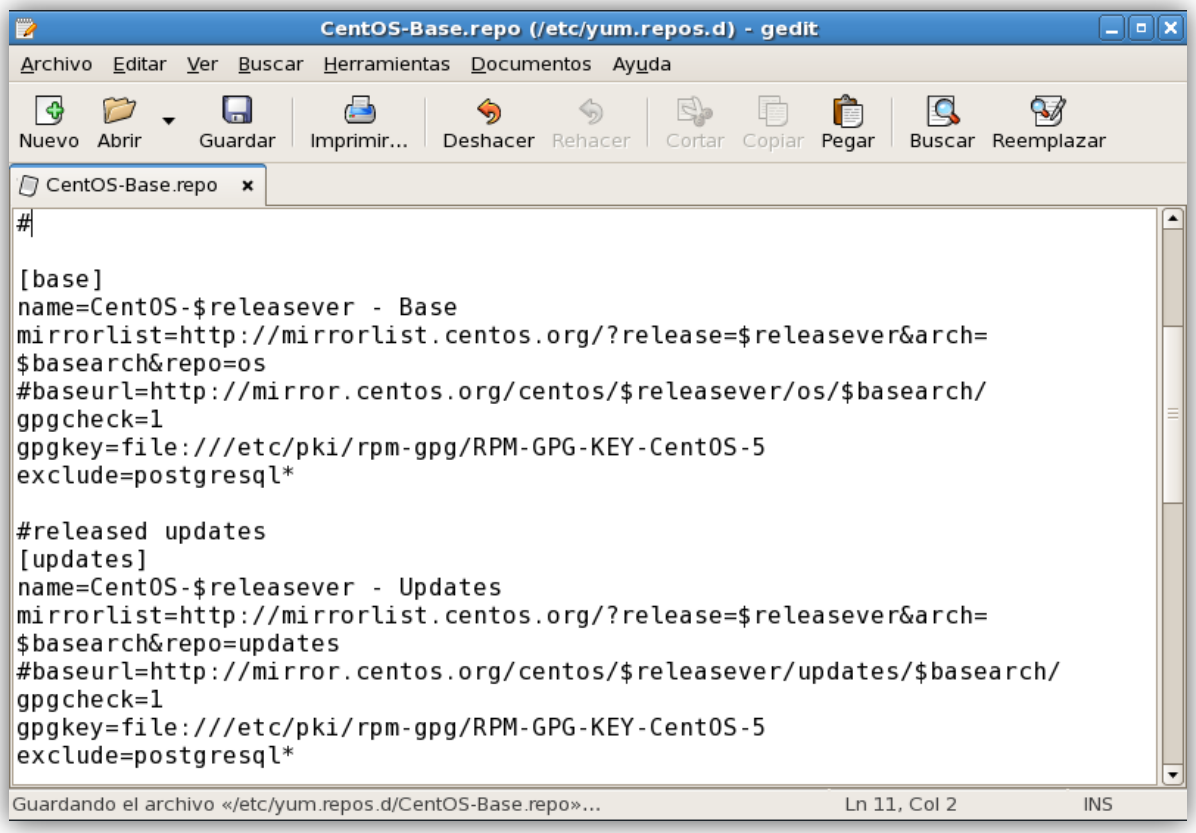
Instalación de PostgreSQL 9.1

Para instalar la versión 9.1 de PostgreSQL se tienen que deshabilitar los repositorios de PostgreSQL que vienen por defecto en el sistema operativo. Si no hiciéramos este paso es posible que sólo consiguiéramos instalar la versión que proporcionan los paquetes de la distribución en cuestión.

Para desactivar dichos repositorios se debe editar el fichero */etc/yum.repos.d/CentOS-Base.repo*

Nosotros lo haremos con gedit porque nos parece un editor de textos de propósito general, que enfatiza la simplicidad y facilidad de uso.

En la sección de [base] y [updates] se tiene que excluir PostgreSQL, para ello se añade *exclude=postgresql** al final de cada sección. Dichas secciones deben acabar pareciéndose a lo siguiente:

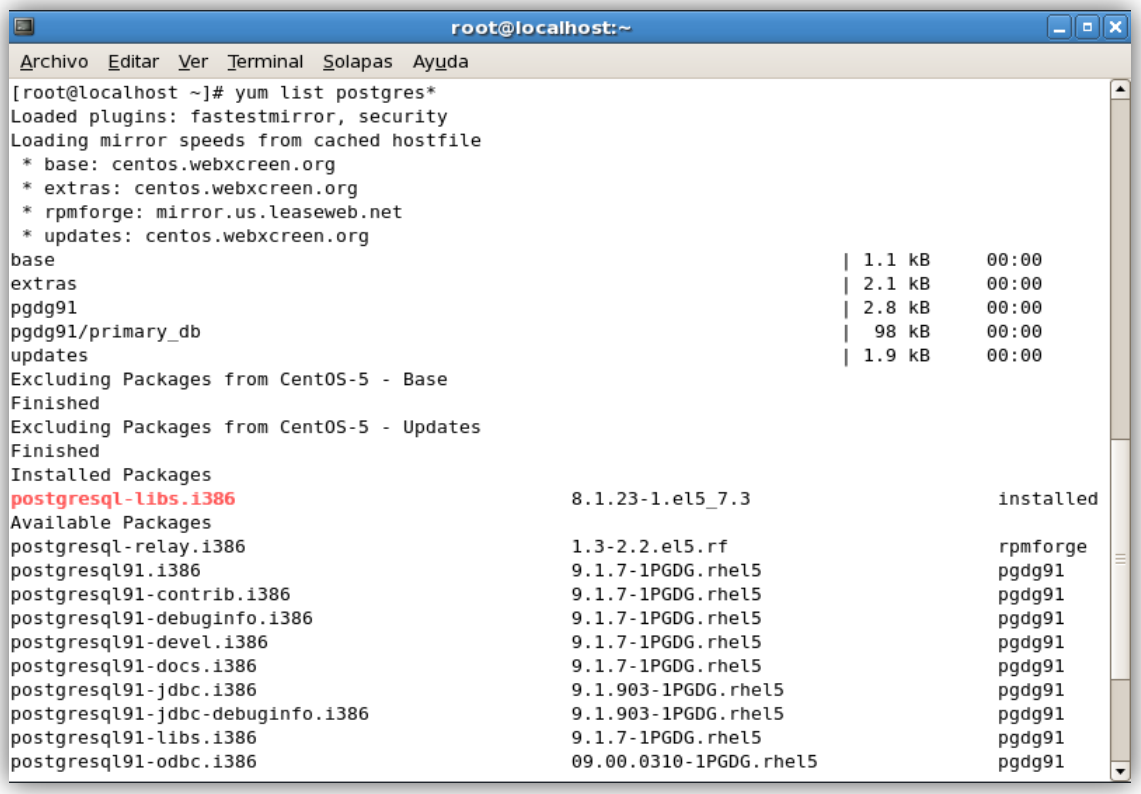


Una vez desactivados los repositorios por defecto se deben definir los nuevos repositorios. Para ello utilizaremos los RPM de la página de repositorios de PostgreSQL, es decir, en la página oficial de repositorios http://yum.postgresql.org se descarga el repositorio con la versión 9.1 y se instala el repositorio con el siguiente comando *rpm –ivh pgdg-centos91-9.1-4.noarch.rpm*.

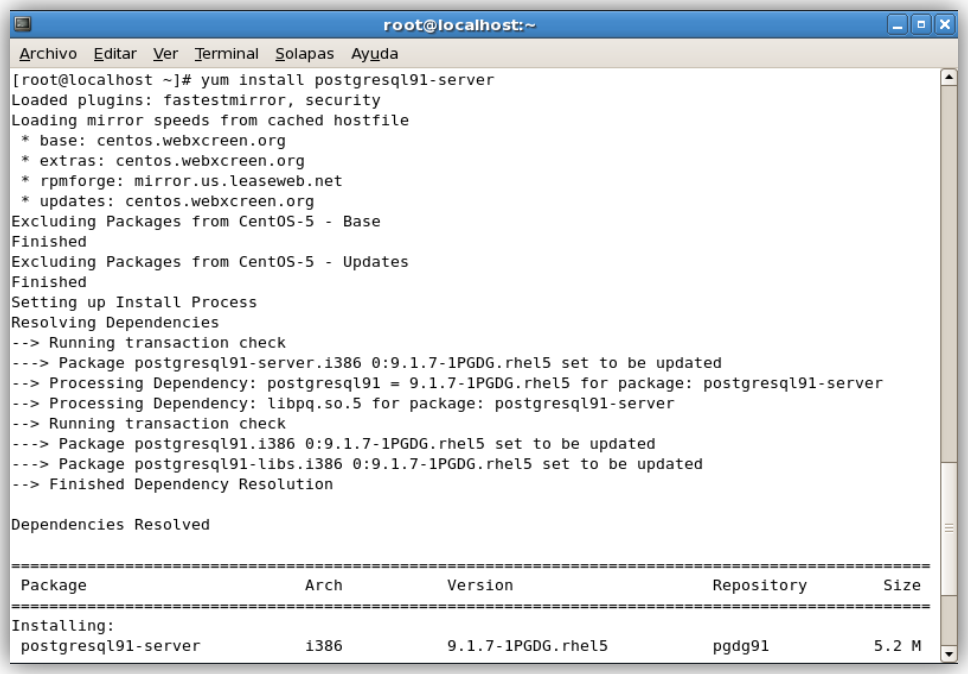
```
[root@localhost ~]# curl -O http://yum.postgresql.org/9.1/redhat/rhel-5-i386/pgdg-centos91-9.1-4.
noarch.rpm
  % Total    % Received % Xferd  Average Speed   Time    Time     Time  Current
                                 Dload  Upload   Total   Spent    Left  Speed
100  4964  100  4964    0     0  48742      0 --:--:-- --:--:-- --:--:-- 41095
[root@localhost ~]#
```

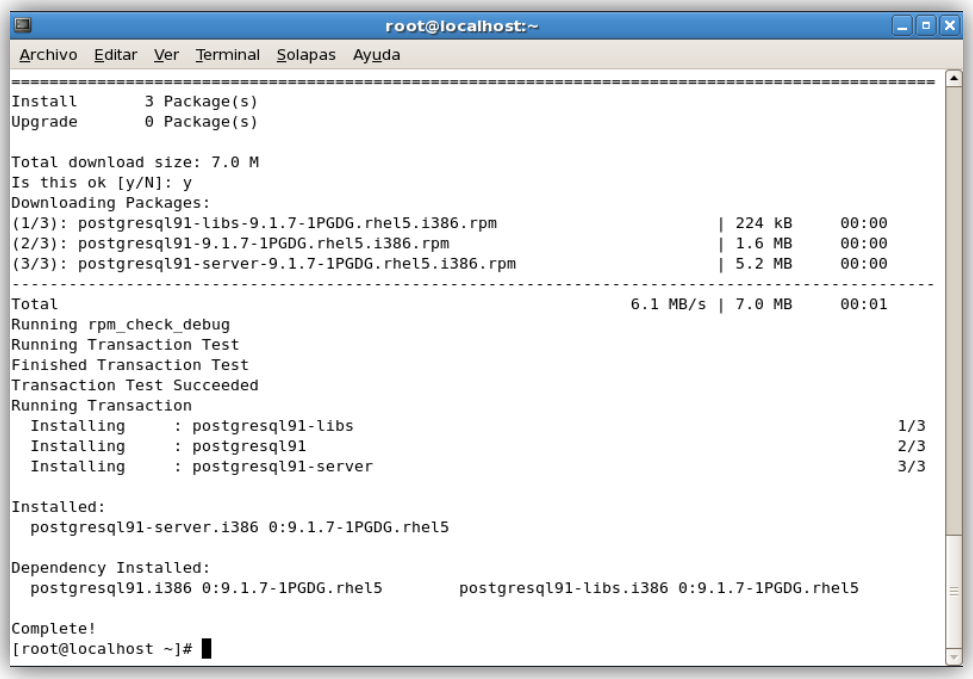
```
[root@localhost ~]# rpm -ivh pgdg-centos91-9.1-4.noarch.rpm
Preparando...                ########################################### [100%]
   1:pgdg-centos91           ########################################### [100%]
[root@localhost ~]#
```

Ahora se verifica la nueva lista de paquetes que se instalaron mediante el comando *yum list postgres**.



Una vez activados los nuevos paquetes de instalación, hay que seleccionar el servidor 9.1 de postgresql para que se ejecute su instalación con el comando *yum install postgresql91-server*.





Se inicializa el servidor postgresql 9.1.

```
[root@localhost ~]# service postgresql-9.1 initdb
Iniciando la base de datos:                                [  OK  ]
[root@localhost ~]#
```

Se agrega el servicio para que arranque al iniciar el sistema con el comando *chkconfig postgresql-9.1 on* y se reinicia el servicio con el comando *restart*.

```
[root@localhost ~]# chkconfig postgresql-9.1 on
[root@localhost ~]# service postgresql-9.1 restart
Parando el servicio  postgresql-9.1:                       [  OK  ]
Iniciando servicios postgresql-9.1:                        [  OK  ]
You have new mail in /var/spool/mail/root
[root@localhost ~]#
```

Con esto ya tendremos el servidor de base de datos como servicio al arrancar el sistema y aceptando conexiones en local, pero no podremos acceder al servidor desde conexiones remotas. Para habilitar las conexiones remotas hay que modificar dos ficheros y reiniciar el servicio.

El primer fichero que se va a modificar es */var/lib/pgsql/9.1/data/postgresql.conf*.

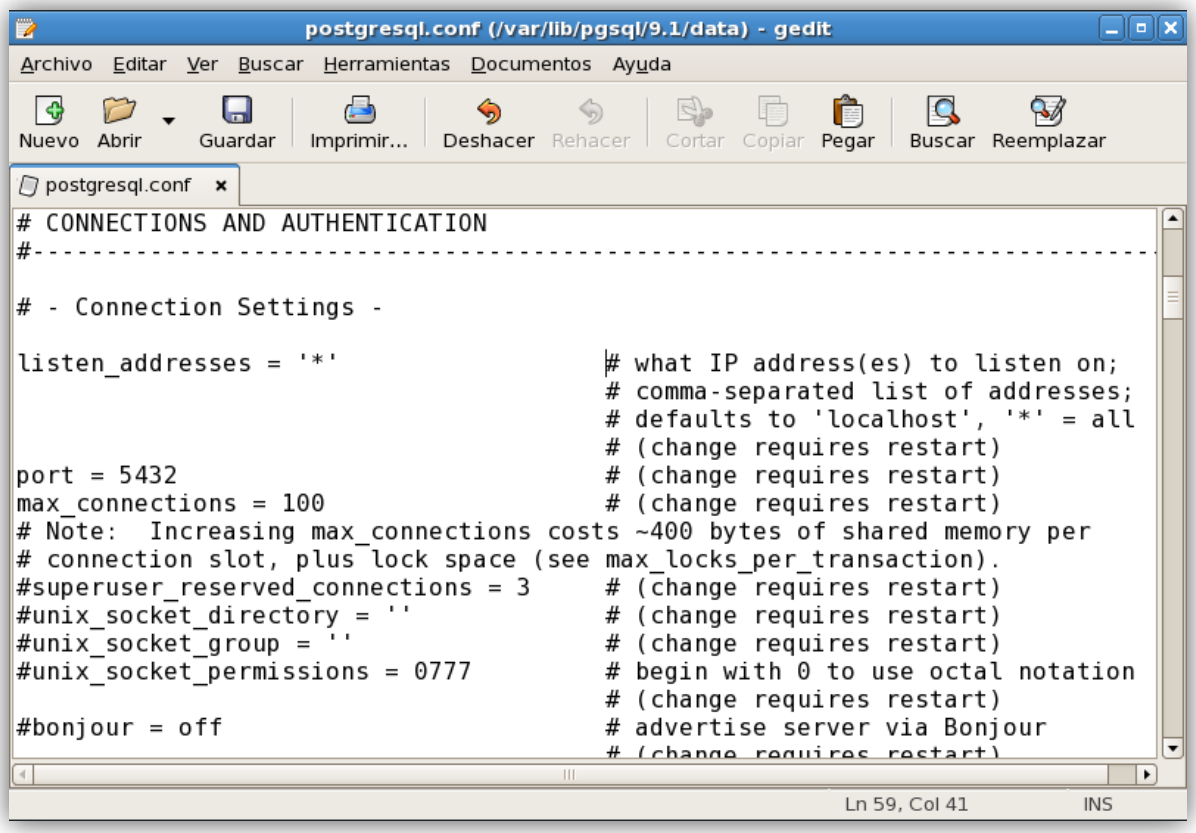
```
[root@localhost ~]# gedit /var/lib/pgsql/9.1/data/postgresql.conf
```

