



RÍO NEGRO  
UNIVERSIDAD NACIONAL

# **Implementación de un Data Mart para la ayuda en la toma de decisiones de la Gerencia de Recursos Humanos**

**Trabajo Final de Carrera**

**Licenciatura en Sistemas**

Tesista: T.U.P. Traiman Schroh, Rossana Ayelén

Directora: Mg. Formia, Sonia Alejandra

Co-director: Ing. García Martínez, Nicolás

Viedma, Río Negro. Año 2021

# AGRADECIMIENTOS

En especial a mi mamá y papá, por el acompañamiento y apoyo incondicional en todo momento y sobre todo en el transcurso de mi carrera universitaria.

A la Mg. Sonia Formia por dirigir y guiar la elaboración de este trabajo. Por brindar su conocimiento e infundir ánimos para finalizar esta etapa.

A mis amigos por la ayuda, los consejos, por estar siempre en los buenos y malos momentos y por alegrarse por mis logros.

Y a todas aquellas personas que fueron parte de mi vida desde diferentes lugares y que me ayudaron a concluir con este proyecto, en especial a mis compañeros de trabajo por su aliento y motivación y a todos los profesores de la carrera Licenciatura en Sistemas que colaboraron en mi proceso de aprendizaje y formación.

## RESUMEN

En el presente trabajo se describe el desarrollo de una solución de Inteligencia de Negocios en un área de un Organismo del Estado Provincial, en particular de la Gerencia de Recursos Humanos de la Agencia de Recaudación Tributaria de la Provincia de Río Negro. Se propone para ello el diseño e implementación de un Data Mart y su integración al Data Warehouse existente en el Organismo.

Con este desarrollo se brinda al área un mayor grado de agilidad en la obtención de reportes para el análisis de los datos, permitiendo además visualizar y analizar datos estadísticos y su evolución. Toda esta información se dispone mediante un fácil acceso y de forma estructurada para ayudar en la toma de decisiones de la Gerencia.

En este proyecto se explican detalladamente los conceptos necesarios para una mejor comprensión del mismo y las herramientas utilizadas. También se describen los pasos de la metodología elegida y la forma en la que se siguieron dichos lineamientos para llegar al resultado final, teniendo en cuenta lo implementado previamente en la Organización.

Palabras claves: Inteligencia de negocios, Data Warehouse, Data Mart

# ÍNDICE

1.	INTRODUCCIÓN.....	6
1.1.	Contexto.....	6
1.2.	Contenido del TFC .....	7
2.	ESTADO DE LA CUESTIÓN .....	9
2.1	¿Qué es Inteligencia de negocios (BI)? .....	9
2.2.	Importancia de BI para la organización .....	10
2.3.	Data Warehouse (DW).....	11
2.3.1.	Componentes de un Data Warehouse.....	12
2.4.	Data Mart (DM) .....	14
2.5.	Conceptos .....	15
2.5.1.	Medidas .....	15
2.5.2.	Dimensiones .....	15
2.5.3.	Cubo .....	16
2.5.4.	Modelo Estrella .....	17
2.6.	Metodologías de desarrollo de un DW.....	17
2.7.	Ciclo de vida del DW .....	18
2.8.	Construcción de un DM .....	18
3.	PROBLEMA A RESOLVER.....	21
3.1.	Delimitación del problema .....	21
3.2.	Objetivos .....	22
4.	SOLUCIÓN PROPUESTA.....	23
4.1.	Planificación .....	23
4.2.	Metodología de trabajo y justificación.....	23
4.2.1.	Fase 1: Definición del modelo lógico del negocio .....	23
4.2.2.	Fase 2: Definición del modelo lógico del Data Mart .....	24
4.2.3.	Fase 3: Definición del modelo dimensional.....	27

4.2.4. Fase 4: Definición del modelo físico .....	28
4.2.5. Fase 5: Proceso de ETL .....	29
4.3. Seguimiento del proyecto .....	30
4.4. Herramientas utilizadas .....	31
4.4.1. Entornos de trabajo.....	31
4.5. Desarrollo de la solución .....	32
4.5.1 Fase 1: Definición del modelo lógico del negocio .....	32
4.5.2. Fase 2: Definición del modelo lógico .....	36
4.5.3. Fase 3: Definición del modelo dimensional.....	39
4.5.4. Fase 4: Definición del modelo físico .....	40
4.5.5. Fase 5: Proceso de ETL .....	44
4.5.7. Programación y actualización del cubo.....	64
5. VERIFICACIÓN Y ANÁLISIS DE LOS RESULTADOS .....	66
5.1. Casos de prueba .....	66
5.1.1. Caso de prueba 1 .....	66
5.1.2. Caso de prueba 2 .....	71
5.1.3. Caso de prueba 3 .....	74
5.1.4. Caso de prueba 4 .....	77
5.1.5. Controles adicionales.....	78
6. CONCLUSIONES .....	80
6.1. Líneas Futuras .....	81
7. REFERENCIAS BIBLIOGRÁFICAS .....	82
8. ANEXOS.....	85
ANEXO A: ARCHIVO XML CREADO PARA LA DEFINICIÓN DEL CUBO EMPLEADOS .....	85
ANEXO B: SCRIPT PARA LA EJECUCIÓN DEL JOB DE ETL PRINCIPAL .....	93

# 1. INTRODUCCIÓN

En este capítulo se presenta el contexto del Trabajo Final de Carrera (TFC) y se describe el contenido del mismo.

## 1.1. Contexto

El presente trabajo se desarrolla en el ámbito de la Agencia de Recaudación Tributaria de Río Negro (ARTRN), un organismo público cuyo objetivo es administrar en forma eficiente y eficaz la aplicación, recaudación y fiscalización de los impuestos provinciales.

La gran acumulación de datos en sistemas transaccionales, o simplemente en archivos de oficina constituye un problema, debido a su creciente volumen y diversidad. La dificultad en aprovechar la información aumenta cuando no se dispone de las herramientas necesarias que posibiliten su consulta y sobre todo, es difícil decidir cuáles datos son realmente útiles y reúnen los requisitos de calidad necesarios.

Para poder afrontar estos inconvenientes, surgen los Sistemas de Gestión de Información, que pueden ser muy efectivos cuando se logra combinar una infraestructura de información bien diseñada con la tecnología adecuada. Es decir, no basta con tener toda la información localizada y bien distribuida, si no se tienen las herramientas adecuadas para explotar al máximo el conocimiento que ella encierra.

La ARTRN maneja un alto grado de información con muchos datos por lo que se ha desarrollado desde el año 2012 un Data Warehouse para ayudar a la toma de decisiones, implementado con herramientas Pentaho. El Data Warehouse permite brindar información histórica y de fácil manipulación para las diferentes áreas de la Organización según lo requieran.

Dado que la Agencia se centra principalmente en la recaudación de impuestos, el área de Recursos Humanos (RRHH) queda un tanto relegada de las prioridades de la misma en este sentido.

Esta área actualmente tiene un aplicativo que permite administrar una base de datos con información de todos los empleados de la ARTRN, mediante el cual se

pueden obtener diferentes informes. Pero además cuenta con una estructura de base de datos separada en la que actualizan información adicional y que consideran de interés, que no se refleja en el aplicativo principal y mediante la cual también tienen la posibilidad de realizar ciertos reportes.

Este proyecto aspira a contribuir en la resolución del problema descrito anteriormente, mediante la utilización de la tecnología de data warehousing, proveyendo al área de RRHH con las herramientas que permitan analizar la información disponible y disminuir los costos que implica el desarrollo de reportes para la toma de decisiones en el ámbito de referencia.

## **1.2. Contenido del TFC**

El presente Trabajo Final de Carrera se estructura mediante siete capítulos: “Introducción”, “Estado de la Cuestión”, “Problema a resolver”, “Solución Propuesta”, “Verificación y Análisis de los Resultados” y “Referencias” a los que se agregan dos anexos con información complementaria.

En el capítulo “Introducción” se presenta el contexto de este trabajo y se describe el contenido del mismo.

En el capítulo “Estado de la Cuestión” se introducen los conceptos de Inteligencia de Negocios y su importancia en las Organizaciones, y también los de Data Warehouse y Data Mart. Además se detallan los componentes, metodologías de desarrollo y ciclo de vida de un Data Warehouse y el proceso de construcción de un Data Mart.

El apartado “Problema a resolver” se centra en el problema tratado en este TFC, se propone un objetivo general y objetivos específicos que se pretenden cumplir.

En el capítulo “Solución Propuesta” se describe paso a paso la metodología utilizada para resolver el problema planteado, las herramientas utilizadas para ello y el desarrollo de la solución. También se presenta el resultado final obtenido.

En la sección “Verificación y Análisis de los Resultados” se describen diferentes casos de validación aplicados en la solución propuesta.

En el capítulo “Conclusiones” se enumeran los aportes que presenta este TFC junto a las posibles líneas futuras de trabajos que permitan la evolución del proyecto propuesto.

En el capítulo “Referencias” se lista la bibliografía consultada para el desarrollo del presente TFC.



## 2. ESTADO DE LA CUESTIÓN

En este capítulo se introducen los conceptos necesarios para una mejor comprensión del trabajo llevado a cabo. Se presenta el concepto de Inteligencia de negocios, y la importancia que tiene en una Organización y se definen los conceptos de: Data Warehouse, sus componentes, una metodología para el desarrollo y su ciclo de vida; y también se desarrolla el concepto de Data Mart y su proceso de construcción.

### 2.1 ¿Qué es Inteligencia de negocios (BI<sup>1</sup>)?

El aumento en la digitalización de la información y la utilización de sistemas informáticos hace que las organizaciones cuenten con abundancia de datos. Esto hace que sea muy difícil verlos en su totalidad, analizarlos y poder definir acciones a partir de ellos.

Inteligencia de Negocios (Business Intelligence, BI), se puede definir como un concepto que integra el almacenamiento de grandes cantidades de datos y su procesamiento para poder transformar esta información en conocimiento y decisiones a través del análisis.

En las organizaciones, para mejorar la eficiencia del trabajo se depende mucho de las decisiones que se tomen al respecto, por lo tanto un factor importante es contar con el conocimiento suficiente y explotar de la mejor manera los datos para lograr ese objetivo. Es decir que cuanto más conocimiento obtenga una organización de sus datos va a ser capaz de tomar mejores decisiones.

La Inteligencia de Negocios se centra principalmente en un conjunto de tecnologías, técnicas y métodos para la recopilación, depuración y efectiva utilización de la información.

Las tecnologías que se utilizan cumplen varias funciones como la generación de informes, gestión de rendimiento, funciones analíticas, evolución, optimización de recursos, entre otras.