Se modifica la siguiente linea:

listen_addresses='localhost'

por

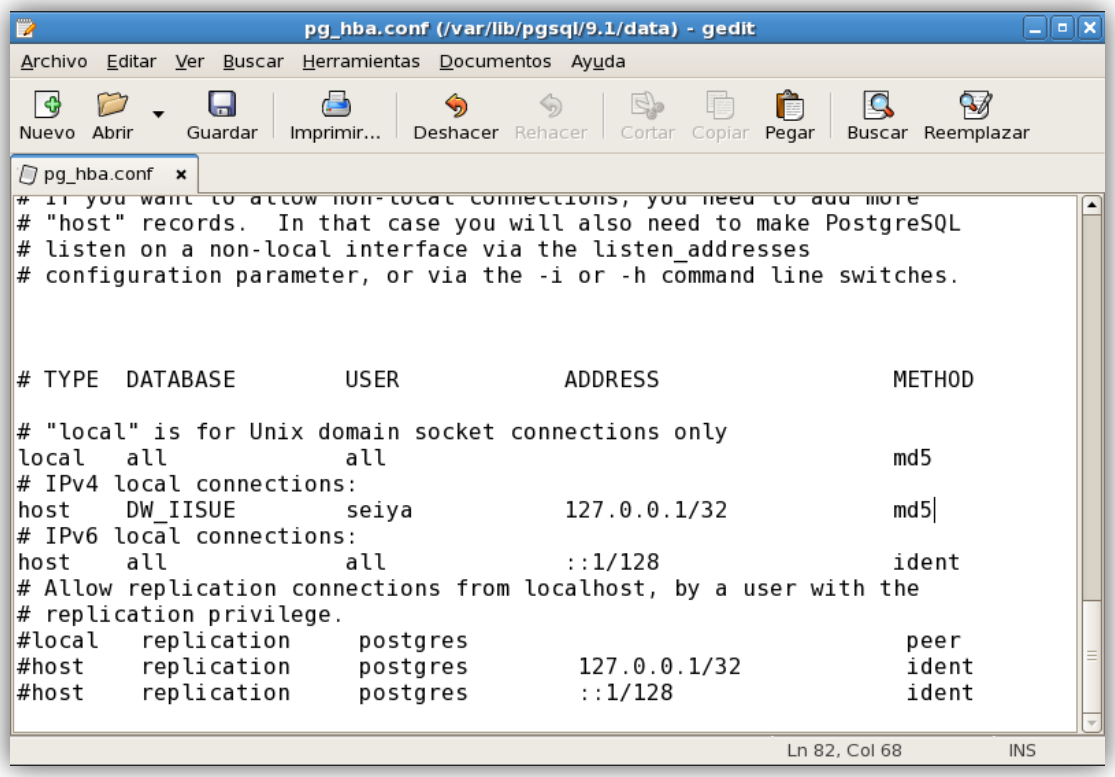
listen_addresses='*'


postgresql.conf (/var/lib/pgsql/9.1/data) - gedit
Archivo Editar Ver Buscar Herramientas Documentos Ayuda
Nuevo Abrir Guardar Imprimir... Deshacer Rehacer Cortar Copiar Pegar Buscar Reemplazar
postgresql.conf
CONNECTIONS AND AUTHENTICATION
#---
# - Connection Settings -
listen_addresses = '*' # what IP address(es) to listen on;
# comma-separated list of addresses;
# defaults to 'localhost', '*' = all
# (change requires restart)
port = 5432 # (change requires restart)
max_connections = 100 # (change requires restart)
# Note: Increasing max_connections costs ~400 bytes of shared memory per
# connection slot, plus lock space (see max_locks_per_transaction).
#superuser_reserved_connections = 3 # (change requires restart)
#unix_socket_directory = '' # (change requires restart)
#unix_socket_group = '' # (change requires restart)
#unix_socket_permissions = 0777 # begin with 0 to use octal notation
(change requires restart)
#bonjour = off # advertise server via Bonjour
Ln 59, Col 41
INS

El segundo fichero que se debe modificar es */var/lib/pgsql/9.1/data/pg_hba.conf*.

```
[root@localhost ~]# gedit /var/lib/pgsql/9.1/data/pg_hba.conf
```

En este fichero se tiene que dar el acceso al servidor postgresql 9.1, en este caso como es una Máquina Virtual no asignaremos una dirección IP, sino usaremos la dirección IP de localhost, es decir para IPv6 es ::.1/128 y para IPv4 es 127.0.0.a/32, luego se agrega el usuario y la base de datos a la cual queremos que se tenga acceso y cambiamos el método de autenticación por md5, sin embargo se pueden ir agregando tantas reglas de conexión se requieran, esto con el objetivo de tener un control más estricto en el acceso al servidor.



Una vez realizados estos cambios se tiene que reiniciar el servicio.

```
[root@localhost ~]# service postgresql-9.1 restart
Parando el servicio  postgresql-9.1:                       [  OK  ]
Iniciando servicios postgresql-9.1:                        [  OK  ]
[root@localhost ~]#
```

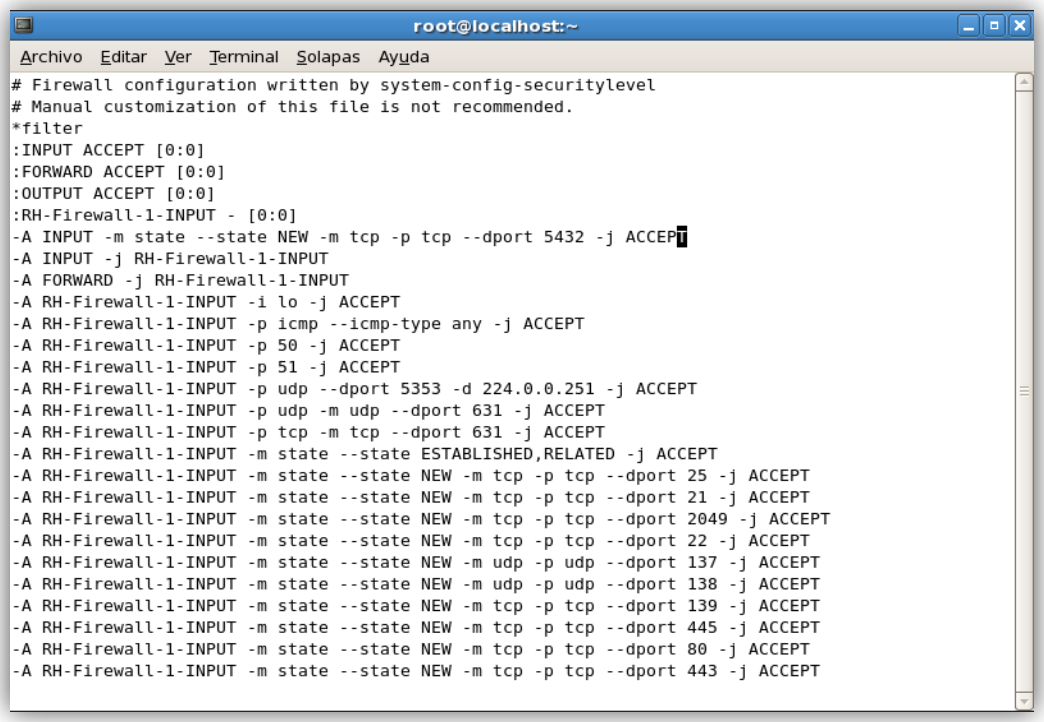
Para verificar si la instalación fue satisfactoria accedemos a la shell del servidor de bases de datos.

```
[root@localhost ~]# su - postgres
-bash-3.2$ psql -d template1 -U postgres
psql (9.1.7)
Digite «help» para obtener ayuda.
```

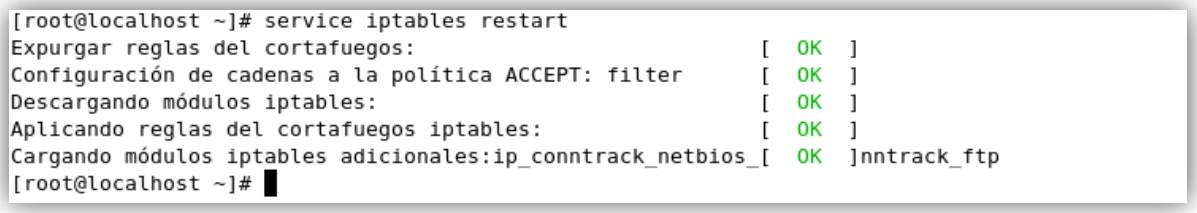
Si el acceso ha sido satisfactorio se cambia la contraseña del superusuario postgres y después de eso se verifica la versión instalada.

```
template1=# ALTER USER postgres WITH PASSWORD 'postgres';
ALTER ROLE
template1=# SELECT VERSION();
                                                 version

-------------------------------------------------------------------------------------------------------
-------
 PostgreSQL 9.1.7 on i686-pc-linux-gnu, compiled by gcc (GCC) 4.1.2 20080704 (Red Hat 4.1.2-52),
32-bit
(1 fila)

template1=#
```

Por último se agregará una regla de entrada al cortafuegos de Centos 5.8, esto para que se permitan las conexiones por el puerto 5432. Para ello se edita el fichero */etc/sysconfig/iptables* y se añade una nueva regla de entrada.



Y luego se reinicia el servicio de iptables.



Configuración de Pentaho BI Server 4.8 para utilizar con PostgreSQL 9.1

El servidor de Pentaho BI utiliza por defecto la base de datos HSQLDB[15]. Esta base de datos se crea y se carga en memoria cada vez que se inicia el servidor.

La idea es configurar Pentaho para que la base de datos del repositorio de usuarios y permisos esté en base de datos PostgreSQL y no siga utilizando la base de datos por defecto.

[15] HSQLDB (Hyperthreaded Structured Query Language Database) es un sistema gestor de bases de datos libre escrito en Java.

Esto se recomienda para la instalación de Pentaho en un entorno de producción, mientras que en un entorno de desarrollo o testing se puede utilizar la configuración que viene por defecto en Pentaho BI Server 4.8.

Es necesario descargar el driver JDBC para PostgreSQL. Para hacer esto se ingresa al sitio http://jdbc.postgresql.org/download.html y se descarga el driver de acuerdo a la versión de PostgreSQL y Java instalados.

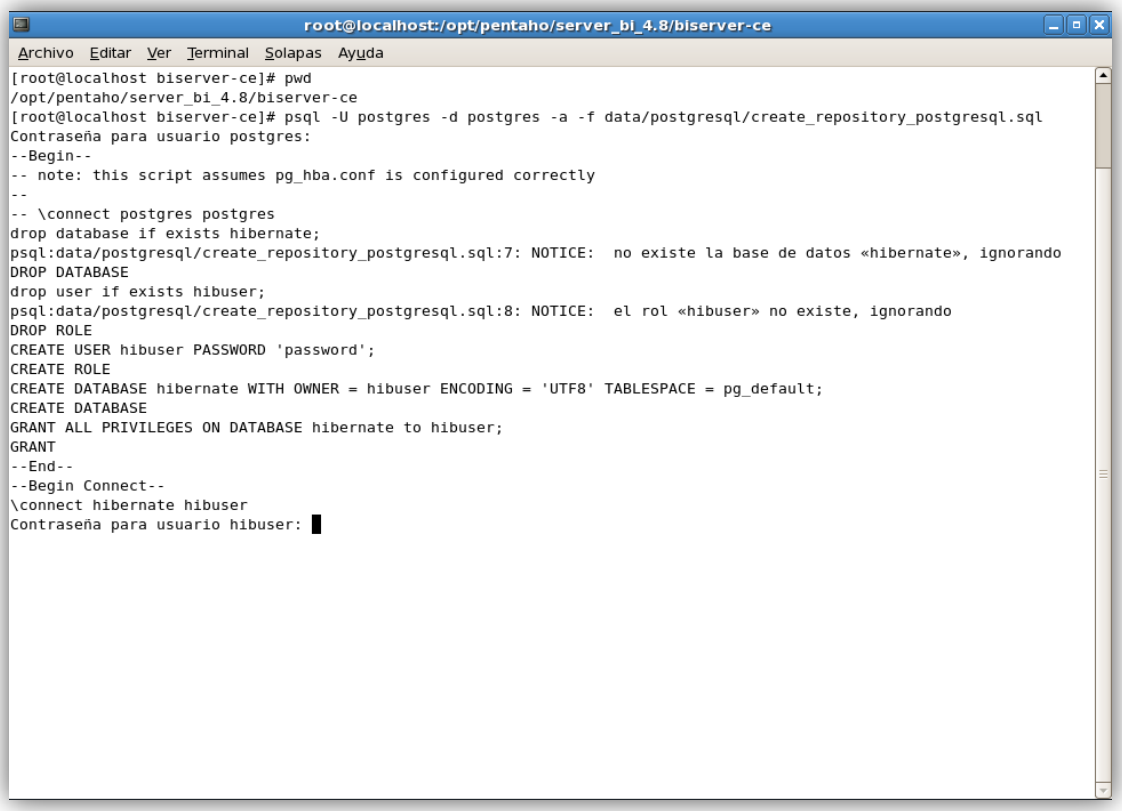
Después de descargar el driver de postgresql que es un archivo .jar se moverá dentro del directorio */opt/pentaho/server_bi_4.8/biserver-ce/tomcat/lib/*.

En el directorio */opt/pentaho/server_bi_4.8/biserver-ce/data/postgresql/* en donde habrá 5 scripts SQL:

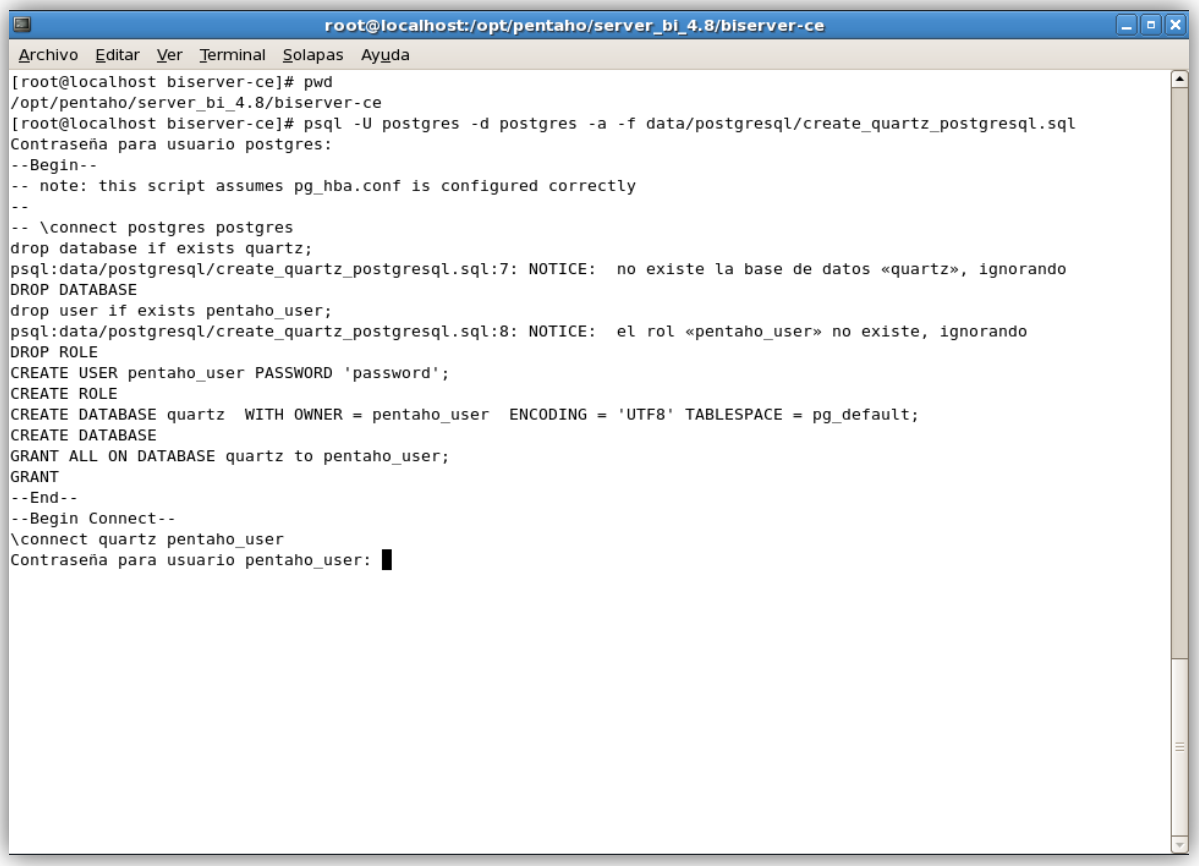
1. **create_repository_postgresql.sql:** Crea la base de datos "hibernate" y el usuario "hibuser".
2. **create_quartz_postgresql.sql:** Crea la base de datos "quartz" y el usuario "pentaho_user".
3. **create_sample_datasource_postgresql.sql:** Inserta el datasource de ejemplo de la base de datos "sampledata" que utiliza por default.
4. **migrate_quartz_postgresql.sql:** Actualiza la base de datos "quartz".
5. **migration.sql:** Actualiza la base de datos "hibernate"

Los scripts deben ser corridos en el orden en que aparecen listados a continuación. Para ejecutar los scripts, debemos hacerlo mediante los siguientes comandos:

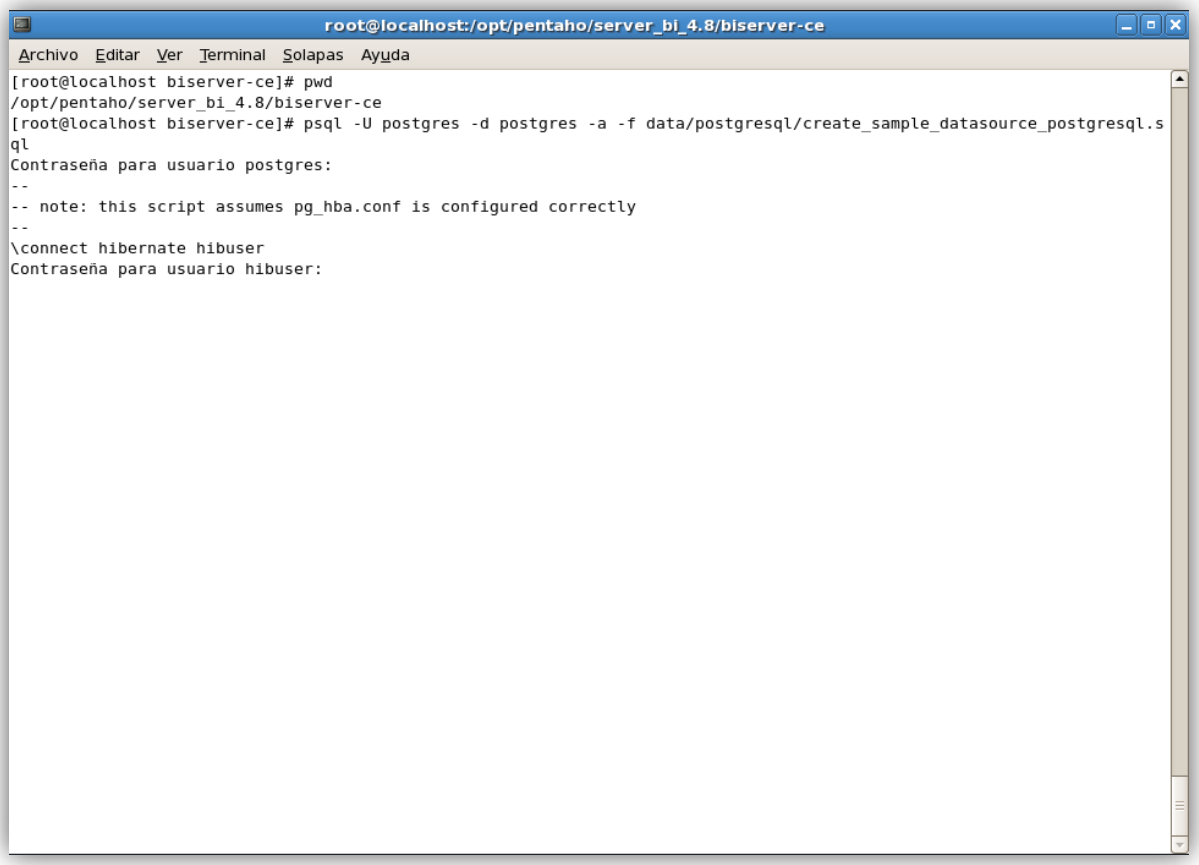
```
$ psql -U postgres –a -f data/postgresql/create_repository_postgresql.sql
Contraseña para usuario postgres: [ingresar la contraseña de postgres]
...
Contraseña para usuario hibuser: [ingresar la contraseña "password"]
...
```



```
$ psql -U postgres -a -f data/postgresql/create_quartz_postgresql.sql
Contraseña para usuario postgres: [ingresar la contraseña de postgres]
...
Contraseña para usuario pentaho_user: [ingresar la contraseña "password"]
...
```



```
$ psql -U postgres -a -f
data/postgresql/create_sample_datasource_postgresql.sql
Contraseña para usuario postgres: [ingresar la contraseña de postgres]
...
Contraseña para usuario hibuser: [ingresar la contraseña "password"]
...
```

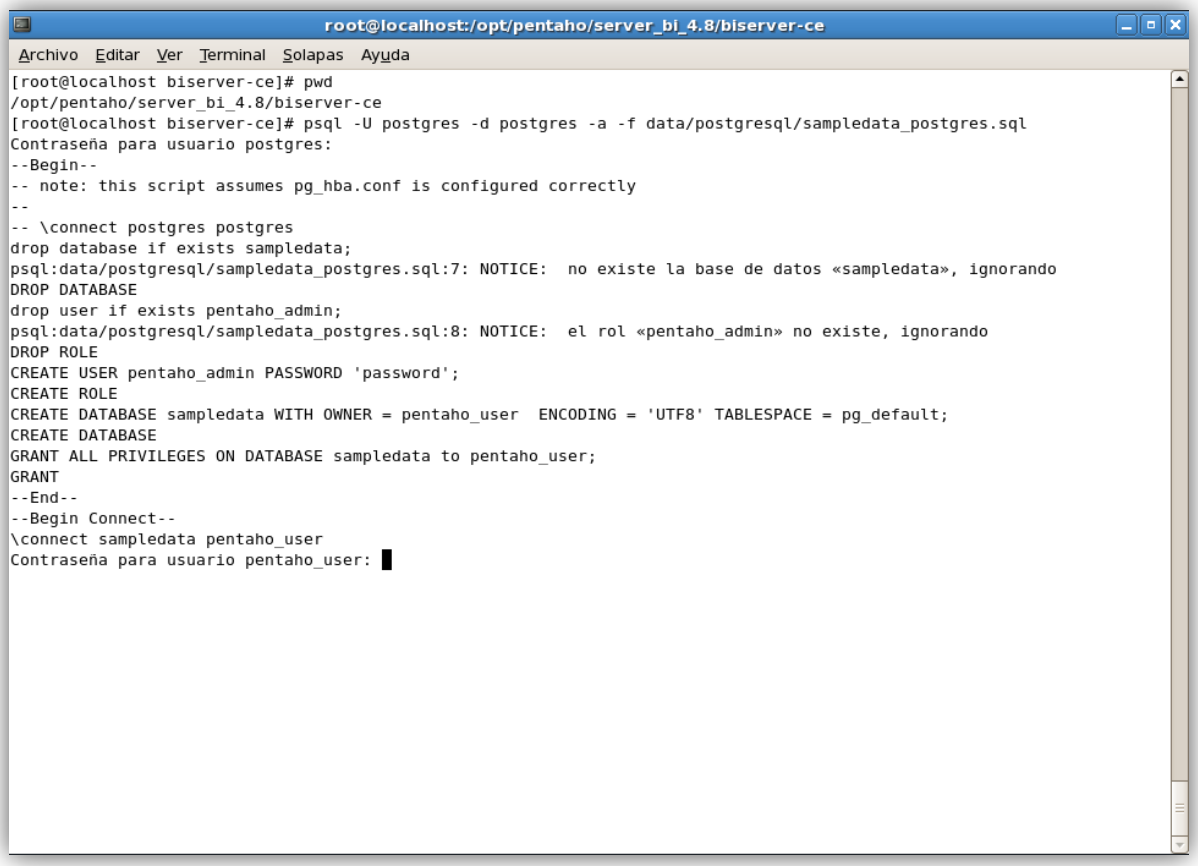


Y por último se descarga el archivo sql de la base de datos “sampledata” mediante el link oficial que se muestra a continuación:

http://source.pentaho.org/svnroot/legacy/pentaho-data/trunk/postgresql/sampledata_postgres.sql

Y de igual forma se ejecuta con el siguiente comando:

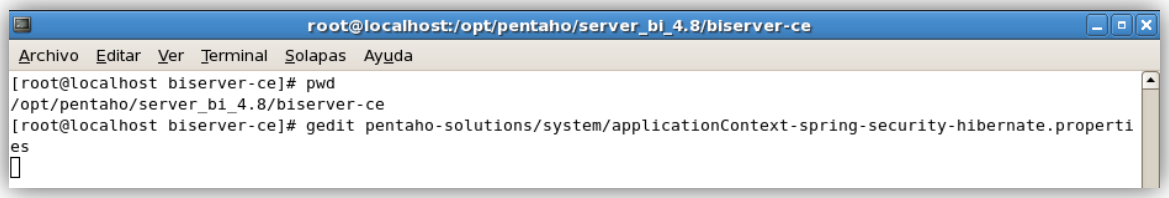
```
$ psql -U postgres -a -f data/postgresql/sampledata_postgresql.sql
Contraseña para usuario postgres: [ingresar la contraseña de postgres]
...
Contraseña para usuario pentaho_user: [ingresar la contraseña "password"]
...
```



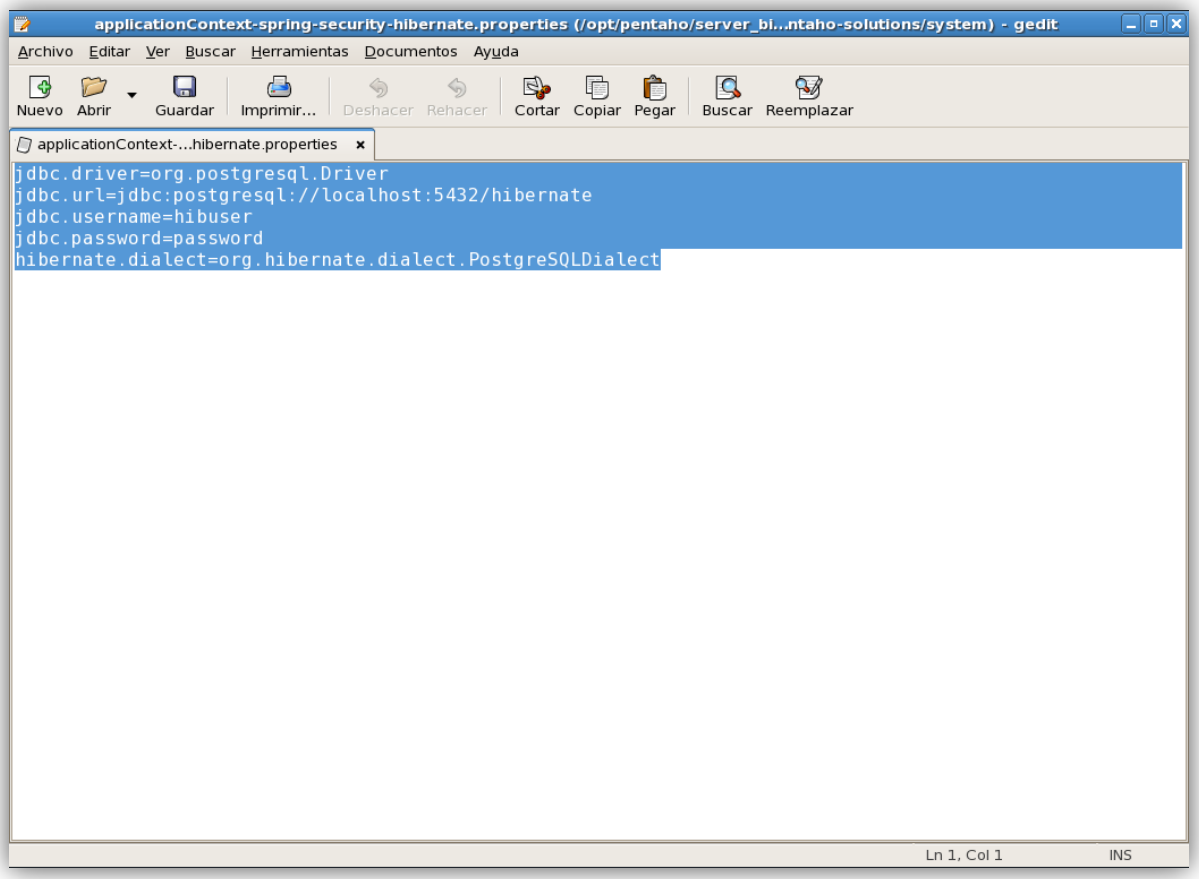
Configuración de la conexión JDBC

En esta sección configuramos la seguridad JDBC de Pentaho BI Server para utilizar como servidor de base de datos a PostgreSQL, esto significa que ahora el Pentaho BI Server apuntará a las base de datos "hibernate", "quartz" y "sampledata" del servidor PostgreSQL, en vez de las base de datos de HSQL que viene por defecto.