Las técnicas permiten descubrir asociaciones y anomalías que se tienen entre los datos para brindar a los usuarios la posibilidad de tomar decisiones basadas en conocimiento certero y correcto, disminuyendo la incertidumbre y el

---

<sup>1</sup> BI, por su siglas en inglés Business Intelligence

riesgo que esto conlleva. Además permiten identificar situaciones que supondría un riesgo futuro en la organización o reconocer una oportunidad.

## **2.2. Importancia de BI para la organización**

El objetivo principal de la Inteligencia de Negocios es que ayude en la toma de decisiones, es decir, brindar diferentes alternativas que permitan resolver una situación de forma precisa y rápida. Si bien estas opciones se proporcionan, el resultado de cada búsqueda de conocimiento no siempre es el mismo, sino que depende de la persona que la utilice ya que está condicionado por varios factores como la formación, experiencia, entre otros.

Se espera que los objetivos de la organización o las áreas que la componen se transformen en indicadores que puedan estudiarse para ser analizados desde diferentes puntos de vista, encontrando información histórica, actual, predecir comportamientos futuros y resolver problemas del negocio.

Debido a la gran cantidad de datos y su constante incremento es importante, para evaluar las opciones, contar con el acceso completo a toda la información y de manera eficiente y rápida. Esto se dificulta cuando los datos están estructurados y distribuidos en diferentes sistemas transaccionales por lo que las herramientas de BI actúan como una especie de puente entre todos estos sistemas.

Entre los beneficios que conlleva la implementación de este tipo de soluciones podemos identificar los siguientes:

- Proporción de herramientas de comparación y análisis.
- Flexibilidad al no depender de informes programados, sino que se pueden generar de manera dinámica.
- Reducción del tiempo que implica la obtención de toda la información de un determinado tema y sus relaciones.
- Obtención de indicadores que inciden en un mal funcionamiento de la organización.
- Predicción de comportamientos futuros.
- Integración de los datos que se tienen almacenados en distintos lugares utilizados por diferentes sistemas transaccionales, brindando una visión global de los mismos.

También se pueden enumerar las siguientes desventajas:

- La implementación de BI en una organización o empresa es costosa, no solo porque implica una inversión económica en tecnología sino también en recursos humanos para llevarlo adelante y en tiempo y esfuerzo de capacitación de los usuarios.
- La utilización eficiente de las herramientas surge principalmente de la experiencia por lo que al principio costará la obtención de resultados útiles.
- Incremento de los requerimientos de los usuarios.

### **2.3. Data Warehouse (DW)**

Según William Harvey Inmon, un Data Warehouse es un conjunto de datos históricos, orientados por temas, integrados y no volátiles, que tienen por objetivo dar soporte a la toma de decisiones [Inmon, W. H. 2002]

Un Data Warehouse es un repositorio corporativo de grandes volúmenes de datos, que pueden provenir de uno o varios sistemas operacionales. Además hace accesible la información de la organización en su conjunto de manera consistente, clara y de calidad y tiene como principal objetivo convertir datos en información estratégica para servir como apoyo en la toma de decisiones en las gerencias de la organización. [Bernabeu, R. D. 2009].

Un DW se puede ver como una implementación de una herramienta para la Inteligencia de Negocios que brinda la posibilidad de organizar y almacenar los datos necesarios de forma que la utilización de esta información se haga de manera amigable, intuitiva y directa.

Las principales características o fundamentos en los que se basa un DW, según la definición de W. H. Inmon son:

- Orientado al tema: Los datos se enfocan en diferentes áreas u objetivos estratégicos de interés para la empresa u organización de manera que se puedan analizar los que correspondan a un tema del negocio específico.
- Integrado: Los datos se recolectan de diversos sistemas operacionales que pueden ser de la propia organización o externos. Una característica clave es que están integrados, es decir, son datos consistentes, utilizando una misma codificación, convenciones en los

nombres y atributos físicos de los datos. La información está estandarizada de forma general y almacenada de esa manera con distintos niveles de detalle.

- **Histórico:** Los datos almacenados en un Data Warehouse son históricos, es decir que quedan almacenados en el tiempo para permitir por ejemplo, comparaciones, estudiar la evolución de un determinado elemento, tendencias, entre otros. Muchas veces, un DW se puede ver como una serie de instantáneas o fotos que reflejan el estado de la organización en diferentes momentos en el tiempo.
- **No volátil:** Un Data Warehouse tiene la función de almacenar los datos, una vez hecho esto se convierten en información de sólo lectura. Es decir que no se pueden realizar operaciones de modificación o eliminación, sino que sólo se pueden insertar datos y acceder a ellos para poder cumplir con su propósito que es el análisis. La actualización ocurre a intervalos definidos y es realizada por procesos masivos, no existe una actualización individual de registros por parte de un usuario como es el caso de los sistemas operacionales.

### **2.3.1. Componentes de un Data Warehouse**

Dentro de los componentes básicos de un DW podemos enumerar los siguientes: Sistemas Fuentes, Área Intermedia, Área de Presentación y Herramientas de Acceso. En la Figura 2.1 se pueden observar gráficamente y a continuación se detalla cada una de estas partes.

#### **Sistemas Fuentes**

La información almacenada en un DW surge de la obtención de datos de diversas fuentes y áreas. Las mismas pueden ser internas a la organización, principalmente Bases de Datos (BD) de los sistemas operacionales, pueden también incluir sistemas obsoletos que proveen información histórica, o bien fuentes externas desde donde se reciben datos útiles para la toma de decisiones.

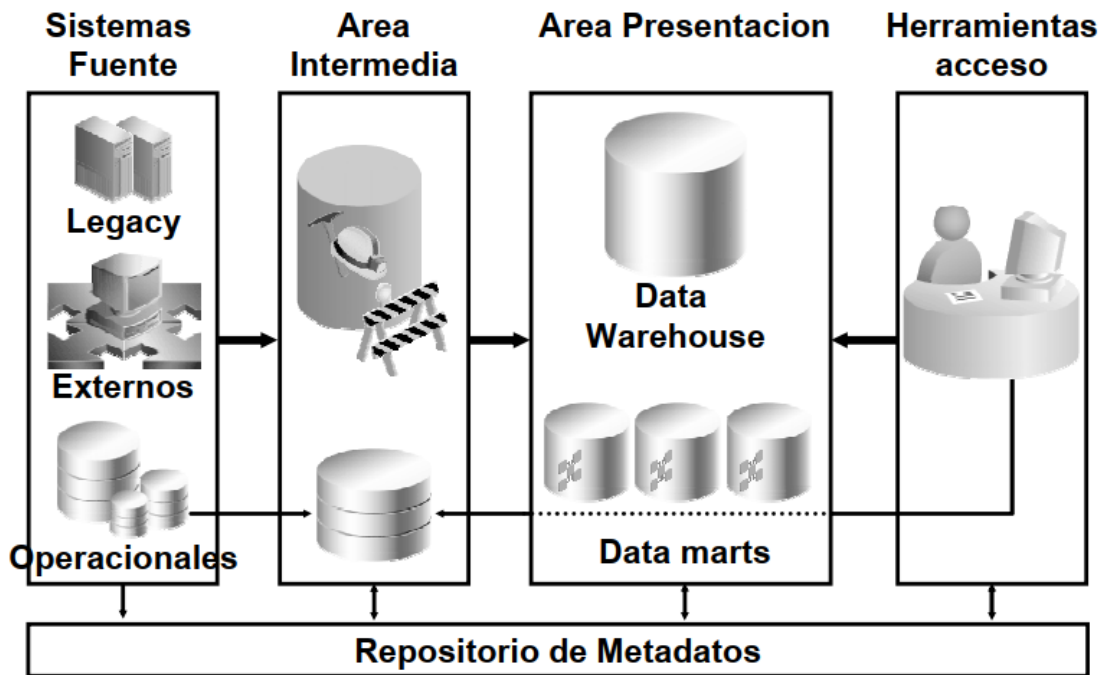


Fig.2.1. Componentes de un DW

### Área intermedia (Staging Area)

La Staging Area es un espacio de almacenamiento intermedio entre las fuentes de datos de origen y el Data Warehouse. Los datos se almacenan en ella dependiendo de las necesidades del negocio y de la factibilidad de implementación. El objetivo principal es facilitar la obtención de datos, realizando una limpieza y transformación de la información sin afectar a los sistemas de origen.

Esta área se puede localizar en la base de datos del DW, en la base de datos del sistema operacional, o eventualmente en un área propia. La disponibilidad de hardware y las velocidades de lectura y escritura a través de la red determinan la mejor opción.

Las ventajas de utilizar un área intermedia son varias: quedan claros y separados los procesos de transformación de datos, de aquellos de carga de datos al DW. Si las transformaciones se realizan en la base de datos operacional, se puede ahorrar gran cantidad de transferencia de datos (los procesos de transformación actúan sobre la misma base de datos, y luego se transfieren solo las tablas resultantes, ya limpias de información operacional). Por otro lado, si la Staging Area no se encuentra en la base de datos del Data Warehouse, se favorece la implementación física del mismo, configurando la base de datos solamente para el procesamiento propio de un DW.

## **Área de presentación**

Uno de los componentes fundamentales de un DW es la base de datos que lo sustenta o área de presentación donde residen los datos en el formato adecuado para su utilización como fuente de información para la toma de decisiones.

## **Herramientas de acceso**

Estas herramientas de acceso o análisis son las que permiten que el usuario final pueda acceder a los datos del DW o DM de una forma amigable, el usuario genera consultas que obtienen datos desde el área de presentación y se los muestra en un formato con el que pueda trabajar.

## **Repositorio de metadatos**

Los metadatos se refieren a la información sobre los datos, sirven para documentar y describir de forma clara todo el contenido dentro de un DW definiendo también el mapeo de las consultas a la base. Cada metadato se define con dominios, reglas de validación y de derivación.

Este repositorio debe ser la única fuente de documentación del DW y tiene que ser accesible desde cualquiera del resto de los componentes.

## **2.4. Data Mart (DM)**

Un Data Mart es un concepto de división lógica de un Data Warehouse. Está enfocado a un determinado tema y a un área de la organización como por ejemplo un departamento o gerencia en particular.

Contiene menos volumen de datos que un Data Warehouse y se puede construir a partir de las necesidades específicas de un área organizacional. De esta manera aísla un conjunto de datos para ofrecer un acceso más fácil a ellos.

Los Data Marts pueden ser dependientes o independientes. Un DM dependiente tiene como fuente de datos el DW. Es decir, extrae sus datos de un DW ya creado y se actualiza a través de él. Generalmente los Data Marts dependientes se crean con propósitos de mejoramiento de performance y disponibilidad, mejor control, o acceso local.