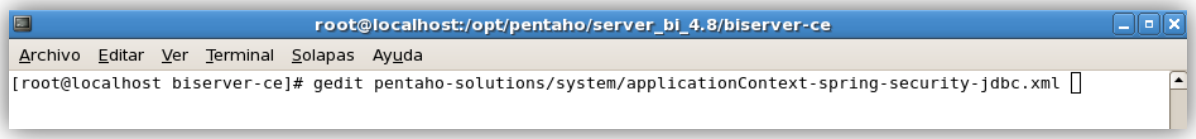
Se busca el archivo *applicationContext-spring-security-hibernate.properties* que se encuentra en el directorio *<path_biserver>/system/*. Donde *<path_biserver>* es */opt/pentaho/server_bi_4.8*



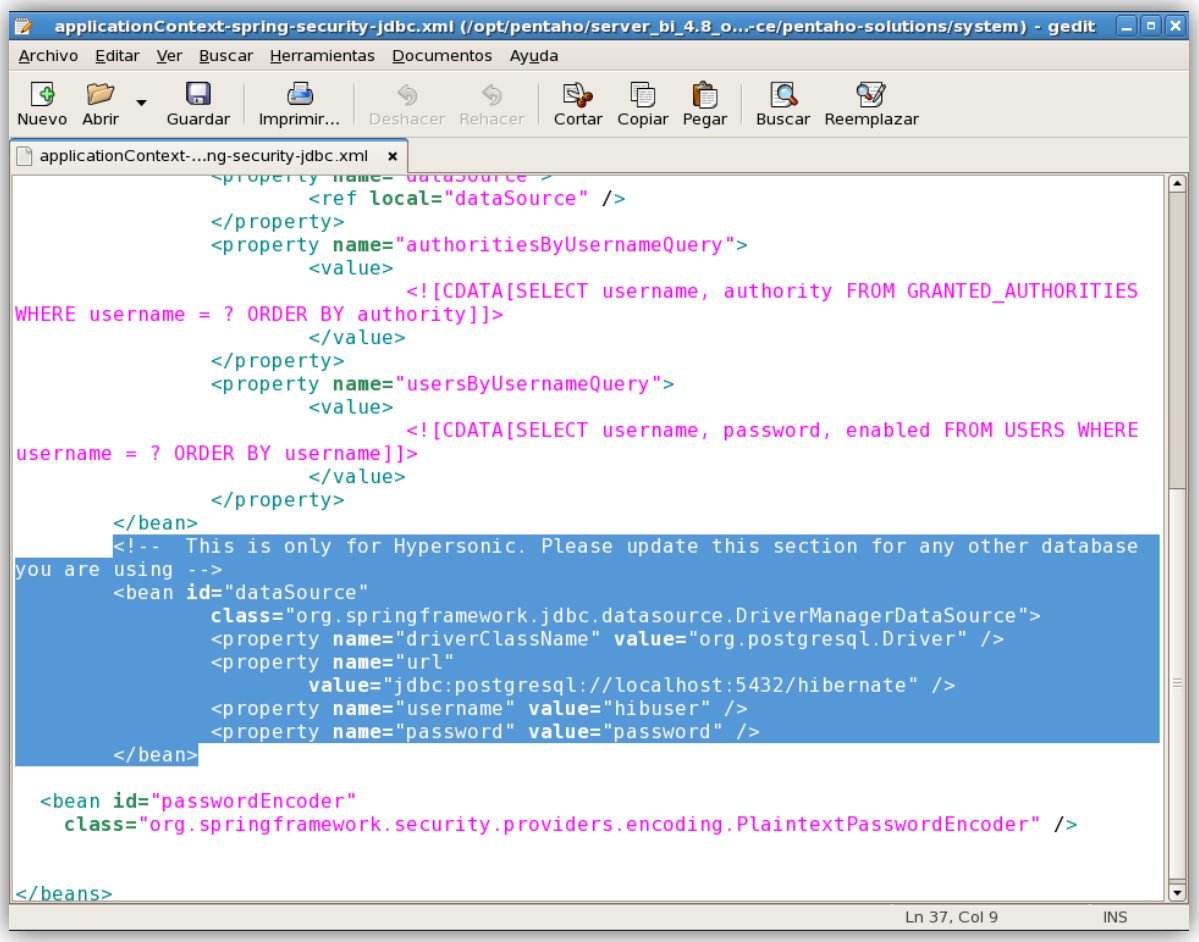
Se modificará el archivo de la siguiente forma:



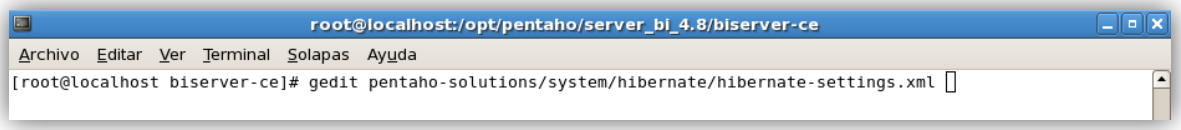
Se busca el archivo *applicationContext-spring-security-jdbc.xml* que se encuentra en el directorio *<path_biserver>/pentaho-solutions/system/* y se edita.



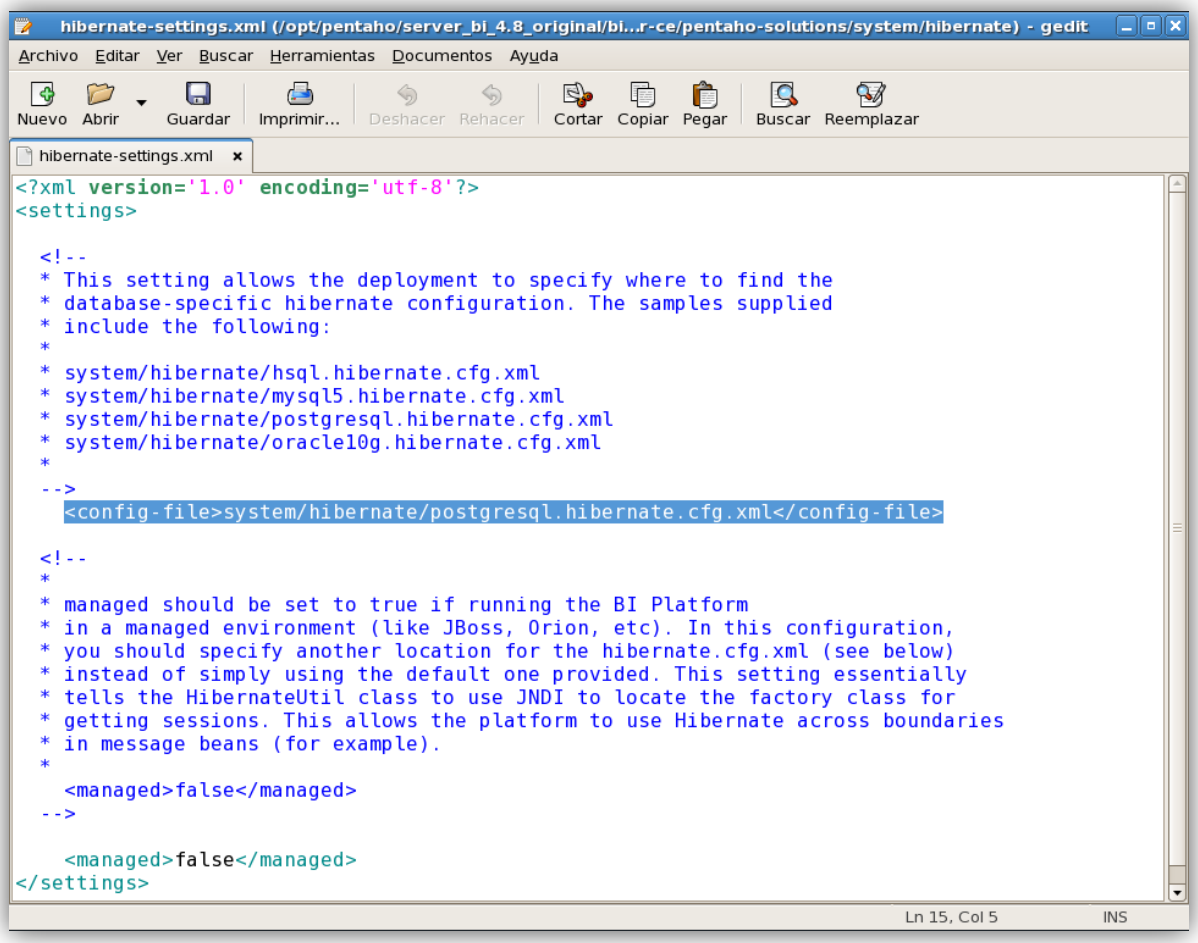
Modificaremos el archivo de la siguiente forma:



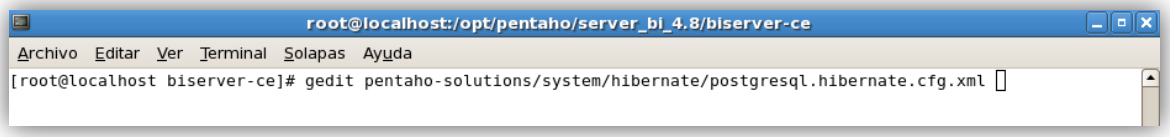
En el directorio *<path_biserver>/pentaho-solutions/system/hibernate/* se encuentra el archivo *hibernate-settings.xml*.



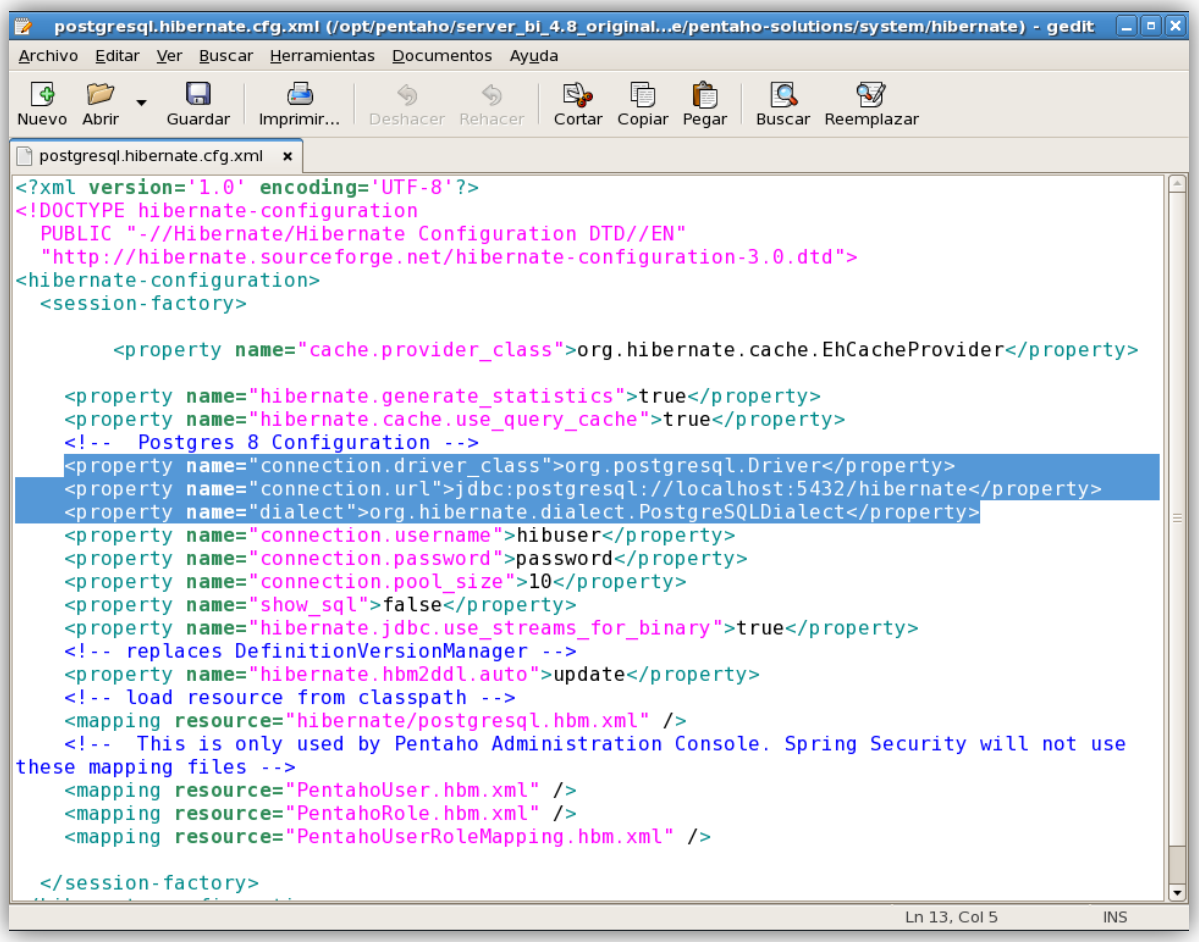
Modificaremos el archivo de la siguiente forma:



Y en el mismo directorio anterior se encuentra el archivo *postgresql.hibernate.cfg.xml***.**

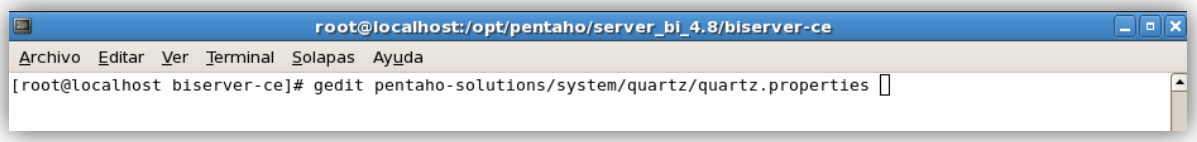


Se modifica el archivo de la siguiente forma:

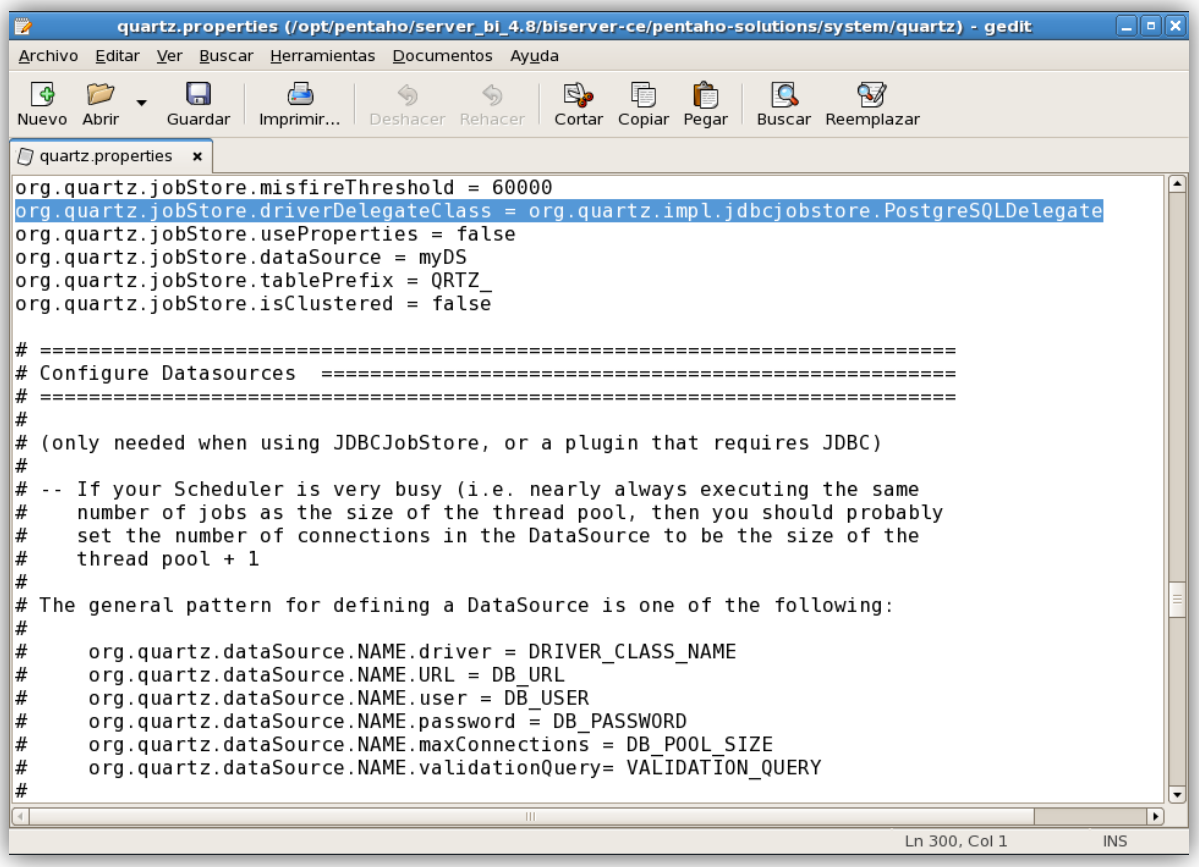


Configuración de Hibernate y Quartz

Hibernate[16] y Quartz[17] se necesita específicamente utilizar las bases de datos "hibernate" y "quartz". Por lo que hay que modificar el archivo *quartz.properties* en el directorio *<path_biserver>/pentaho-solutions/system/hibernate/*.



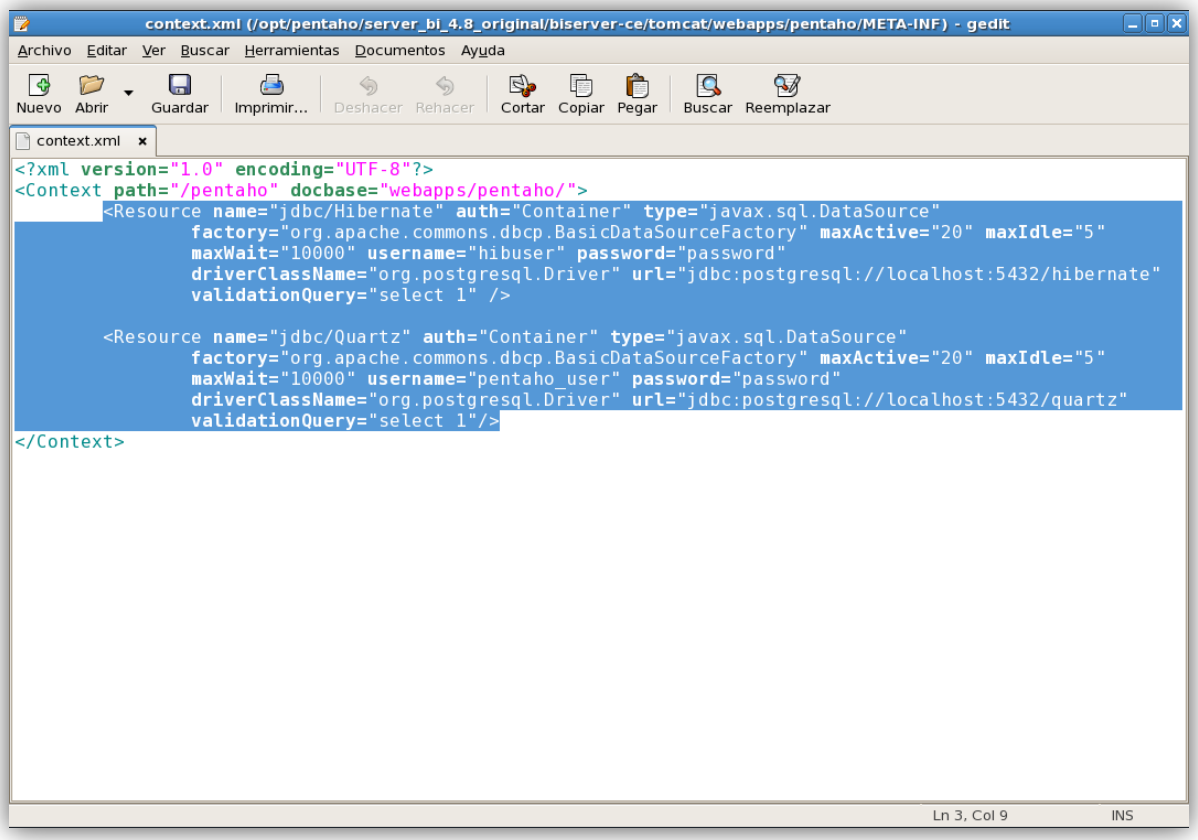
En la siguiente imagen se mostrara el código editado.



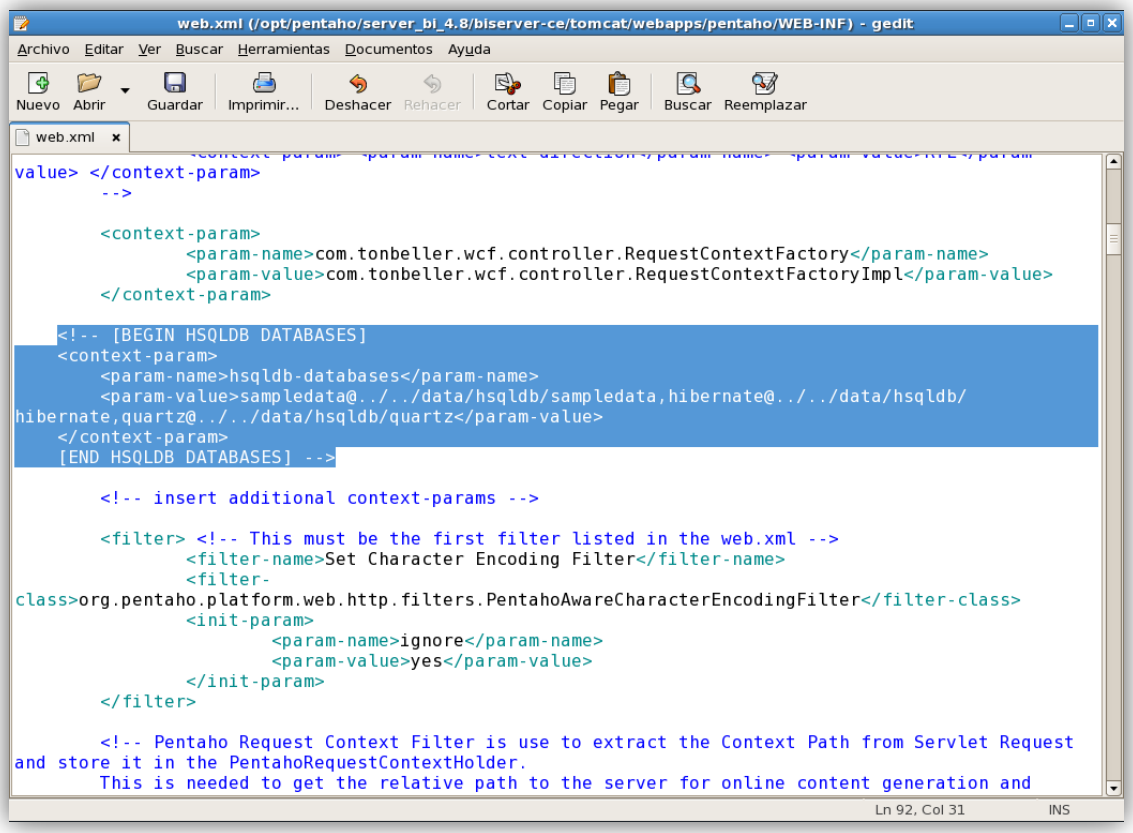
[16] Hibernate es un framework de Software Libre que permite hacer el mapeo objeto-relacional.
[17] Quartz es un framework de Software Libre que permite la planeación y gestión de tareas.

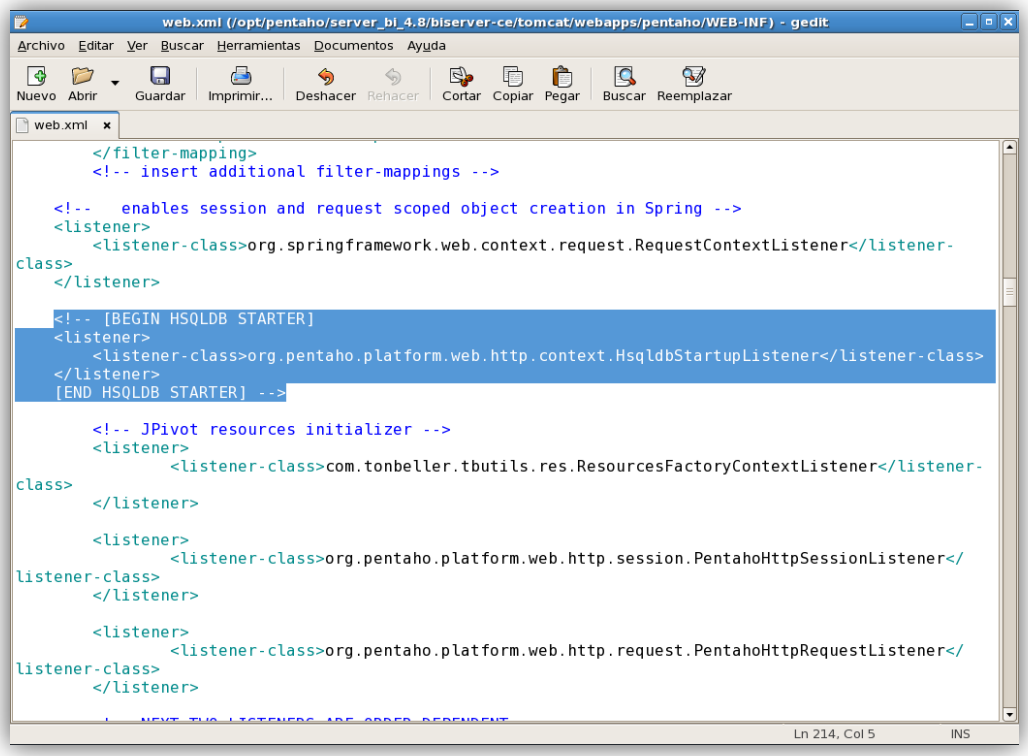
Después se edita el archivo *context*.xml ubicado en el directorio *<path_biserver>/tomcat/webapps/pentaho/META-INF/*

Se modifica el archivo de la siguiente forma:



También se edita el archivo *web.xml* debido a que en él está el código para activar el servidor HSQL que viene por defecto. Este archivo está ubicado en el directorio *<path_biserver>/tomcat/webapps/pentaho/WEB-INF/web.xml*

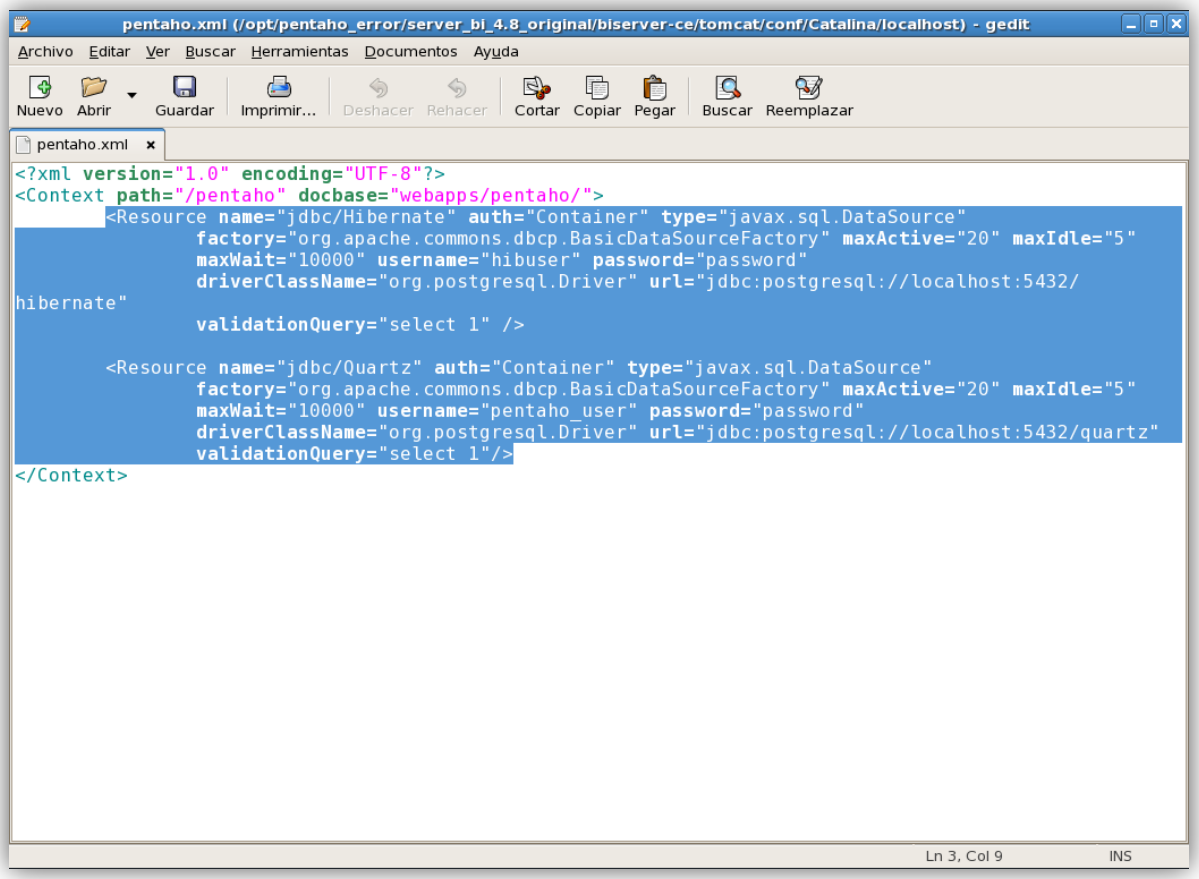




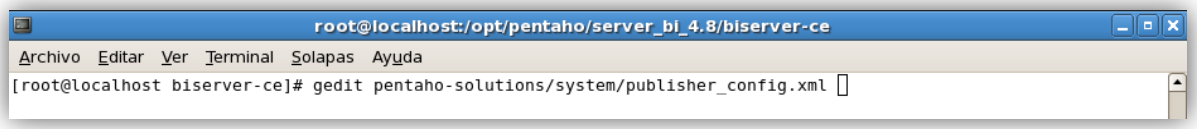
El directorio *<path_biserver>/tomcat/conf/Catalina/localhost/* se crea automáticamente una vez que se inicia Pentaho BI Server por primera vez y ahí aparece el archivo *pentaho.xml*, que debemos editar. Por lo tanto, los pasos serían los siguientes:

Verificar si existe el directorio *<path_biserver>/tomcat/conf/Catalina/localhost/*.

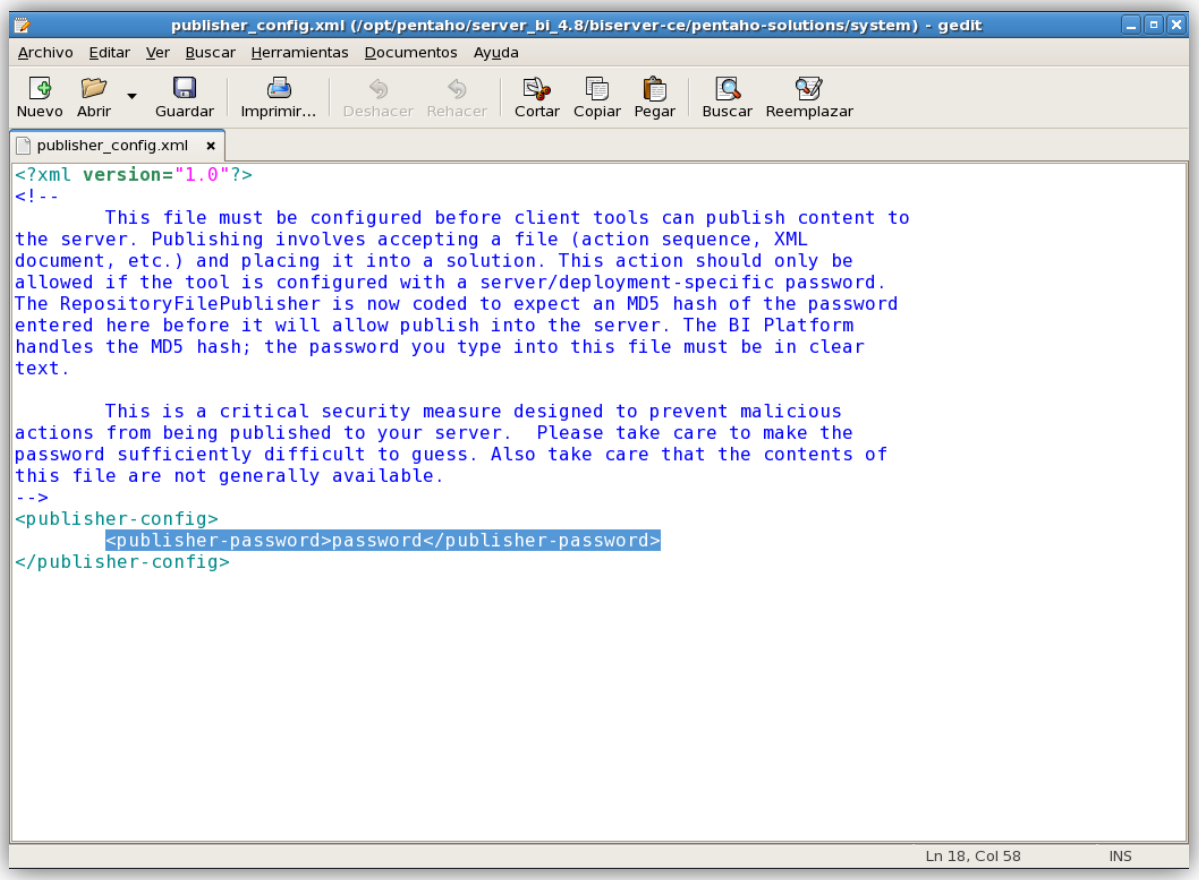
En caso de que exista, se edita el archivo *pentaho.xml* en dicho directorio, haciendo que quede igual al archivo *context.xml* que se configuró anteriormente.