Un Data Mart independiente se construye desde cero a partir las fuentes de datos operacionales o externas y requiere de un proceso de extracción,

transformación y carga similar al de un DW. Normalmente son el primer paso hacia la construcción de un DW empresarial.

## **2.5. Conceptos**

### **2.5.1. Medidas**

Las medidas son métricas o indicadores de un proceso del negocio, están incluidas en las tablas de hechos, que son el núcleo del DW. Una medida (o hecho) contiene un valor numérico, por ejemplo: cantidad de ventas, costo, monto recaudado, promedio de notas, cantidad de empleados.

### **2.5.2. Dimensiones**

Las dimensiones son los parámetros de análisis que categorizan a los procesos del negocio, son los atributos por los cuales una medida puede ser caracterizada o analizada, por ejemplo: clientes, países, tiempo, empleados, oficinas, carreras.

Las dimensiones son las que le dan riqueza al modelo y permiten mejorar el análisis. Cuando se define una dimensión se debe tener en cuenta su estructura y granularidad.

La granularidad de una dimensión es el nivel de detalle al cual se puede llegar en su análisis. Por ejemplo, para la dimensión Tiempo, se puede llegar a un nivel de detalle anual, mensual, diario, etc. A mayor nivel de detalle, más fino es el nivel de granularidad.

Las dimensiones se implementan al nivel más bajo de detalle y se va agregando a los niveles más altos, eso determina una jerarquía en la dimensión. Si la dimensión tiempo se define a nivel día, la jerarquía puede estar compuesta por los niveles: Año, Mes y Día, como lo muestra la Figura 2.2. Otro ejemplo, la dimensión Lugar puede estar definida como una jerarquía con los siguientes niveles: País, Provincia, Ciudad.

Las jerarquías y niveles de las dimensiones (su estructura) facilitan de manera rápida y sencilla el profundizar en el nivel de detalle (en dw se llama drill-down), o disminuir el detalle es decir, ver la generalidad (roll-up).

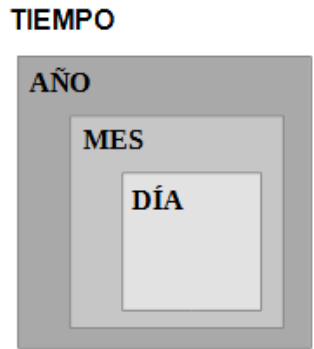


Fig.2.2. Granularidad de la dimensión Tiempo

**2.5.3. Cubo**

Un cubo es un modelo multidimensional para una tabla de hechos determinada con todas las dimensiones involucradas. Se puede ver como una matriz donde cada punto es una combinación de diferentes dimensiones y con una o varias medidas que se necesite analizar en ese conjunto.

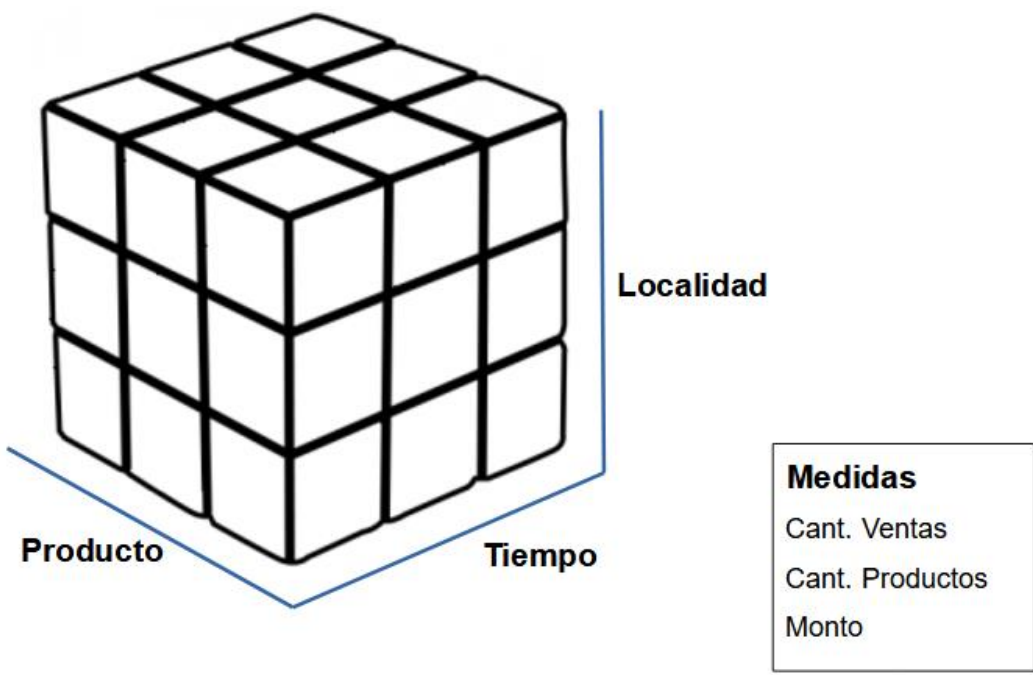


Fig.2.3. Cubo multidimensional

Un modelo multidimensional, adaptable a DW, prioriza la rapidez de acceso a grandes volúmenes de datos en detrimento de evitar la redundancia. El modelo más comúnmente utilizado para diseñar un DW es el Modelo Estrella.



### 2.5.4. Modelo Estrella

El modelo estrella consta de una tabla central que contiene datos de hechos (medidas) y múltiples tablas de dimensiones que se vinculan a la tabla de hechos central por claves primarias (en la dimensión), foráneas (en la tabla de hechos) (Figura 2.4). En este modelo, para cada dimensión, existe una sola tabla que se conecta a la central de hechos. Es por esto que las tablas de dimensiones están des-normalizadas, es decir que no cumplen la tercera forma normal, dado que incluyen dependencias funcionales transitivas. Esta propiedad de des-normalización, si bien repite datos, simplifica la estructura y mejora el acceso en cuanto a velocidad en las consultas.

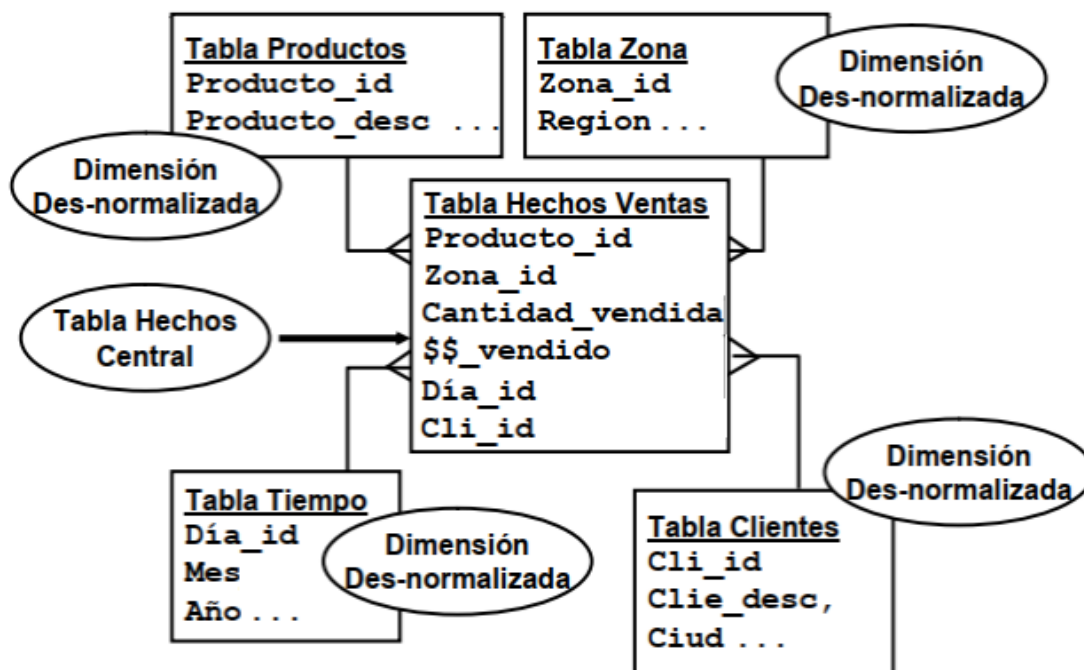


Fig.2.4. Modelo estrella

### 2.6. Metodologías de desarrollo de un DW

Una de las metodologías para el desarrollo de un DW es la denominada Bottom-Up que fue propuesta por Ralph Kimball. Tiene como idea principal llevar adelante la construcción de diferentes Data Marts independientes pero que a su vez se relacionen entre sí para cumplir con las necesidades del negocio. [Ross M., Kimball,R. 2002]

Este tipo de implementación es incremental, es decir que se desarrolla en iteraciones. Si bien se definen los pasos o fases, se vuelve a estos reiteradamente a medida que avanza el proyecto.

En primer lugar, se realiza un relevamiento de la totalidad de los requerimientos a nivel general teniendo en cuenta la disponibilidad de datos en los sistemas operacionales. La definición del área por la que se inicia la construcción depende entonces de la disponibilidad operacional de brindar datos a dicha área, del volumen de esos datos y del beneficio que el desarrollo conlleve. Una vez elegida el área de inicio con este criterio, se ubican los datos que la alimentarán en los sistemas operacionales, se definen los procesos de extracción, transformación y carga, y se crea con ellos un Data Mart que resuelve las necesidades del área sin perder de vista la visión global.