Finalmente se edita el archivo *publisher_config.xml* ubicado en el directorio *<path_biserver>/pentaho-solutions/system/*, este archivo tiene por objetivo modificar el password que nos permite publicar los Cubos de Datos en el Pentaho BI Server 4.8 desde Schema Workbench.



El password que designamos para esto, es “password”.



Configurar la Consola de Administración

Para configurar la Consola de Administración con PostgreSQL es necesario copiar el driver JDBC para PostgreSQL 9.1. Se busca el driver en */opt/pentaho/server_bi_4.8/biserver-ce/tomcat/lib/* y luego se copia en */opt/pentaho/server_bi_4.8/administration-console/jdbc/.*

Probar la instalación

El servidor de Pentaho es una aplicación web que corre en el servidor Apache-Tomcat. Para iniciar el servidor Apache-Tomcat hay que ejecutar el script *start-pentaho.sh* ubicado en el directorio */opt/pentaho/server_bi_4.8/biserver-ce/*.

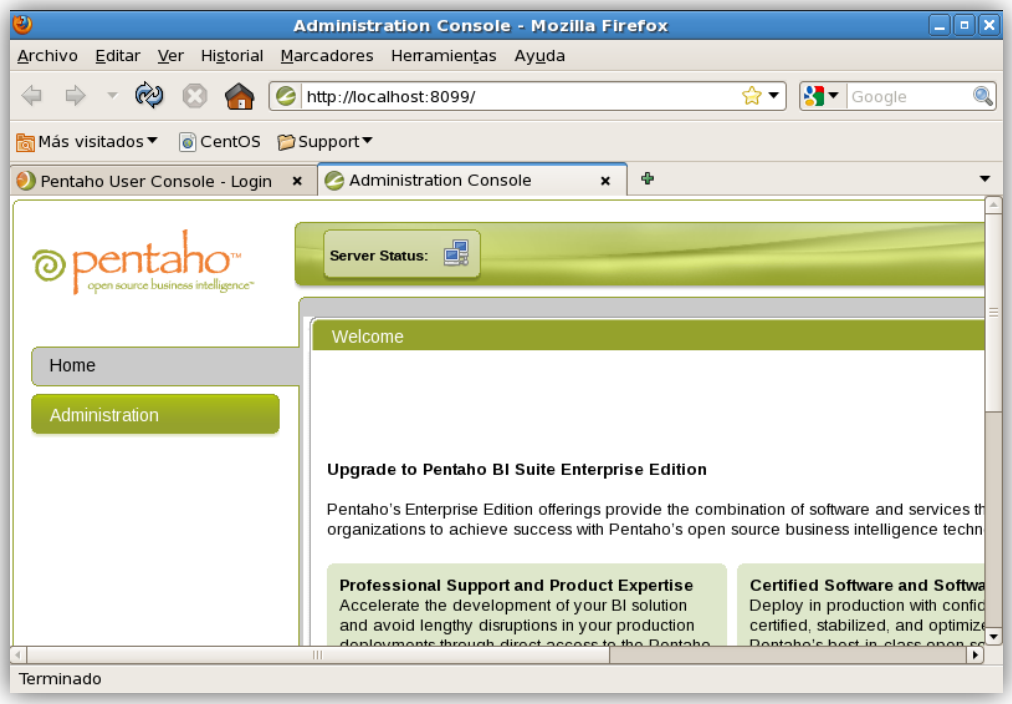
```
[root@localhost server_bi_4.8]# ls
administration-console  biserver-ce
[root@localhost server_bi_4.8]# cd biserver-ce/
[root@localhost biserver-ce]# ./start-pentaho.sh
/opt/pentaho/server_bi_4.8/biserver-ce
/opt/pentaho/server_bi_4.8/biserver-ce
DEBUG: Using JAVA_HOME
DEBUG: _PENTAHO_JAVA_HOME=/usr/java/latest
DEBUG: _PENTAHO_JAVA=/usr/java/latest/bin/java
------------------------------------------------------------------------------
------------
The Pentaho BI Platform now contains a version checker that will notify you
when newer versions of the software are available. The version checker is enable
d by default.
For information on what the version checker does, why it is beneficial, and how
it works see:
http://wiki.pentaho.com/display/ServerDoc2x/Version+Checker
Press Enter to continue, or type cancel or Ctrl-C to prevent the server from sta
rting.
You will only be prompted once with this question.
------------------------------------------------------------------------------
------------
[OK]:
Using CATALINA_BASE:   /opt/pentaho/server_bi_4.8/biserver-ce/tomcat
Using CATALINA_HOME:   /opt/pentaho/server_bi_4.8/biserver-ce/tomcat
Using CATALINA_TMPDIR: /opt/pentaho/server_bi_4.8/biserver-ce/tomcat/temp
Using JRE_HOME:        /usr/java/latest/jre
Using CLASSPATH:       /opt/pentaho/server_bi_4.8/biserver-ce/tomcat/bin/bootstr
ap.jar
[root@localhost biserver-ce]#
```

Posteriormente se arranca la consola de administración de Pentaho, para ello hay que colocarse en el directorio */opt/pentaho/server_bi_4.8/a*dministration-console/ y ejecutar el script *start-pac-sh*.

```
[root@localhost server_bi_4.8]# ls
administration-console  biserver-ce
[root@localhost server_bi_4.8]# cd administration-console/
[root@localhost administration-console]# ./start-pac.sh
/opt/pentaho/server_bi_4.8/administration-console
/opt/pentaho/server_bi_4.8/administration-console
DEBUG: Using JAVA_HOME
DEBUG: _PENTAHO_JAVA_HOME=/usr/java/latest
DEBUG: _PENTAHO_JAVA=/usr/java/latest/bin/java
06:31:42,874 INFO  [JettyServer] Console is starting
06:31:43,082 INFO  [/] org.pentaho.pac.server.BrowserLocaleServlet-26613447: init
06:31:43,087 INFO  [/] org.pentaho.pac.server.DefaultConsoleServlet-21171036: init
06:31:43,089 INFO  [/] org.pentaho.pac.server.PacServiceImpl-21061094: init
06:31:43,089 INFO  [/] org.pentaho.pac.server.SchedulerServiceImpl-4115088: init
06:31:43,089 INFO  [/] org.pentaho.pac.server.SolutionRepositoryServiceImpl-26715004: in
it
06:31:43,089 INFO  [/] org.pentaho.pac.server.SubscriptionServiceImpl-24529889: init
06:31:43,089 INFO  [/] org.pentaho.pac.server.common.HibernateConfigurationServiceImpl-1
2067688: init
06:31:43,089 INFO  [/] org.pentaho.pac.server.common.JdbcDriverDiscoveryServiceImpl-2625
0401: init
06:31:43,130 INFO  [JettyServer] Console is now started. It can be accessed using http:/
/localhost.localdomain:8099 or http://127.0.0.1:8099
```

Se abre un navegador y se coloca la dirección http://localhost:8080/pentaho para poder ingresar a la aplicación de Pentaho BI Server 4.8 Community Edition.

Se abre otra pestaña en el navegador y se coloca la siguiente dirección http://localhost:8099. Para ingresar a la consola de administración de Pentaho BI Server 4.8 Community Edition hay que especificar el usuario y contraseña que por defecto son "*admin*" y "*password*". Una vez logueados correctamente se verá la siguiente pantalla:



Aspectos de seguridad

Si siguió los pasos anteriores, el BI Server 4.8 y la Consola de Administración deberían poder usarse. Sin embargo, para un entorno de producción, recomendamos cambiar todas las contraseñas por defecto y atender a la seguridad de las distintas bases de datos. Además recomendamos considerar los puntos mencionados en el link http://wiki.pentaho.com/display/ServerDoc2x/Security+Configuration+Checklist de la wiki de Pentaho.

Anexo II. Manual de instalación de complementos (Saiku Analytics)

Para poder utilizar las vistas de Saiku Analytics predefinidas, los tableros y los reportes, es necesario instalar los componentes visuales. A diferencia de la versión anterior, los mismos se instalan desde la consola de usuario con un rol administrador.

Instalación desde Marketplace

Esta funcionalidad permite instalar distintos plugins de Pentaho BI Server 4.8. Requiere que el servidor tenga acceso a internet para listar y descargar los plugins.

Desde la consola de usuario hacer clic en el botón  , o Herramientas -> Marketplace. Se abrirá una solapa como se muestra en la imagen:

Instalar y dejar actualizado por lo menos:

- Community Dashboard Editor (CDE)
- Community Data Access (CDA)
- Community Dashboards Framework (CDF)
- Saiku Analytics

Anexo III. Manual de instalación de Pentaho Data Integration 4.4 Community Edition

Este manual tiene por objetivo guiar al usuario para la instalación de **Pentaho Data Integration 4.4 Community Edition**, más conocido como **kettle**, sobre la plataforma **Windows 7** (previamente instalado en la Máquina Real). La instalación comienza descargando el paquete Pentaho Data Integration 4.4 desde la siguiente URL:

http://sourceforge.net/projects/pentaho/files/Data%20Integration/4.4.0-stable/pdi-ce-4.4.0-stable.tar.gz/download

Requerimientos mínimos

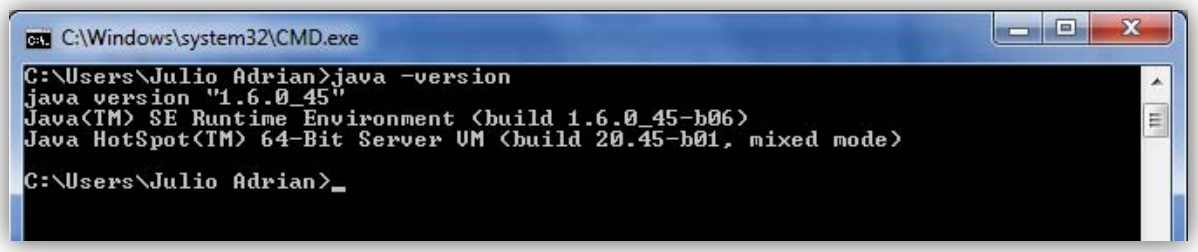
Para realizar una instalación básica de Pentaho Data Integration 4.4 recomendamos contar con al menos el siguiente hardware y software especificado:

- Hardware
  - ✓ RAM: 1 GB o más.
  - ✓ Disco Duro: 1 GB o más.
  - ✓ Procesador: Doble núcleo o superior.

- Software
  - ✓ Sistemas Operativos soportados: Windows, Linux o Mac OS X.
  - ✓ Pentaho: Data Integration 4.4 Community Edition.
  - ✓ Java: JDK (Java Development Kit) version 1.6.X.

Configuración de Java

Antes de poder iniciar con la instalación y configuración del servidor de Pentaho hay que verificar que la JVM (Java Virtual Machine) esté instalada y que las variables de entorno estén configuradas correctamente.



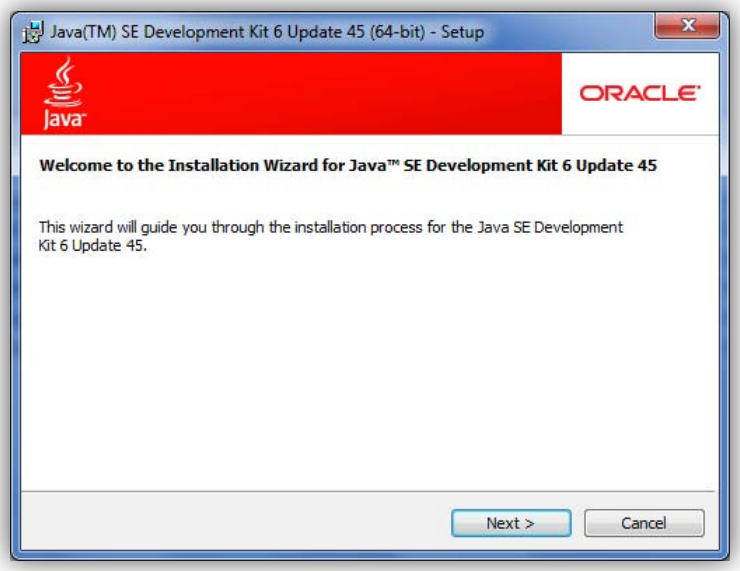
Si no le aparece un mensaje parecido al de la imagen anterior, debe actualizar a la versión oficial, en este manual se trabajara con la versión de **Java 1.6.u45**.

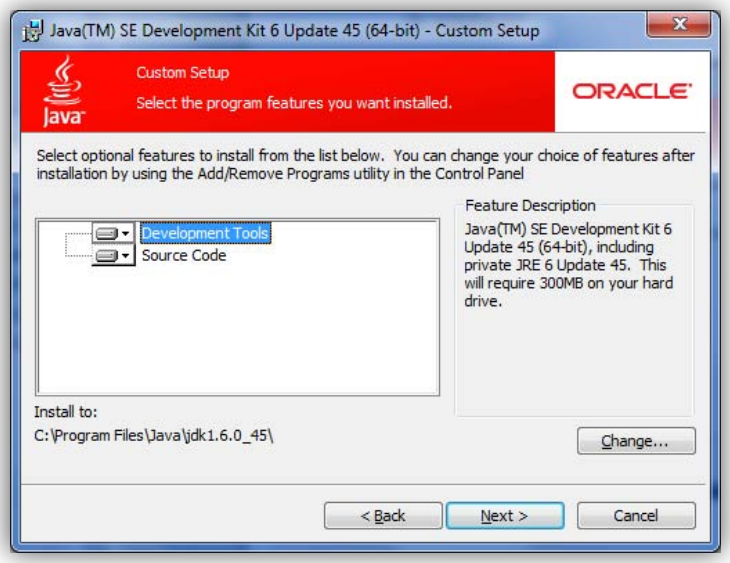
Actualización de Java

Primero hay que descargar el paquete de instalación del jdk de acuerdo a la arquitectura que tenga nuestro S.O., en nuestro caso es de 64 bits entonces se descarga el x64 desde la siguiente URL:

JDK 1.6.u45: http://www.oracle.com/technetwork/java/javasebusiness/downloads/java-archive-downloads-javase6-419409.html#jdk-6u45-oth-JPR

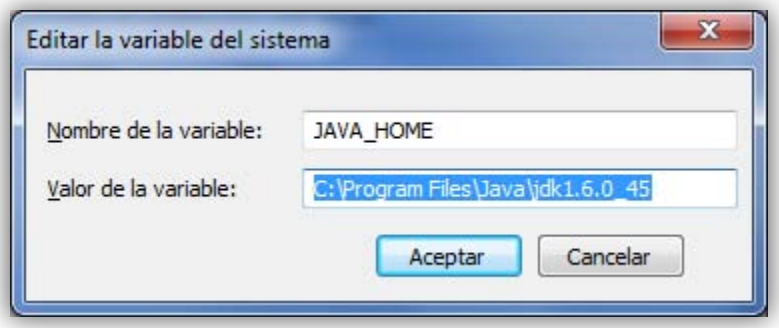
Se ejecuta el paquete de instalación y se da clic en Next, luego se seleccionan los paquetes de instalación y se vuelve a dar clic en Next.





Ya terminada la instalación se agregan las nuevas variables de entorno. Se abre la ventana *Propiedades del sistema* y se da clic en la pestaña *Opciones avanzadas*. Luego se da clic en *Variables de entorno…*

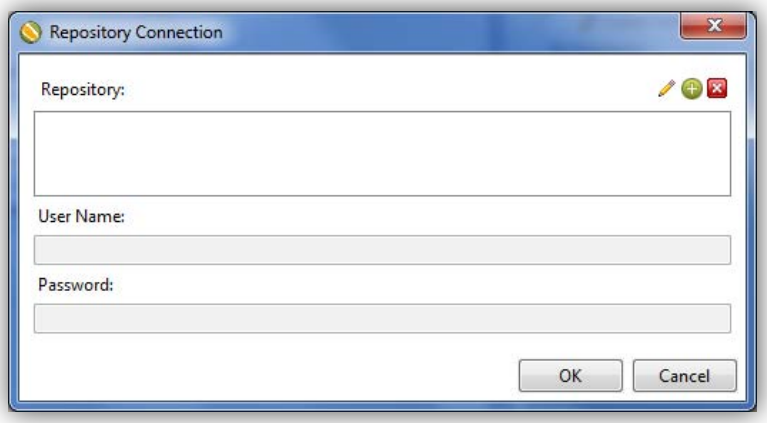
Se agregan dos variables de sistema; *JAVA_HOME* y *Path*, como se muestra en las siguientes imágenes.



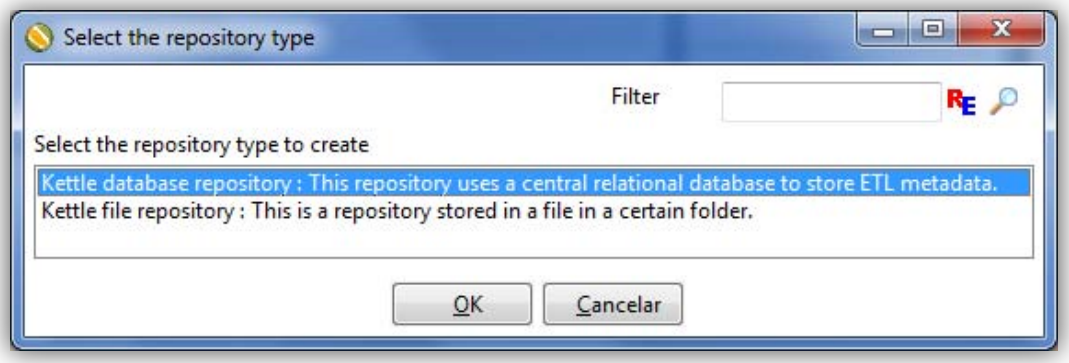
El agregar estas variables de entorno nos permite tener un correcto funcionamiento del software de Pentaho.

Instalación de Pentaho Data Integration 4.4 Community Edition

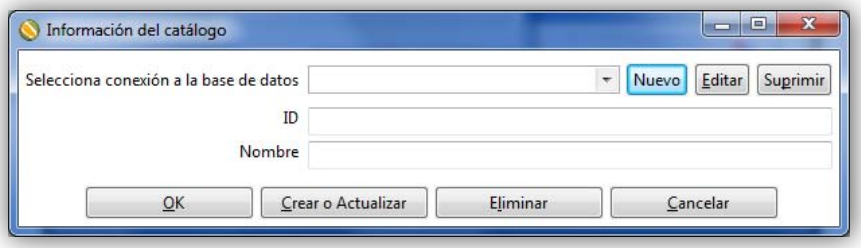
Ya realizado lo anterior, se descomprime el paquete *pdi-ce-4.4.0-stable.tar.gz* en cualquier directorio de nuestra PC. Finalmente se ejecuta el archivo *Spoon.bat* y se espera a que se muestre la siguiente ventana.



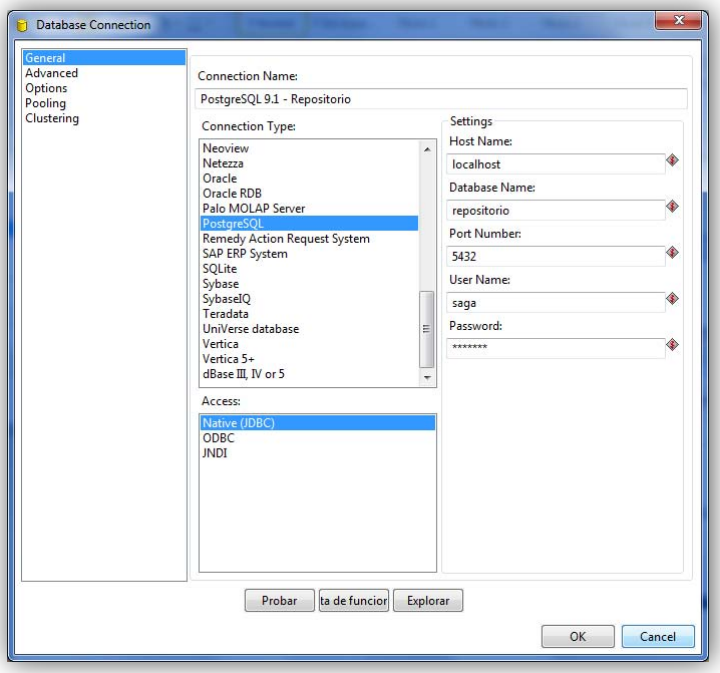
Posteriormente se crea el *Repositorio*, para ello damos clic en  para añadir uno nuevo de tipo *Kettle database repository*.



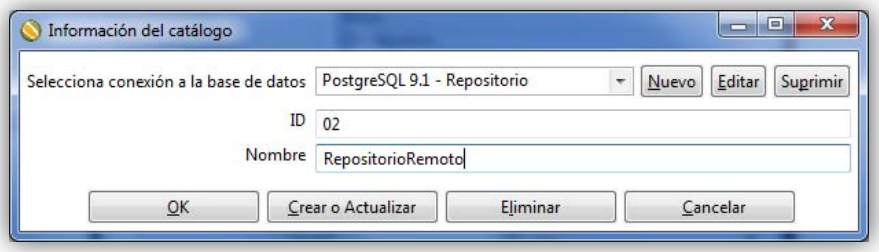
Se da clic en *Ok* y nos muestra una ventana para la configuración de nuestro repositorio.



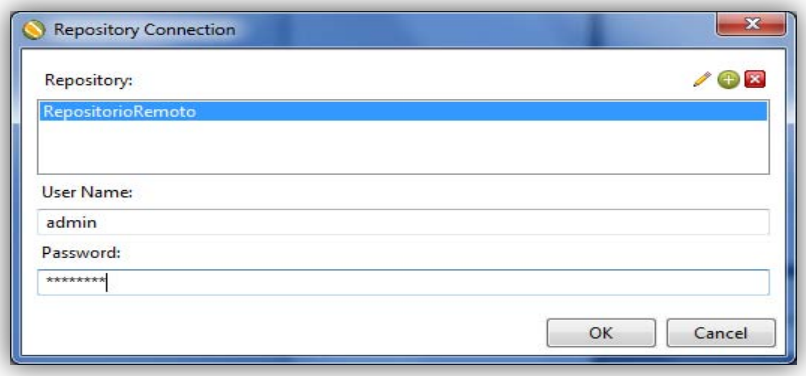
En la opción de *Selecciona conexión a la base de datos* se escribe lo siguiente:



Y en las opciones *ID* y *Nombre* se escribe lo siguiente:



Una vez asignadas las opciones, hay que dar clic en *Crear* y se termina con la configuración de nuestro *Repositorio* el cual almacenará todas las transformaciones y trabajos del ETL. Finalmente se accede a **kettle,** usando como User Name y Password a “admin” y “admin” respectivamente, ya que es el usuario por defecto.



Anexo IV. Manual de instalación de Pentaho Schema Workbench 3.5 Community Edition

Este manual tiene por objetivo guiar al usuario para la instalación de **Pentaho Schema Workbench 3.5 Community Edition**, sobre la plataforma **Windows 7** (previamente instalado en la Máquina Real). La instalación comienza con la descarga del paquete Pentaho Data Integration 4.4 desde la siguiente URL:

http://sourceforge.net/projects/mondrian/files/schema%20workbench/3.5.0-stable/psw-ce-3.5.0.tar.gz/download

Requerimientos mínimos

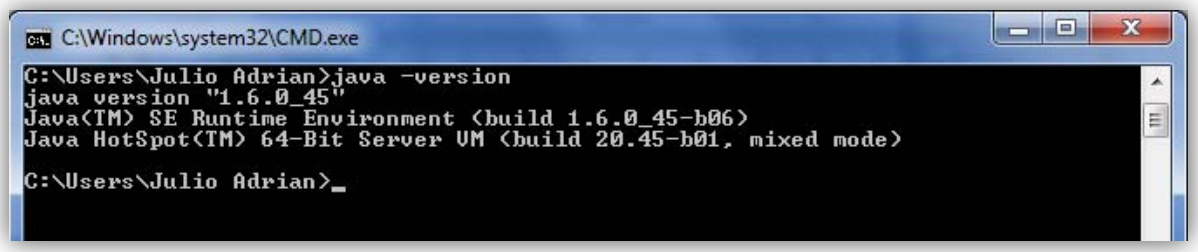
Para realizar una instalación básica de Pentaho Schema Workbench 3.5 recomendamos contar con al menos el siguiente hardware y software especificado:

- Hardware
  - ✓ RAM: 1 GB o más.
  - ✓ Disco Duro: 1 GB o más.
  - ✓ Procesador: Doble núcleo o superior.

- Software
  - ✓ Sistemas Operativos soportados: Windows, Linux o Mac OS X.
  - ✓ Pentaho: Schema Workbench 3.5 Community Edition.
  - ✓ Java: JDK (Java Development Kit) version 1.6.X.

Configuración de Java

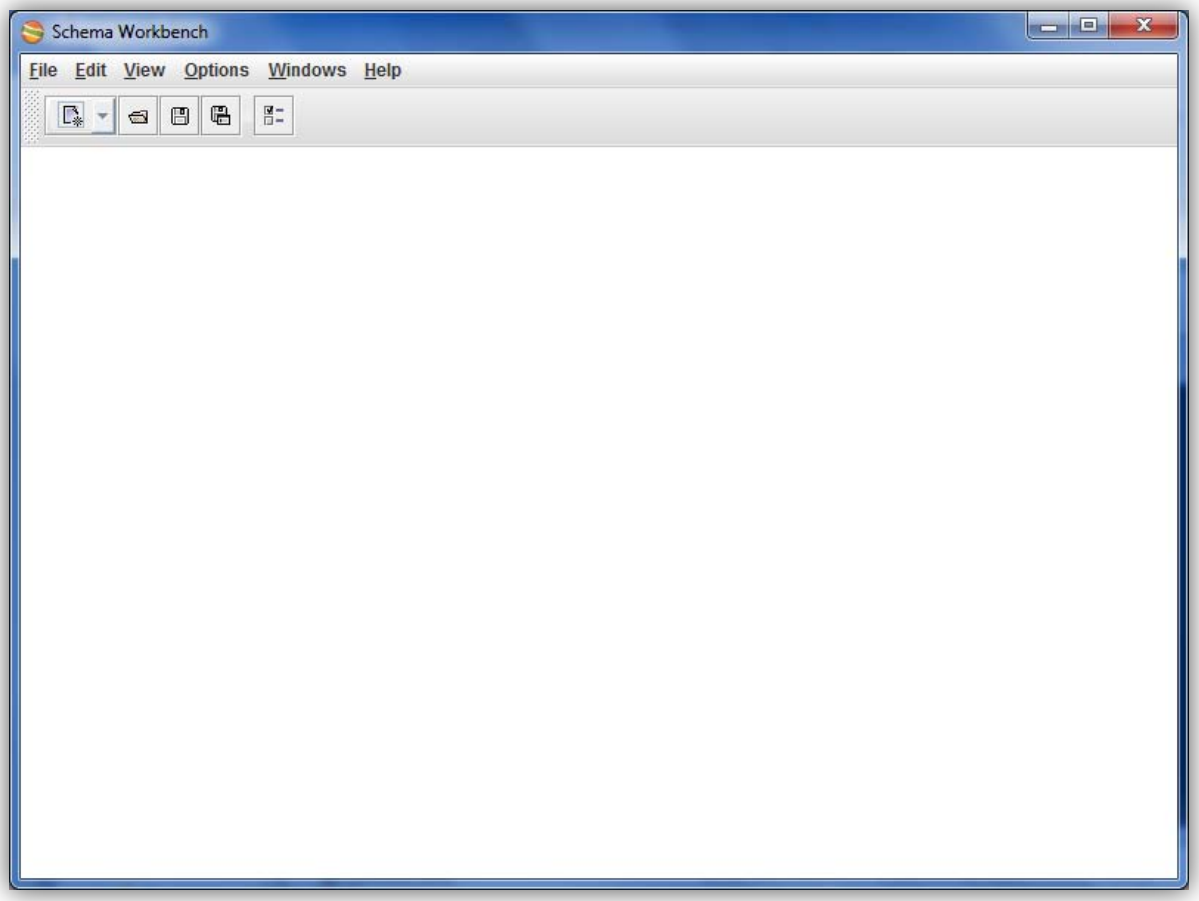
Antes de poder iniciar con la instalación y configuración del servidor de Pentaho hay que verificar que la JVM (Java Virtual Machine) esté instalada y que las variables de entorno estén configuradas correctamente.



Si no le aparece un mensaje parecido al de la imagen anterior, debe actualizar a la versión oficial, en este manual se trabajará con la versión de **Java 1.6.u45**.

Instalación de Pentaho Schema Workbench 3.5 Community Edition

Ya realizado lo anterior, se descomprime el paquete *psw-ce-3.5.0.tar.gz* en cualquier directorio de nuestra PC. Finalmente se ejecuta el archivo *workbench.bat* y se espera a que se muestre la siguiente ventana.



En esta ventana ya podemos comenzar a crear nuestro esquema (mapear de las tablas físicas a las dimensiones y medidas que sean de nuestro interés).

Anexo V. Documento de Alcance

Título de proyecto: Análisis, diseño y desarrollo de un Data Warehouse al personal académico del sistema de informes académicos de humanidades para el IISUE.

Descripción: Crear un Data Warehouse que sirva como base de un sistema informático gerencial en el análisis y consulta de información (reportes y gráficas) brindando soporte dentro de la toma de decisiones con base a la planeación y administración de recursos en el IISUE.

Justificación: Dentro del Instituto de Investigaciones sobre la Universidad y la Educación (IISUE) siempre se ha tenido la tarea de elaborar reportes que muestren una visión detallada sobre el desarrollo de actividades e investigaciones elaboradas por parte del Personal Académico, tal información suele ser necesaria para la toma de decisiones en la planeación y administración de recursos dentro del propio IISUE y además permite conservar un histórico estadístico sobre esa información.

Entregas clave: Documentos de diseño (DER y Scripts SQL) sobre el OLTP y los 3 ROLAP pertenecientes a cada Cubo OLAP.

Documentos de flujo del ETL (KTR's y JOB's).