## **2.7. Ciclo de vida del DW**

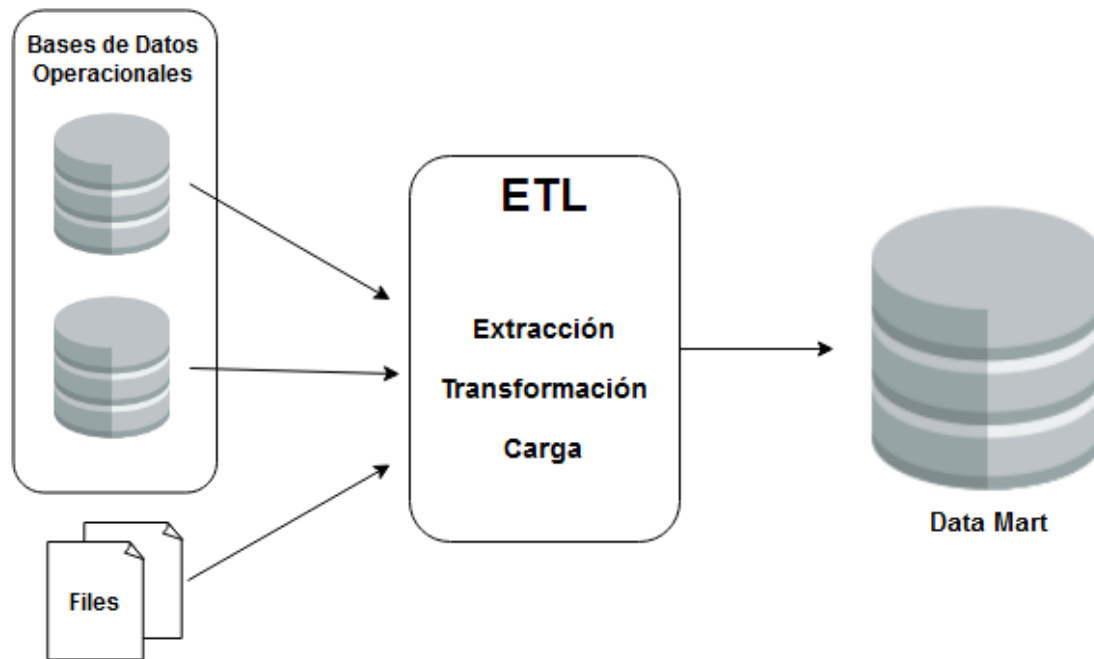
Existen dos tipos de operaciones básicas sobre un DW: la primera de ellas normalmente sucede sola vez, y se denomina Carga Inicial. En esta operación se inicializan todos los datos por los cuales se analiza el negocio y se cargan los datos históricos necesarios al inicio. A partir de ese momento y cada un tiempo estipulado por ejemplo mensualmente, se agregan al DW los datos actuales de funcionamiento, estas cargas se denominan Actualizaciones.

También a lo largo del tiempo, los datos más antiguos pueden resultar obsoletos para el análisis y se puede decidir purgarlos del DW y eventualmente enviarlos a otro almacenamiento.

## **2.8. Construcción de un DM**

El proceso de construcción de un Data Mart se denomina Proceso ETL, que se refiere a la Extracción, Transformación y Carga (ETL, por sus siglas en inglés Extract, Transform and Load). Se puede visualizar en la Figura 2.5.

A continuación se describe cada uno de estos conceptos:



**Fig.2.5.** Proceso ETL

**Extracción de datos:** Las fuentes de datos que alimentan a un DM pueden ser variadas y, en su mayoría internas a la organización, pero también pueden ser externas.

Las rutinas de extracción seleccionan los campos de las fuentes de datos, incluyen reglas de negocio y manejo de errores. Cuando se extraen los datos se puede hacer una extracción completa de los mismos cada vez que se va a actualizar el DM, de manera que las tablas de hechos se limpian y se vuelven a cargar completamente con datos actualizados, o de manera incremental, insertando solo los datos que cambiaron desde la última actualización.

Cuando los sistemas fuentes son de naturaleza histórica, es decir que en su diseño conservan la temporalidad de los hechos, a veces la construcción del DM es más eficiente si se extraen todos los datos, que utilizando una lógica complicada para determinar los cambios, es en esos casos en los que puede tener sentido la extracción completa.

**Transformación de datos:** Los datos deben transformarse en particular para ser presentados en un formato multidimensional. Además, en esta etapa de la construcción es cuando se validan los datos fuente y se limpian de errores, se eliminan inconsistencias, se agregan elementos calculados o derivados, se mezclan e integran datos y/o se suman los mismos.

**Carga de datos:** Aquí es donde definitivamente se mueven los datos al Data Warehouse. Este es un proceso que consume tiempo y recursos, por lo tanto es importante elegir correctamente la ventana temporal de carga, y programar dentro de ella los procesos y su flujo.

La frecuencia de actualización de las tablas de hechos y dimensiones se debe definir en este momento teniendo en cuenta el ciclo de vida del negocio y la disponibilidad de recursos.

### **3. PROBLEMA A RESOLVER**

En este capítulo se presenta la problemática actual que se pretende resolver y se plantean los objetivos generales y específicos.

#### **3.1. Delimitación del problema**

La Gerencia de Recursos Humanos (RRHH) de la Agencia de Recaudación Tributaria de Río Negro (ARTRN) necesita contar con una herramienta de visualización simple, intuitiva y completa donde puedan servirse de información para generar sus propios reportes de análisis e informes sin la intervención constante de la Gerencia de Tecnologías de la Información.

Los reportes que se realizan en la Gerencia de RRHH de forma manual implican mucho tiempo para la elaboración, codificación y mantenimiento de los mismos. Esto también implica que el esfuerzo en su mayoría esté puesto en la obtención de los datos y no en el análisis.

La gerencia cuenta con la implementación de un sistema de gestión de empleados que almacena la información en una base de datos operacional. Si bien desde esta herramienta se pueden generar algunos reportes, estos están predefinidos y el cambio que deseen implementar conlleva un desarrollo o cambio en el sistema.

El área de RRHH también adoptó el uso de los Spreadmarts, que son hojas de cálculo y bases de datos propias. Si bien su uso aporta ventajas a los usuarios, como facilidad en la creación, comodidad, al ser una herramienta conocida por ellos, y control total de los datos, el principal problema es que se acota la visión de la gerencia en su totalidad a un fragmento de la misma, conteniendo cada Spreadmart un grupo o set de datos, cada uno con sus reglas y nomenclaturas.

En este caso, el área utiliza la herramienta Microsoft Access en donde tienen estructurada una base de datos que contiene información formateada que es de importancia para el área.

### **3.2. Objetivos**

Teniendo en cuenta este marco y, dado que el organismo cuenta con la implementación de un Data Warehouse, se plantea como objetivo general el siguiente:

Implementar una solución de Inteligencia de Negocios mediante la creación de un Data Mart que se acople al Data Warehouse existente, para ayudar a la toma de decisiones dentro del área de RRHH.

Objetivos Específicos:

- Estudiar las metodologías de Data Warehouse para desarrollar e integrar el nuevo Data Mart.
- Utilizar la base de datos Oracle con la que cuenta la Agencia y las herramientas que provee.
- Utilizar las herramientas de la suite Pentaho Enterprise utilizada en la construcción y mantenimiento del Data Warehouse de la Agencia.
- Diseñar, construir, implementar y programar la actualización de un Data Mart para permitir la explotación de información analítica de los empleados de la Agencia.

## **4. SOLUCIÓN PROPUESTA**

El presente capítulo detalla la implementación de la solución a la problemática indicada en secciones anteriores. También se mencionan y describen las herramientas y tecnologías utilizadas durante todo el desarrollo del proyecto.

### **4.1. Planificación**

Para poder llevar adelante el proyecto es necesario tener en cuenta factores como el proceso a aplicar para llevarlo a cabo y el resultado que se espera obtener. Se aspira a agregar valor a los datos existentes en la organización y transformarlos en indicadores para que puedan ser analizados desde distintos puntos de vista.

### **4.2. Metodología de trabajo y justificación**

Dentro de las metodologías que se pueden implementar en la creación de un Data Warehouse se encuentran las incrementales. En la ARTRN se llevó a cabo una solución de este tipo siguiendo la metodología sugerida por Ralph Kimball conocida como Bottom-up [Ross M., Kimball R. 2002] que permite la construcción de diferentes Data Marts y es lo que facilita el crecimiento del DW. Este tipo de implementaciones consta de una serie de iteraciones cada una con un conjunto de requerimientos a cumplir. Es decir, se divide el proyecto en etapas sobre las cuales se vuelve en reiteradas ocasiones hasta llegar al resultado final. Para la construcción del presente proyecto se siguen los lineamientos planteados como una nueva iteración, teniendo como objetivo final la construcción de un nuevo Data Mart.

El ciclo de vida que propone Kimball consta de 5 fases que se detallan a continuación.

#### **4.2.1. Fase 1: Definición del modelo lógico del negocio**

Aquí es donde se analizan e identifican los procesos más importantes del negocio y se les asigna una prioridad para la implementación. En esta fase es necesario tener comunicación constante con los usuarios de la Organización para poder decidir juntos los temas que se abordarán en el Data Mart.

#### 4.2.2. Fase 2: Definición del modelo lógico del Data Mart

En esta fase es cuando se definen los denominados cubos que conforman el Data Mart.

Un cubo es un modelo multidimensional que resulta en una tabla de hechos con todas sus dimensiones relacionadas. Aquí es donde se identifican las medidas y las dimensiones.

Las medidas son las métricas del proceso de negocio, físicamente están incluidas en la tabla de hechos. Son valores numéricos, por ejemplo, en el caso de estudio: cantidad de empleados.

Por otro lado, las dimensiones se refieren a los parámetros de análisis o atributos por los cuales una medida puede ser caracterizada y que permiten mejorar el análisis, por ejemplo, en este caso: unidad organizativa, estado civil, título.

Una característica que tienen las dimensiones es que generalmente son lentamente cambiantes, es decir que pueden sufrir modificaciones en el tiempo pero muy lentamente. Por ejemplo, una Oficina puede cambiar de nombre o se pueden agregar nuevas unidades organizativas.

Cuando un registro de una dimensión cambia alguno de sus valores, el DW tiene la posibilidad de almacenar tanto el valor corriente como el histórico. Esto es por la naturaleza histórica del DW, por ejemplo, si una unidad organizativa de la Agencia cambió su nombre en el 2016 y pasó de llamarse “Departamento de Informática” a “Gerencia de Tecnologías de la Información”, el DW deberá resolver la manera en que mostrarán las consultas referidas a ese dato para las fechas (dimensión tiempo) anteriores al 2016 y las posteriores. Los problemas que surgen a partir de esto son:

Si el cambio se realiza directamente en la dimensión, el DW pierde el dato anterior, siguiendo con el ejemplo, las consultas para años anteriores al 2016 tendrán un nombre incorrecto, ya que será el de una unidad que en ese momento no existía.

Si se conserva el dato anterior en la dimensión y se agrega el nuevo registro, los reportes para los años anteriores al 2016 serán correctos, y los reportes para años posteriores también. Esto sucede ya que cada fila de la tabla de hechos se corresponde con cada una de las filas de la dimensión tiempo y con la referencia al nombre correcto. Lo que sucede también en este caso es que las consultas que



incluyan años anteriores y posteriores a 2016 a la vez, mostrarán la unidad dividida en dos filas diferentes, cuando en realidad es una sola que cambió de nombre.

Teniendo en cuenta estos inconvenientes hay 3 maneras de manejar dimensiones lentamente cambiantes:

Sobrescribir el atributo (no mantiene historia). En este caso se dará el problema número 1 descrito anteriormente, es decir que independientemente de la fecha por la que se quiera visualizar el dato, siempre se mostrará el último valor de la dimensión. Es decisión de diseño si esto afecta o no a la información provista.

La segunda opción es agregar un nuevo registro. Esto preserva la historia pero incrementa el tamaño de las tablas. Es decir, cuando hay algún cambio en un dato, se crea una nueva entrada en la dimensión con el nuevo valor y la fecha que corresponda. De esta manera, los registros viejos harán referencia al antiguo valor y los que se creen a partir de ese momento usarán el nuevo registro cargado. Hay que tener cuidado en el tratamiento de claves. Esto puede provocar el problema mencionado número 2 si no se agrega lógica a la herramienta de usuario que permita resolver las consultas que involucran ambos valores del atributo.

Se puede también optar por lo siguiente, mantener el valor actual y agregar campos para retener la información del valor previo y un atributo de fecha. Esto incrementa la complejidad del manejo y el tamaño del registro. Para resolver el problema número 2 también hay que agregar lógica a la herramienta de usuario que consulte las fechas de los registros.

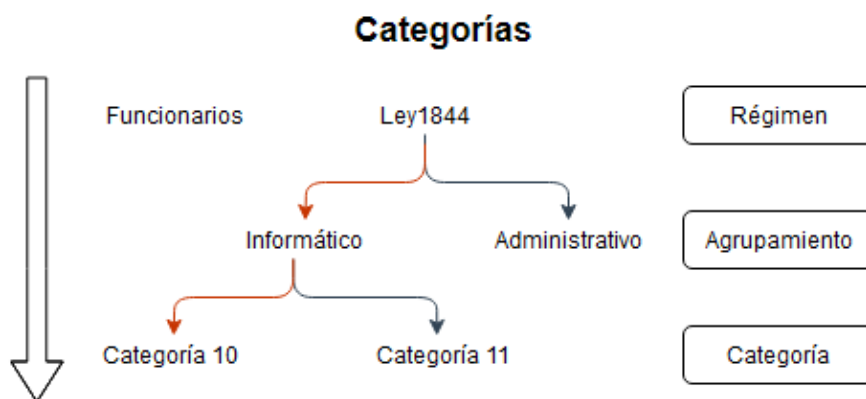
Para este desarrollo se optó por la solución de tipo 1, es decir, se sobrescribe el valor anterior y por lo tanto no mantener la historia. Cuando hay alguna modificación en algún atributo de texto de la dimensión, se sobrescribe el registro. En caso que el cambio haya sido sobre el identificador del registro (ID) en la fuente de datos origen, se agrega una nueva fila a la dimensión formando un nuevo registro.

Otro punto importante al momento de definir las dimensiones es que hay que tener en cuenta la estructura general y la granularidad de cada una de las dimensiones, es decir el nivel de detalle al cual se puede llegar con el análisis para cubrir las necesidades de los usuarios.

Las jerarquías son los diferentes niveles de una dimensión y permiten realizar Data Drilling. Esto significa la investigación a mayor o menor detalle desde un punto de inicio.

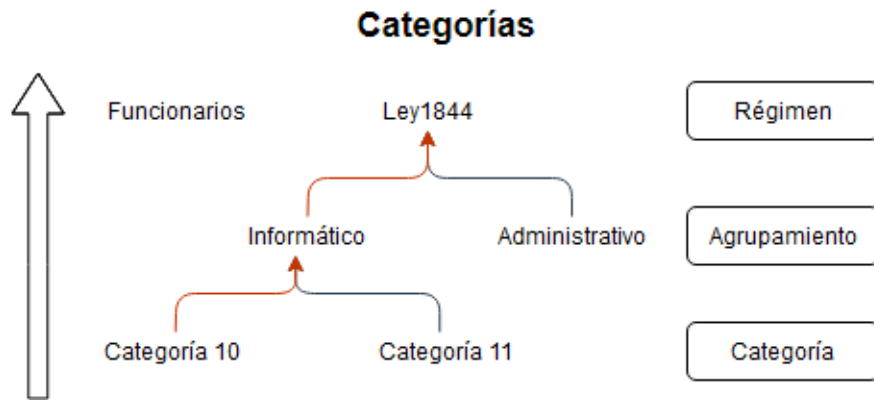
En un entorno analítico se comienza investigando por un nivel alto en la jerarquía, es decir con baja cantidad de detalles, por ejemplo con valores globales o grandes sumas. Partiendo desde ese punto, se puede ir descubriendo hacia abajo por medio de la jerarquía, un mayor detalle. Esto sirve para realizar por ejemplo un análisis global y general sobre un tema en particular y en caso que surja la necesidad, se puede analizar un dato en particular de manera más detallada.

Teniendo en cuenta esto podemos recuperar información a un nivel más fino de granularidad, partiendo desde datos generales a los más específicos, lo que se conoce como Drilling Down. Por ejemplo si se están analizando los empleados por Régimen y luego se toma uno en particular, como puede ser Ley 1844 se puede continuar dicho análisis con el dato del agrupamiento y más en profundidad con el dato de la categoría. Este ejemplo lo podemos observar en la Figura 4.1.



**Fig.4.1.** Operación Drill Down

En cambio, el concepto de Drilling Up o Roll Up, es el proceso inverso que implica generalmente sumarizar. Se habla en este caso de disminuir el nivel de detalle por medio del uso de la jerarquía en nivel ascendente, es decir, ir de lo particular a lo general. Se puede visualizar un ejemplo en la Figura 4.2.



Una dimensión que requiere un análisis especial y detallado en la etapa de diseño es la dimensión Tiempo, dado que es crítica para la flexibilidad del DW e impacta significativamente en el tamaño del mismo (pasar de un nivel inferior de mes a un nivel inferior de día multiplica aproximadamente por 30 la cantidad de filas de las tablas de hechos). Es importante elegir correctamente la granularidad de esta dimensión, en general es preferible tener un grano más fino en el DW y luego tener que sumar para las consultas, que tener un grano grueso y no poder responder algunas solicitudes.

Otra tarea importante en esta etapa de definición lógica del modelo es definir los metadatos, lo que implica documentar el registro de lo que significa cada dimensión, sus niveles y los atributos importantes para definirlos. Se registran también los significados de las medidas y en caso que corresponda, la forma de cálculo de cada una. También se deben documentar las reglas y las definiciones del negocio que acompañan a cada una de estas dimensiones y medidas.

### 4.2.3. Fase 3: Definición del modelo dimensional

Luego de identificar la información que se representa con medidas, las dimensiones involucradas y las jerarquías que las componen con sus respectivos niveles de granularidad, se procede al armado del modelo multidimensional.

Para cada cubo se deben identificar y definir las tablas de hechos y las tablas de dimensiones, y también es necesario saber cómo se conectarán todas estas estructuras entre sí.

En este caso se elige como solución el modelo estrella, que brinda alta performance y se adapta a la mayoría de las herramientas de usuario del mercado, incluyendo la que se utiliza en el presente proyecto. Este modelo se compone de una sola tabla de hechos que se relaciona por medio de claves foráneas a cada una de las tablas de dimensiones.

La tabla de hechos se compone de un conjunto de medidas acompañadas de las claves foráneas a las dimensiones y todas ellas juntas forman la clave primaria de la tabla.

Generalmente, las medidas son aditivas, es decir que para cualquiera de las dimensiones, el resultado de la suma de alguna medida tiene significado para el negocio, como por ejemplo cantidad de empleados dentro de una categoría. Pero existen también las que se denominan no aditivas, las cuales no dan un resultado válido al sumarlas para algunas dimensiones, por ejemplo no tiene sentido sumar la medida Edad para los empleados de una oficina.

Las tablas de dimensiones contienen información de tipo texto que representan a los atributos del negocio. Son datos medianamente estáticos, y se vinculan a la tabla de hechos por medio de sus claves primarias. La idea principal del modelo es que se centra en la riqueza de los atributos de estas tablas dado que son los que determinan cómo se pueden analizar los hechos. Estas tablas se encuentran des-normalizadas dado que incluyen dependencias funcionales transitivas. Por ejemplo, si hay una tabla con las Categorías a la que pertenecen los empleados, la misma contendrá los datos del Agrupamiento y Régimen para cada una de ellas. Esto simplifica la estructura y permite un acceso más rápido en las consultas.

#### **4.2.4. Fase 4: Definición del modelo físico**

En esta fase es donde se traduce el diseño de la etapa anterior a un diseño físico para su implementación en la base de datos. Si bien esta traducción es sencilla, se deben tener en cuenta una serie de factores, como por ejemplo determinar los índices de las tablas, las estrategias de sumarización y cuestiones de seguridad.

Dado que la actualización se realiza automáticamente en un momento programado y el resto del tiempo sólo se ejecutan consultas, hay que tener en cuenta el tipo de tablas e índices que se utilizan. Generalmente para estos casos lo

más eficiente para resolver consultas masivas son los índices bitmap<sup>2</sup> dado que utilizan operaciones lógicas sobre los mapas de bits.

Otro punto importante es que dentro de la Base de Datos que almacene estas tablas, se debe tener en cuenta la recolección de estadísticas ya que estas estructuras de datos pueden crecer muy rápido con una sola carga.

#### **4.2.5. Fase 5: Proceso de ETL**

Esta es la fase final de construcción del Data Mart y consiste en desarrollar el proceso que se denomina de Extracción, Transformación y Carga o ETL.

La Extracción se refiere a la obtención de los datos desde las fuentes de origen, sean bases de datos, sistemas obsoletos que aún guarden información, archivos externos, entre otros.

Estos datos fuentes deben ser transformados, limpiados y reestructurados para poder ser almacenados en las tablas en el formato multidimensional planteado. Se limpian errores, eliminan inconsistencias y unifican granularidades para los que provienen de diferentes fuentes.

Finalmente, los datos se cargan o insertan en las tablas del DW. Este proceso consume tiempo y recursos, por lo tanto es importante definir la ventana temporal donde se llevará a cabo.

En primera instancia se realiza una carga inicial en donde se insertan todos los datos que se definieron para dar respuesta a los usuarios. También si es necesario se realiza la carga de la información histórica en las tablas del modelo del Data Mart. A partir de ese momento y cada cierto tiempo estipulado, se van agregando al DM todos los datos actuales de funcionamiento, esta periodicidad puede ser por ejemplo mensual, semanal, diaria según se requiera. A dichas cargas se las denomina Actualizaciones.

---

<sup>2</sup> Índices Bitmap: Utiliza un mapa de bits para cada valor clave que puede tomar el atributo indexado. Los bits representan las tuplas de la tabla. Cada bit del mapa corresponde a una fila posible. Si el bit está en 1, significa que la fila contiene dicho valor clave, caso contrario el bit es 0.

A medida que transcurre el tiempo puede suceder que la información más antigua ya no sea de utilidad o relevante para el análisis por lo que se puede decidir sacarla del DM y enviarla a otro almacenamiento histórico de archivo.