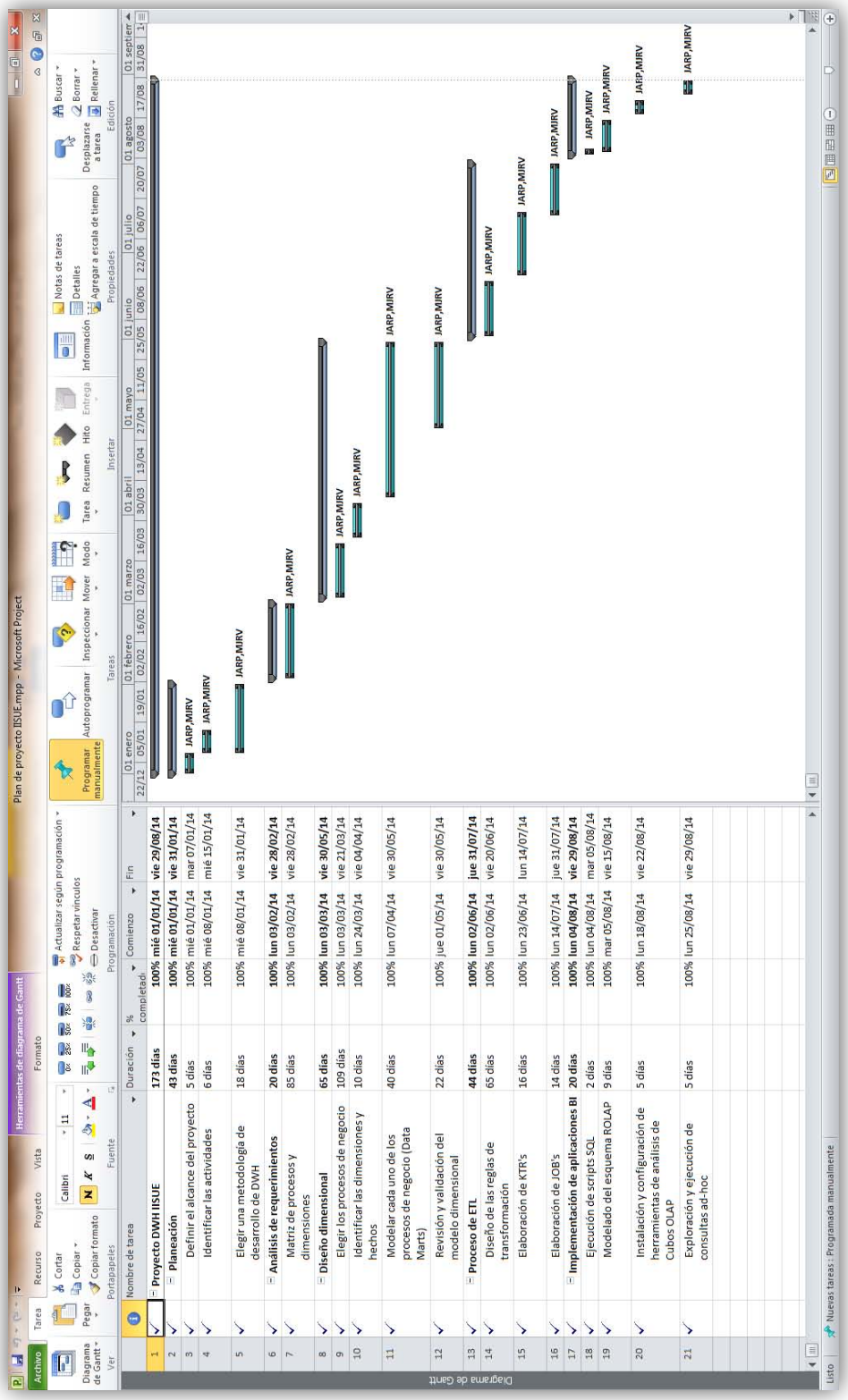
Guías de instalación y configuración del ambiente para la implementación de un DWH y guía de usuario.

Objetivos: Aplicar una metodología de desarrollo de Data Warehouse.

Satisfacer las necesidades de los usuarios en las consultas ad-hoc que den soporte a la toma de decisiones.

Implementar un ambiente de DWH piloto con información útil y de calidad por medio del ETL. La información será accesible a los usuarios para ayudar a la toma de decisiones con ayuda de las consultas ad-hoc.

Anexo VI. Plan de proyecto



Proceso de Negocio		Dimensiones					
		Tiempo	Lugar	Tipo de académico	Nivel académico	Tipo de producto/ servicio	Estimulos
Grupos Académicos	Se identifico como proceso de negocio debido a que en el se centra todo lo relacionados a los Academicos.	x	x	x	x		x
Actividades Docentes	Se identifico como proceso de negocio debido a que es un aspecto particular de los Academicos	x	x	x		x	
Proyectos	Se identifico como posible proceso de negoio ya que es la principal tarea que elaboran los Academicos	x	x	x		x	

Anexo VIII. Script SQL del DWH

```
-------------------------------------------------
-------------Script DW_IISUE_Postgresql-------------
-------------------------------------------------

DROP DATABASE IF EXISTS dw_iisue;
DROP TABLESPACE IF EXISTS dwspace;
DROP USER IF EXISTS saga;

-- Creación de Usuario con privilegio de Crear Bases de Datos
CREATE USER saga WITH PASSWORD 'g3m1n1s' SUPERUSER CREATEDB;

-- Acceder como usuario "saga"
\c postgres saga

-- Creación de un Tablespace que almacene el Data Warehouse
CREATE TABLESPACE dwspace OWNER saga LOCATION '/mnt/DATA/dws';

-- Creación de la BD
CREATE DATABASE dw_iisue OWNER saga ENCODING 'UTF8' TEMPLATE template1
TABLESPACE dwspace;
GRANT ALL ON DATABASE dw_iisue to saga;

-- ----------------------------
-- Tabla - academicos
-- ----------------------------
DROP TABLE IF EXISTS "public"."academicos";
CREATE TABLE "public"."academicos" (
"id_academico" int4 NOT NULL,
"num_trabajador" varchar(10) COLLATE "default" NOT NULL,
"nombre" varchar(255) COLLATE "default" NOT NULL,
"timestamp" timestamp(6) NOT NULL,
PRIMARY KEY ("id_academico")
)
TABLESPACE dwspace;


-- ----------------------------
-- Tabla - anios
-- ----------------------------
DROP TABLE IF EXISTS "public"."anios";
CREATE TABLE "public"."anios" (
"anio" int2 NOT NULL,
"timestamp" timestamp(6) DEFAULT now() NOT NULL,
PRIMARY KEY (anio)
)
TABLESPACE dwspace;
```

```
-- ----------------------------
-- Tabla - entidades
-- ----------------------------
DROP TABLE IF EXISTS "public"."entidades";
CREATE TABLE "public"."entidades" (
"id_entidad" int4 NOT NULL,
"tipo_entidad" varchar(40) COLLATE "default" NOT NULL,
"abr_tipo_entidad" varchar(4) COLLATE "default" NOT NULL,
"entidad" varchar(90) COLLATE "default" NOT NULL,
"abr_entidad" varchar(7) COLLATE "default" NOT NULL,
"facultad" char(1) COLLATE "default" NOT NULL,
"depto" varchar(255) COLLATE "default" NOT NULL,
"abr_depto" varchar(6) COLLATE "default" NOT NULL,
"timestamp" timestamp(6) NOT NULL,
PRIMARY KEY (id_entidad)
)
TABLESPACE dwspace;


-- ----------------------------
-- Tabla - grados_estudios
-- ----------------------------
DROP TABLE IF EXISTS "public"."grados_estudios";
CREATE TABLE "public"."grados_estudios" (
"id_grado_estudio" int4 NOT NULL,
"grado_estudio" varchar(15) COLLATE "default" NOT NULL,
"abr_grado_estudio" varchar(3) COLLATE "default" NOT NULL,
"timestamp" timestamp(6) NOT NULL,
PRIMARY KEY (id_grado_estudio)
)
TABLESPACE dwspace;


-- ----------------------------
-- Tabla - estimulos
-- ----------------------------
DROP TABLE IF EXISTS "public"."estimulos";
CREATE TABLE "public"."estimulos" (
"id_estimulo" int4 NOT NULL,
"estimulo" varchar(12) COLLATE "default" NOT NULL,
"sni" char(1) COLLATE "default" NOT NULL,
"pride" char(1) COLLATE "default" NOT NULL,
"paipa" char(1) COLLATE "default" NOT NULL,
"fomdoc" char(1) COLLATE "default" NOT NULL,
"timestamp" timestamp(6) NOT NULL,
PRIMARY KEY (id_estimulo)
)
TABLESPACE dwspace;
```

```
-- ----------------------------
-- Tabla - nacionalidades
-- ----------------------------
DROP TABLE IF EXISTS "public"."nacionalidades";
CREATE TABLE "public"."nacionalidades" (
"id_nacionalidad" int4 NOT NULL,
"nacionalidad" varchar(100) COLLATE "default" NOT NULL,
"abr_nacionalidad" varchar(2) COLLATE "default" NOT NULL,
"pais" varchar(50) COLLATE "default" NOT NULL,
"timestamp" timestamp(6) NOT NULL,
PRIMARY KEY (id_nacionalidad)
)
TABLESPACE dwspace;


-- ----------------------------
-- Tabla - tipos_academicos
-- ----------------------------
DROP TABLE IF EXISTS "public"."tipos_academicos";
CREATE TABLE "public"."tipos_academicos" (
"id_tipo_academico" int4 NOT NULL,
"clase" varchar(30) COLLATE "default" NOT NULL,
"categoria" varchar(20) COLLATE "default" NOT NULL,
"nivel" varchar(9) COLLATE "default" NOT NULL,
"genero" varchar(10) COLLATE "default" NOT NULL,
"timestamp" timestamp(6) NOT NULL,
PRIMARY KEY (id_tipo_academico)
)
TABLESPACE dwspace;


-- ----------------------------
-- Tabla - tipos_proyectos
-- ----------------------------
DROP TABLE IF EXISTS "public"."tipos_proyectos";
CREATE TABLE "public"."tipos_proyectos" (
"id_tipo_proyecto" int4 NOT NULL,
"tipo_proyecto" varchar(23) COLLATE "default" NOT NULL,
"abr_tipo_proyecto" varchar(7) COLLATE "default" NOT NULL,
"timestamp" timestamp(6) NOT NULL,
PRIMARY KEY (id_tipo_proyecto)
)
TABLESPACE dwspace;
```

```
-- ----------------------------
-- Tabla - estatus_proyectos
-- ----------------------------
DROP TABLE IF EXISTS "public"."estatus_proyectos";
CREATE TABLE "public"."estatus_proyectos" (
"id_estatus_proyecto" int4 NOT NULL,
"estatus_proyecto" varchar(11) COLLATE "default" NOT NULL,
"abr_estatus_proyecto" varchar(3) COLLATE "default" NOT NULL,
"timestamp" timestamp(6) NOT NULL,
PRIMARY KEY (id_estatus_proyecto)
)
TABLESPACE dwspace;



DROP TABLE IF EXISTS "public"."tipos_participacion";
CREATE TABLE "public"."tipos_participacion" (
"id_tipo_participacion" int4 NOT NULL,
"tipo_participacion" varchar(17) COLLATE "default" NOT NULL,
"abr_tipo_participacion" varchar(3) COLLATE "default" NOT NULL,
"timestamp" timestamp(6) NOT NULL,
PRIMARY KEY (id_tipo_participacion)
)
TABLESPACE dwspace;


-- ----------------------------
-- Tabla - grupos_academicos
-- ----------------------------
DROP TABLE IF EXISTS "public"."grupos_academicos";
CREATE TABLE "public"."grupos_academicos" (
"anio" int2 NOT NULL,
"id_tipo_academico" int4 NOT NULL,
"id_grado_estudio" int4 NOT NULL,
"id_entidad" int4 NOT NULL,
"id_nacionalidad" int4 NOT NULL,
"id_estimulo_sni" int4 NOT NULL,
"id_estimulo_pride" int4 NOT NULL,
"id_estimulo_fomdoc" int4 NOT NULL,
"id_estimulo_paipa" int4 NOT NULL,
"total_academicos" int4 NOT NULL,
PRIMARY KEY ("anio", "id_tipo_academico", "id_grado_estudio",
"id_entidad", "id_nacionalidad", "id_estimulo_sni", "id_estimulo_pride",
"id_estimulo_fomdoc", "id_estimulo_paipa")
)
TABLESPACE dwspace;
```

```
-- ----------------------------
-- Tabla - actividades_docentes
-- ----------------------------
DROP TABLE IF EXISTS "public"."actividades_docentes";
CREATE TABLE "public"."actividades_docentes" (
"anio" int2 NOT NULL,
"id_academico" int4 NOT NULL,
"id_tipo_academico" int4 NOT NULL,
"id_grado_estudio" int4 NOT NULL,
"id_entidad" int4 NOT NULL,
"total_asignaturas" int4 NOT NULL,
"total_tesis" int4 NOT NULL,
"total_tesinas" int4 NOT NULL,
PRIMARY KEY ("anio", "id_academico", "id_tipo_academico",
"id_grado_estudio", "id_entidad")
)
TABLESPACE dwspace;


-- ----------------------------
-- Tabla - proyectos
-- ----------------------------
DROP TABLE IF EXISTS "public"."proyectos";
CREATE TABLE "public"."proyectos" (
"anio" int2 NOT NULL,
"id_academico" int4 NOT NULL,
"id_tipo_academico" int4 NOT NULL,
"id_tipo_proyecto" int4 NOT NULL,
"id_estatus_proyecto" int4 NOT NULL,
"id_tipo_participacion" int4 NOT NULL,
"total_proyectos" int4 NOT NULL,
PRIMARY KEY ("anio", "id_academico", "id_tipo_academico",
"id_tipo_proyecto", "id_estatus_proyecto", "id_tipo_participacion")
)
TABLESPACE dwspace;


-- ----------------------------
-- Tabla - dw_iisue
-- ----------------------------
DROP TABLE IF EXISTS "public"."dw_iisue";
CREATE TABLE "public"."dw_iisue" (
"bd_access" varchar(50) COLLATE "default" NOT NULL,
"timestamp" timestamp(6) DEFAULT now() NOT NULL
)
TABLESPACE dwspace;
```

```
-- ----------------------------
-- FK's
-- ----------------------------
ALTER TABLE "public"."grupos_academicos"
ADD FOREIGN KEY ("id_tipo_academico") REFERENCES
"public"."tipos_academicos" ("id_tipo_academico") ON DELETE NO ACTION ON
UPDATE NO ACTION,
ADD FOREIGN KEY ("id_grado_estudio") REFERENCES
"public"."grados_estudios" ("id_grado_estudio") ON DELETE NO ACTION ON
UPDATE NO ACTION,
ADD FOREIGN KEY ("id_entidad") REFERENCES "public"."entidades"
("id_entidad") ON DELETE NO ACTION ON UPDATE NO ACTION,
ADD FOREIGN KEY ("id_nacionalidad") REFERENCES "public"."nacionalidades"
("id_nacionalidad") ON DELETE NO ACTION ON UPDATE NO ACTION,
ADD FOREIGN KEY ("id_estimulo_sni") REFERENCES "public"."estimulos"
("id_estimulo") ON DELETE NO ACTION ON UPDATE NO ACTION,
ADD FOREIGN KEY ("id_estimulo_pride") REFERENCES "public"."estimulos"
("id_estimulo") ON DELETE NO ACTION ON UPDATE NO ACTION,
ADD FOREIGN KEY ("id_estimulo_paipa") REFERENCES "public"."estimulos"
("id_estimulo") ON DELETE NO ACTION ON UPDATE NO ACTION,
ADD FOREIGN KEY ("id_estimulo_fomdoc") REFERENCES "public"."estimulos"
("id_estimulo") ON DELETE NO ACTION ON UPDATE NO ACTION;

ALTER TABLE "public"."actividades_docentes"
ADD FOREIGN KEY ("id_academico") REFERENCES
"public"."academicos" ("id_academico") ON DELETE NO ACTION ON UPDATE NO
ACTION,
"public"."tipos_academicos" ("id_tipo_academico") ON DELETE NO ACTION ON
UPDATE NO ACTION,
ADD FOREIGN KEY ("id_grado_estudio") REFERENCES
"public"."grados_estudios" ("id_grado_estudio") ON DELETE NO ACTION ON
UPDATE NO ACTION,
ADD FOREIGN KEY ("id_entidad") REFERENCES "public"."entidades"
("id_entidad") ON DELETE NO ACTION ON UPDATE NO ACTION;

ALTER TABLE "public"."proyectos"
ADD FOREIGN KEY ("id_academico") REFERENCES
"public"."academicos" ("id_academico") ON DELETE NO ACTION ON UPDATE NO
ACTION,
"public"."tipos_academicos" ("id_tipo_academico") ON DELETE NO ACTION ON
UPDATE NO ACTION,
ADD FOREIGN KEY ("id_tipo_proyecto") REFERENCES
"public"."tipos_proyectos" ("id_tipo_proyecto") ON DELETE NO ACTION ON
UPDATE NO ACTION,
ADD FOREIGN KEY ("id_estatus_proyecto") REFERENCES
"public"."estatus_proyecto" ("id_estatus_proyecto") ON DELETE NO ACTION
ON UPDATE NO ACTION,
ADD FOREIGN KEY ("id_tipo_participacion") REFERENCES
"public"."tipos_participacion" ("id_tipo_participacion") ON DELETE NO
ACTION ON UPDATE NO ACTION;
```