### 4.3. Seguimiento del proyecto

Para llevar adelante el seguimiento en la totalidad del proyecto se utilizan las herramientas que se detallan a continuación:

Google Drive: Es en donde se registra la documentación y se lleva a cabo el presente Trabajo Final de Carrera dado que permite trabajar con un documento de forma on-line y de manera colaborativa en tiempo real. Además brinda otras facilidades como un control de versiones y la posibilidad de trabajar con el documento sin conexión a internet.

Para el análisis de los requerimientos se realizan entrevistas con el objetivo de investigar el proceso de toma de decisiones y su optimización mediante la herramienta elegida. Estas entrevistas también se registran en Google Drive mediante la siguiente estructura:

Fecha:
Área
Objetivo:
Preguntas o líneas generales:
Resumen:

**Fig.4.3.** Modelo de entrevista utilizado

En donde,

- Fecha: es la fecha en la que se realizó la entrevista
- Área: es la Gerencia o Departamento dentro de la ARTRN a la que pertenecen los entrevistados
- Objetivo: son las principales inquietudes que se pretenden resolver en la reunión realizada

- Preguntas o líneas generales: es el listado de preguntas principales a realizar para llegar al objetivo del encuentro
- Resumen: es una breve descripción de los puntos más importantes de la reunión, incluyendo necesidades, hitos a alcanzar y problemas actuales.

## **4.4. Herramientas utilizadas**

### **4.4.1. Entornos de trabajo**

Dado que la Agencia de Recaudación Tributaria de Río Negro cuenta con un conjunto de herramientas que cubren las necesidades requeridas para llevar adelante el presente trabajo, se utilizaron dichas tecnologías y son las que se mencionan a continuación:

Toad for Oracle Xpert: Utilizado para la administración y gestión de la base de datos. El software fue desarrollado por Quest Software. Y se utilizó la versión 10.6.1.3

Suite Pentaho Enterprise: Es una plataforma compuesta por diferentes herramientas creada en 2004, es una de las líderes en cuanto a soluciones de Inteligencia de Negocios. Ofrece soluciones propias, tanto para desarrollar como para mantener y exportar un proyecto de BI.

El conjunto de herramientas Pentaho sirve para explotar la información generada en el ámbito de la Agencia mediante el uso de reportes analíticos e interactivos, valiéndose de la funcionalidad de un repositorio de datos (Data Warehouse). Cada una de estas herramientas se utiliza en una etapa del proyecto en particular. La versión utilizada es la 8.2 Release 8.2.0.0-324 y los componentes del conjunto de herramientas son los siguientes:

- Pentaho Data Integration (PDI) es la herramienta que permite realizar tareas y procesos ETL. Cuenta con una interfaz gráfica llamada Spoon. Este software es utilizado para la creación, integración, actualización y mantenimiento de los cubos que componen el Data Warehouse de la Agencia, además es utilizado para la automatización de gran cantidad de tareas sobre archivos y base de datos ya que permite a esas fuentes como orígenes de datos y la inserción en ese tipo de estructuras. Es una solución drag & drop, es decir que las tareas o pasos específicos que se apliquen a

los datos se pueden arrastrar desde un listado de una manera fácil, conectándose para lograr el flujo deseado y darle la lógica que se necesite.

- Pentaho Interactive Reporting: herramienta de reporting interactivo.
- Pentaho Analyzer: Es la solución para explotar cubos multidimensionales, gestionado por un motor Mondrian, que es un servidor para el procesamiento analítico. Permite explorar visualmente la información de los cubos armados y cuenta con tecnología apropiada para manejar grandes volúmenes de datos en muy poco tiempo de manera interactiva.
- Pentaho Dashboards: permite la construcción de tableros de control o de mando en la interfaz web. Con esto se pueden ver reportes analíticos, gráficos, mapas y otras herramientas para crear informes amigables en una sola pantalla o documento. Sirve para determinar el estado actual y la tendencia de las variables importantes de la organización con el objetivo de mejorar la comprensión de la información por parte de los usuarios finales.
- Pentaho Enterprise Console: herramienta web que permite la configuración, administración y monitoreo de la plataforma.
- Schema Workbench: Permite la definición de los cubos y es donde se plasma la auto documentación para los usuarios finales.

## **4.5. Desarrollo de la solución**

A continuación se desarrollan las fases descritas anteriormente.

### **4.5.1 Fase 1: Definición del modelo lógico del negocio**

La ARTRN ya cuenta con un DW en funcionamiento que está conformado por diferentes Data Marts como resultado de un proceso evolutivo en el manejo de la información.

Producto de la metodología bottom-up ya explicada, existe entonces un conjunto de Data Marts que componen lo que hoy se denomina el “DW de la Agencia”, que cuenta con información de las siguientes áreas principales [Formia S., Estevez E. 2017]:

- i. RECAUDACIÓN - Contiene información histórica de la recaudación. Se puede clasificar además por fecha de recaudación, impuestos, formas de pago y entidad de cobro.



- ii. CUENTA CORRIENTE - Información detallada e histórica de las obligaciones fiscales de cada contribuyente los saldos y pagos, se pueden también identificar por oficina, tributo, vencimientos, entre otros datos.
- iii. DECLARACIONES JURADAS - Información detallada de las declaraciones realizadas por los contribuyentes de los diferentes tributos.
- iv. RETENCIONES Y PERCEPCIONES - Incluye información en detalle de las retenciones y percepciones tanto de lo declarado por el contribuyente como la que informan los Agentes de Recaudación/Percepción.
- v. DEUDA GESTIONADA - Información de Planes de Pago, Intimaciones, Juicios que influyen en la gestión de la deuda
- vi. MONITOREO USUARIOS / TRÁMITES / MESA DE AYUDA - Contiene información sobre la gestión de usuarios y trámites tanto externos como internos junto a la efectividad en su tratamiento

El presente proyecto tiene como finalidad la creación de un nuevo Data Mart enfocado en los Recursos Humanos que conforman la ARTRN.

En este apartado se explica la definición para la creación del nuevo Data Mart que formará parte del DW descrito en los párrafos anteriores. A esta etapa se la puede dividir en varios pasos, primero se comienza con la preparación de la entrevista y selección de los entrevistados. Luego se desarrollan las preguntas específicas para llegar a un objetivo particular en cada reunión y, finalmente se realiza la entrevista y el análisis de la misma.

Como ejemplo se define a describe a continuación la documentación obtenida en la primera reunión realizada con el área:

<b>Fecha:</b> 9/11/2019
<b>Área:</b> Recursos Humanos
<b>Objetivo:</b> Conocer la situación actual del área de Recursos Humanos de la ARTRN y cómo llevan adelante las tareas. Identificar qué tipos de análisis realizan actualmente

**Preguntas o líneas generales:**

¿Cuál es la función de la Gerencia de Recursos Humanos dentro de la Agencia?

¿Qué sistemas o fuentes de información utilizan para almacenar los datos?

¿Cómo realizan actualmente los informes o reportes?

¿Los informes son realizados para la misma gerencia o son enviados a otras áreas o instituciones?

**Resumen:**

El área cuenta con un aplicativo denominado Legajo Electrónico en donde se guarda información acerca de todos los empleados de la Agencia. Dichos datos son accesibles desde el área para su administración e información; y desde cada empleado para visualizar sus propios datos sin posibilidad de editarlos.

A partir de ese aplicativo tienen la posibilidad de ejecutar reportes que fueron programados desde la Gerencia de Tecnologías de la Información (GTI). El inconveniente que se les presenta es que el cambio, creación o reestructuración de alguno de esos reportes requiere obligatoriamente la intervención de la GTI.

Otra de las herramientas con la que cuentan en el área de RRHH es una base de datos en formato Microsoft Access llamada Planta de Personal, en la que almacenan información adicional y les sirve principalmente como base para reportes que ya tienen estructurados. Si bien para modificar o crear algún reporte en dicha estructura no se necesita un programador, si se requiere una inversión de tiempo por cada una de estas posibles operaciones.

Los informes o reportes creados son utilizados tanto por el área de RRHH, como por otras dentro de la Agencia e incluso algunos son enviados a otros organismos del Estado Provincial.

Una vez investigadas las fuentes de información con la que actualmente llevan a cabo las tareas de análisis manuales que realizan, se lleva a cabo una segunda reunión para poder detallar de manera más específica las necesidades del área dando como resultado la siguiente minuta:

**Fecha:** 20/01/2020

**Área:** Recursos Humanos

**Objetivo:** Identificar necesidades específicas para determinar cuál es la información más relevante y la calidad de la misma

**Preguntas:**

¿Cuáles son los reportes que realizan?

¿A estos reportes se les pueden cambiar los valores de algunos parámetros o son estáticos?

¿Es importante registrar los cambios en el tiempo?

**Resumen:**

Dentro de los reportes que ya tienen estructurados se encuentran entre otros, los siguientes:

- Listado completo de agentes
- Listado de agentes por Unidad Organizativa
- Cantidad de empleados por departamento
- Bajas definitivas y provisorias
- Agentes en condición de jubilarse
- Responsables de un área.

Dichos informes se sacan tanto del sistema de Legajo Electrónico como de la BD Planta de Personal dependiendo donde se encuentre la información. Si bien en muchos de los reportes que se utilizan actualmente la información que se consulta es la última cargada, existe la necesidad en el área de consultar la información histórica. Los datos que se necesitan para la generación de estos reportes la podemos clasificar de la siguiente manera:

- Datos personales de los empleados, como CUIL, DNI, fecha de nacimiento, domicilio, datos del padre y de la madre.
- Datos sobre el grado académico de cada agente, como el máximo título obtenido y la institución que lo avale.
- Información sobre las categorías y agrupamientos y la situación de revista
- Datos sobre el área o lugar de trabajo de cada empleado

A partir de estas reuniones con el área se logran determinar objetivos específicos que se requieren para análisis:

- Ver el número de empleados por género.
- Clasificar a los empleados por edad o rango de edades.
- Conocer el promedio de edad de los empleados por departamento o área.
- Identificar empleados en condiciones de jubilarse.
- Ver empleados distribuidos en las diferentes áreas y/o dependencias.
- Conocer la distribución de los empleados dentro de la provincia o por localidad.
- Ver la cantidad de empleados por categoría administrativa y en caso que tengan, la carga horaria adicional.
- Cantidad de bajas provisorias y definitivas de agentes por área.
- Conocer la cantidad de empleados discriminados por su situación de revista.
- Ver la cantidad de activos y dados de baja dentro de la Agencia

Debido a que es importante para el área contar con la información histórica de todos los agentes y el Data Mart permite este tipo de almacenamiento, se plantea en este caso que el almacenamiento sea en formato de fotos. Esto quiere decir que en cada actualización del DM, se realiza una extracción completa de los datos y se los inserta sin limpiar la información anterior. Esto permite que al consultar algún dato por la fecha de la foto se pueda analizar la información de ese momento del tiempo.

#### **4.5.2. Fase 2: Definición del modelo lógico**

En esta etapa se determinan las medidas y dimensiones y su granularidad. Dado que la información central a obtener es referente a datos de los empleados, se decide realizar la implementación de un cubo denominado "Empleados".

Como se menciona precedentemente, las métricas (medidas) responden a las necesidades del modelo de negocios y para esto se definen las siguientes:

- i. Cantidad: Hace referencia a la cantidad de empleados. La mayoría de los reportes que se necesitan para el análisis responden a las preguntas del tipo: "¿Cuántos empleados cumplen determinadas condiciones?" o "¿Cuáles son los empleados con determinada característica?". Dado que la información en el cubo se plantea como una fila por empleado y manteniendo información histórica, la forma de

contar a los mismos para responder a preguntas similares a la primera es contando las filas por fecha de carga de la foto histórica.

- ii. Edad: Esta medida a diferencia de la anterior, es de tipo no aditiva debido a que no tiene sentido sumarla. Lo que se pretende ofrecer es la posibilidad de realizar promedios, por ejemplo, promedio de edad de los empleados de un determinado sector o filtros que respondan al siguiente enunciado: empleados activos mayores a 60 años.

Por otro lado, como dimensiones para poder resolver las necesidades del área, se plantean las siguientes:

- i. Fecha de foto: En primer lugar se considera la dimensión tiempo para garantizar la perspectiva de almacenamiento histórico que tiene la información, y a la que se debe poder acceder para el análisis de tipo evolutivo de la planta de empleados de la Organización. Es importante notar que cuando se requieren datos, éstos deben ser consultados contemplando siempre un intervalo de tiempo. Esta dimensión está conformada por una jerarquía con tres niveles: Año, Mes y Día.
- ii. Empleados: Hace referencia a los datos particulares y personales de cada empleado de la Agencia como: apellido y nombre, número de CUIL/CUIT, número de legajo, nombre completo del padre y de la madre y número de documento.
- iii. Categoría: Esta dimensión se conforma como una jerarquía con niveles que hacen referencia a las clasificaciones mediante escalafones dentro de la Organización. Cada escalafón está constituido por categorías, cada una de las cuales se corresponde con un agrupamiento y a su vez, este último se asocia con un régimen o estatuto. Los tres niveles identificados nombrándolos de menor a mayor detalle son: Régimen, Agrupamiento y Categoría.
- iv. Estado civil: Una característica por la que es necesario clasificar a los agentes de la ARTRN es el estado civil, para ello se crea una dimensión de un solo nivel ya que no se puede realizar una subclasificación de este dato.
- v. Nacionalidades: Esta dimensión se crea para brindar información sobre las nacionalidades de los empleados.

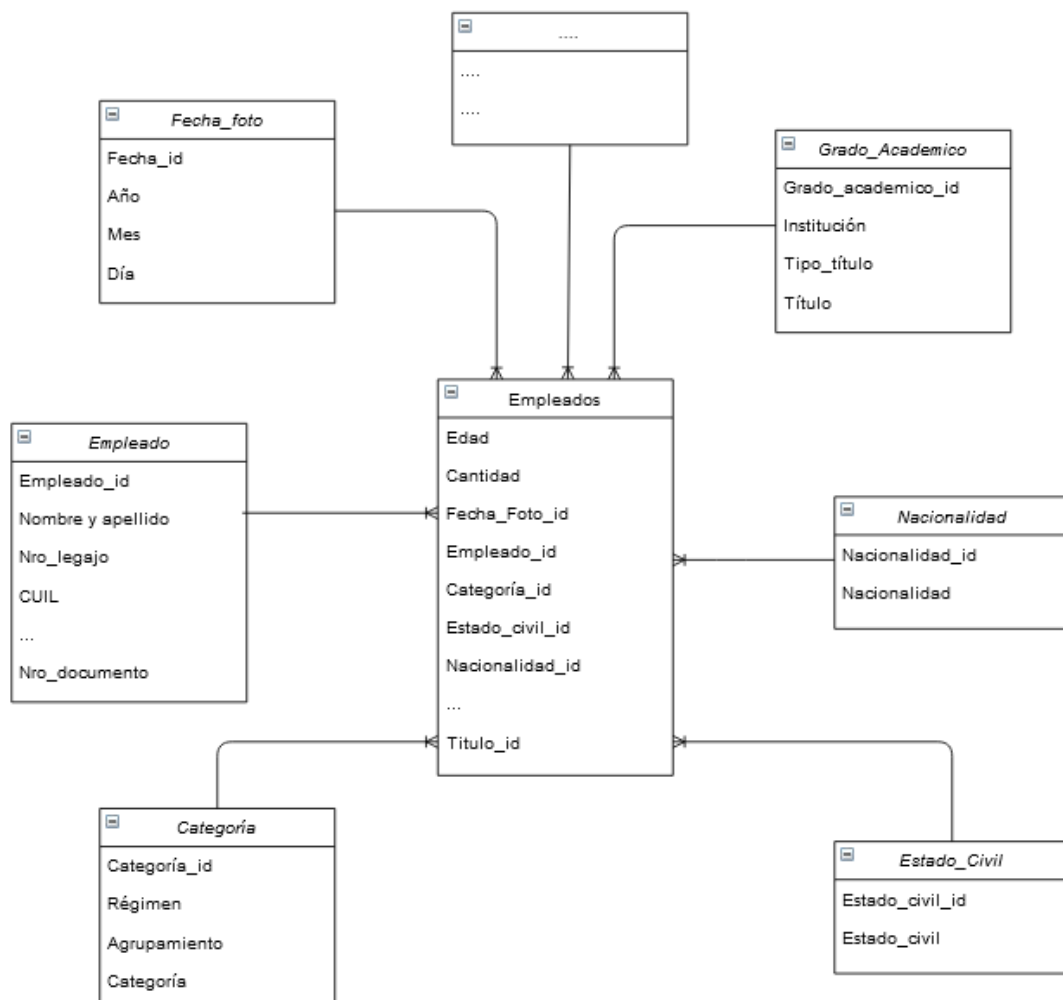
- vi. Unidad organizativa: Esta dimensión se refiere a la sede de funciones del empleado que puede ser por ejemplo una subdelegación, receptoría, gerencia, entre otras.
- vii. Organización: Esta dimensión indica el Área/Gerencia donde se ubica el empleado, por ejemplo Tecnologías de la Información.
- viii. Régimen jubilatorio: Para algunos reportes es necesario conocer si un empleado está o no jubilado. Para eso se crea esta dimensión de un solo nivel.
- ix. Situación de Revista: Dado que los empleados de la Agencia pueden ser clasificados según su situación de revista como por ejemplo planta permanente, se definió la dimensión pensando en esta división.
- x. Grado Académico: Esta dimensión tiene diferentes niveles e indica el mayor grado académico obtenido por el agente. Con el mayor grado de detalle, se define el nivel Título, que especifica el nombre del máximo título obtenido por el agente, como siguiente nivel, se encuentra Tipo de Título que indica por ejemplo Secundario completo o Técnico universitario completo. Finalmente se define la Institución en donde el empleado alcanzó su máximo grado académico
- xi. Carga horaria: Especifica si el agente cuenta con carga horaria adicional.
- xii. Licencia sin goce de haberes: Esta dimensión indica si el empleado cuenta con una licencia sin goce de haberes.
- xiii. Baja Provisoria o definitiva: Con esta dimensión se puede saber si el agente cuenta con una baja en proceso o provisoria o si tiene una baja definitiva
- xiv. Género: La dimensión indica el género del empleado.
- xv. Localidad: Esta dimensión es de dos niveles e indica la localidad en donde cumple sus funciones el agente y en otro nivel la provincia a la que corresponde dicha localidad.
- xvi. Estado: Esta dimensión sirve para clasificar a los empleados en Activos o Dados de Baja

### 4.5.3. Fase 3: Definición del modelo dimensional

En esta fase identifica la tabla de hechos y las tablas de dimensiones asociadas a ella.

Al comenzar con el modelado multidimensional del Data Mart, se definió la utilización e implementación de un modelo estrella. De esta manera se implementa una tabla central conocida como tabla de hechos y múltiples tablas conectadas por medio de identificadores que representan a cada una de las dimensiones.

Este tipo de implementación conlleva que los datos no estén normalizados evitando u obviando joins entre tablas para realizar consultas haciendo que el tiempo de respuesta sea mejor. Para ello se define una tabla central de hechos denominada EMPLEADOS, y una tabla para cada una de las dimensiones que se describen una a una en el próximo apartado. (una parte de este modelo se puede ver en la Figura 4.4.)



**Fig.4.4.** Fragmento del modelo estrella creado para representar el cubo

#### 4.5.4. Fase 4: Definición del modelo físico

El modelo dimensional se tradujo a uno físico para implementar la solución. Este esquema físico se define en la instancia de base de datos Oracle donde se encuentra el resto del DW de la ARTRN.

Todas las dimensiones cuentan con datos en común que son:

- DIMENSION\_KEY: Es el identificador y por lo tanto la clave primaria de cada una de las tablas de dimensiones.
- VERSION: Indica el número de versión del dato. Dado que las dimensiones pueden variar en el tiempo, siendo cada dato relativo a un período en particular se va guardando en este campo un número incremental que se suma cada vez que un valor cambia. La versión es utilizada cuando se manejan dimensiones lentamente cambiantes tipo 2.
- DATE\_FROM: Indica la fecha desde que se tiene que tener en cuenta el dato.
- DATE\_TO: Esta columna guarda, en caso que tenga, la fecha límite del dato.

Para todas las dimensiones se crea una tabla, cada una con una clave primaria (PK, Primary Key) e índices que aceleran la búsqueda en cada una de ellas. A continuación se define el nombre y los campos de cada una de las tablas creadas:

- Dimensión Empleado:

Se crea para esta dimensión una tabla denominada LE\_EMPLEADOS. La misma cuenta con la columna DIMENSION\_KEY como PK y además tiene creados tres índices, uno para la PK, otro para el campo APYNOM (apellido y nombre) y el último para el dato de CUIL dado que son los que se usarán en su mayoría para realizar listados y filtros.

En la Figura 4.5. se puede visualizar la estructura de la tabla creada para la dimensión Empleados y en la Figura 4.6. se muestra a modo de ejemplo la sentencia SQL ejecutada para la creación de la misma.



Column Name	ID	Pk	Null?	Data Type
DIMENSION_KEY	1	1	N	INTEGER
VERSION	2		Y	INTEGER
DATE_FROM	3		Y	DATE
DATE_TO	4		Y	DATE
ID_EMPLEADO	5		Y	NUMBER
NRO_LEGAJO	6		Y	VARCHAR2 (40 Byte)
APYNOM	7		Y	VARCHAR2 (100 Byte)
CUIL	8		Y	VARCHAR2 (40 Byte)
MADRE_APYNOM	9		Y	VARCHAR2 (30 Byte)
DATOS_MADRE	10		Y	VARCHAR2 (55 Byte)
PADRE_APYNOM	11		Y	VARCHAR2 (100 Byte)
DATOS_PADRE	12		Y	VARCHAR2 (55 Byte)
NRO_DOCUMENTO	13		Y	VARCHAR2 (40 Byte)

**Fig.4.5.** Estructura de tabla LE\_EMPLEADOS

```

CREATE TABLE DW.LE_EMPLEADOS
(
  DIMENSION_KEY INTEGER,
  VERSION        INTEGER,
  DATE_FROM     DATE,
  DATE_TO       DATE,
  ID_EMPLEADO   NUMBER,
  NRO_LEGAJO    VARCHAR2(40 BYTE),
  APYNOM        VARCHAR2(100 BYTE),
  CUIL          VARCHAR2(40 BYTE),
  MADRE_APYNOM  VARCHAR2(30 BYTE),
  DATOS_MADRE   VARCHAR2(55 BYTE),
  PADRE_APYNOM  VARCHAR2(100 BYTE),
  DATOS_PADRE   VARCHAR2(55 BYTE),
  NRO_DOCUMENTO VARCHAR2(40 BYTE)
);

ALTER TABLE DW.LE_EMPLEADOS ADD (
  CONSTRAINT LE_EMPLEADOS_PK
  PRIMARY KEY
  (DIMENSION_KEY)
  USING INDEX DW.LE_EMPLEADOS_PK);

CREATE UNIQUE INDEX DW.LE_EMPLEADOS_APYNOM ON DW.LE_EMPLEADOS
(APYNOM);

CREATE UNIQUE INDEX DW.LE_EMPLEADOS_CUIL ON DW.LE_EMPLEADOS
(CUIL);

CREATE UNIQUE INDEX DW.LE_EMPLEADOS_PK ON DW.LE_EMPLEADOS
(DIMENSION_KEY);

```

**Fig.4.6.** Script de creación de la tabla LE\_EMPLEADOS

- Dimensión Categoría

Para representar a esta dimensión se define una tabla denominada LE\_AGRUP\_CAT donde se incluyen los datos de los tres niveles que la componen y se puede observar en la figura 4.7. También se asigna como clave primaria a la columna DIMENSION\_KEY y a su vez, se define un índice sobre dicha columna.

Column Name	ID	Pk	Null?	Data Type
DIMENSION_KEY	1	1	N	INTEGER
VERSION	2		Y	INTEGER
DATE_FROM	3		Y	DATE
DATE_TO	4		Y	DATE
REGIMEN	5		Y	VARCHAR2 (100 Byte)
ID_REGIMEN	6		Y	NUMBER
AGRUPAMIENTO	7		Y	VARCHAR2 (100 Byte)
ID_AGRUPAMIENTO	8		Y	NUMBER
CATEGORIA	9		Y	VARCHAR2 (100 Byte)
ID_CATEGORIA	10		Y	NUMBER

**Fig.4.7.** Estructura de la tabla LE\_AGRUP\_CAT

- Dimensión Estado Civil:
  - Nombre de la tabla: LE\_ESTADOS\_CIVILES
  - Columnas: dimension\_key (PK), version, date\_from, date\_to, id\_estado\_civil, estado\_civil.
- Dimensión Nacionalidad:
  - Nombre de la tabla: LE\_NACIONALIDADES
  - Columnas: dimension\_key (PK), version, date\_from, date\_to, id\_nacionalidad, nacionalidad.
- Dimensión Organización
  - Nombre de la tabla: LE\_ORGANIZACIONES
  - Columnas: dimension\_key (PK), version, date\_from, date\_to, id\_organizacion, organizacion.
- Dimensión Régimen jubilatorio
  - Nombre de la tabla: LE\_REG\_JUBILATORIOS
  - Columnas: dimension\_key (PK), version, date\_from, date\_to, id\_reg\_jubilatorio, reg\_jubilatorio.

- Dimensión Situación de Revista
  - Nombre de la tabla: LE\_SIT\_REVISTA
  - Columnas: dimension\_key (PK), version, date\_from, date\_to, id\_sit\_revista, situacion\_revista.
- Dimensión Unidad Organizativa
  - Nombre de la tabla: LE\_UN\_ORGANIZATIVAS
  - Columnas: dimension\_key (PK), version, date\_from, date\_to, id\_un\_organizativa, un\_organizativa
- Dimensión Grado Académico
  - Nombre de la tabla: LE\_GRADOS\_ACADEMICOS
  - Columnas: dimension\_key (PK), version, date\_from, date\_to, id, titulo, id\_tipo\_titulo, tipo\_titulo, id\_institución, institucion
- Dimensión Carga Horaria
  - Nombre de la tabla: LE\_CARGAS\_HORARIAS
  - Columnas: dimension\_key (PK), version, date\_from, date\_to, id\_carga\_horaria, carga\_horaria
- Dimensión Licencia Sin Goce de Haberes
  - Nombre de la tabla: LE\_LIC\_SIN\_HABERES
  - Columnas: dimension\_key (PK), version, date\_from, date\_to, id\_lic\_sin\_haberes, lic\_sin\_haberes
- Dimensión Baja Definitiva/Provisoria
  - Nombre de la tabla: LE\_BAJAS
  - Columnas: dimension\_key (PK), version, date\_from, date\_to, id\_baja, baja
- Dimensión Género
  - Nombre de la tabla: LE\_GENEROS
  - Columnas: dimension\_key (PK), version, date\_from, date\_to, genero
- Dimensión Localidad
  - Nombre de la tabla: LE\_LOCALIDADES
  - Columnas: dimension\_key (PK), version, date\_from, date\_to, id\_localidad, localidad, id\_provincia, provincia
- Dimensión Estado
  - Nombre de la tabla: LE\_ESTADOS

- Columnas: dimension\_key (PK), version, date\_from, date\_to, id\_estado, estado
- Dimensión Fecha Foto:
  - En el DW ya existía una dimensión que es utilizada para indicar las fechas en la mayoría de los cubos existentes. Dado que la estructura de la tabla, denominada TIEMPOAMD, permite generar el nivel de granularidad que se necesita para este cubo, se utiliza dicha estructura evitando la creación de una nueva tabla.

Finalmente, se define la tabla de hechos denominada CUBO\_EMPLEADOS que está compuesta por los hechos mencionados anteriormente y las claves foráneas a todas las dimensiones ya descritas, la estructura se puede observar en la Figura 4.8.

También en esta instancia, se crean una serie de índices bitmap sobre las columnas: EMPLEADO, FECHA\_FOTO y UNIDAD\_ORG.

Column Name	ID	Pk	Null?	Data Type
EMPLEADO	1	Y	Y	NUMBER
FECHA_FOTO	2		Y	NUMBER
AGRUP_CAT	3		Y	NUMBER
ESTADO_CIVIL	4		Y	NUMBER
NACIONALIDAD	5		Y	NUMBER
UNIDAD_ORG	6		Y	NUMBER
ORGANIZACION	7		Y	NUMBER
REG_JUBILATORIO	8		Y	NUMBER
SITUACION_REVISTA	9		Y	NUMBER
GRADO_ACADEMICO	10		Y	NUMBER
CARGA_HORARIA	11		Y	NUMBER
LIC_SIN_HAB	12		Y	NUMBER
BAJA	13		Y	NUMBER
GENERO	14		Y	NUMBER
LOCALIDAD	15		Y	NUMBER
EDAD	16		Y	NUMBER
ESTADO	17		Y	NUMBER

**Fig.4.8.** Estructura de la tabla de hechos CUBO\_EMPLEADOS

#### 4.5.5. Fase 5: Proceso de ETL

En esta última fase se realiza la selección y extracción de datos y se prepara el ambiente para la carga de los mismos al Data Mart en un formato acorde para la utilización por parte de las herramientas de análisis.

Se utilizó la herramienta Kettle del cliente de Pentaho Data Integration (Spoon) que es una aplicación de escritorio que posibilita realizar el proceso de ETL. Esta herramienta permite crear dos tipos de archivos:

- Transformaciones (Transformations en inglés): En estos archivos es en donde se lleva a cabo la tarea de extracción, transformación y carga de los datos. Gráficamente se pueden ver como una red de tareas lógicas.
- Trabajos (o Jobs en inglés): Definen y organizan a las transformaciones estableciendo el flujo de trabajo y el orden en que deben ejecutarse. Pueden contener pasos adicionales pero la función principal es coordinar recursos y la ejecución de las diferentes actividades.

Las soluciones que se pueden crear con esta herramienta tanto para los trabajos como para las transformaciones son una secuencia de entradas o pasos (steps en inglés) que se van uniendo entre ellos mediante saltos (hops en inglés).

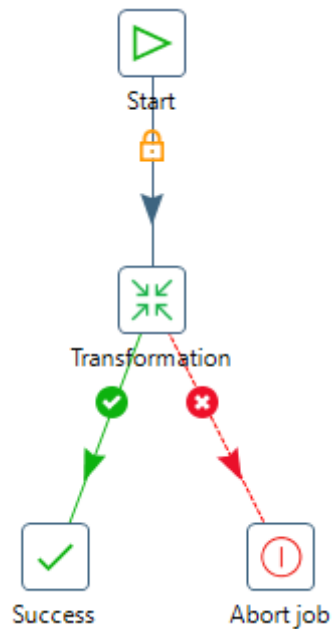
Cada uno de los steps realiza una tarea específica, como leer o insertar datos de una tabla de una base de datos, reemplazar un texto, o llamar a otra transformación o trabajo. En cambio, los hops son los encargados de transportar los datos entre un step y otro. Estos últimos se pueden ver gráficamente desde la herramienta como una flecha con dirección que sirve de ruta para conectar dos o más pasos.

Un flujo de trabajo comienza con un paso de 'Start' y finaliza con un paso de 'Success' si termina bien la secuencia, en caso contrario si hay algún fallo o error el step que se utiliza para darle fin es el de 'Abort Job'. En el ejemplo de la Fig. 4.9, el job inicia la ejecución con el paso de Start y seguidamente ejecuta una transformación. Desde ese punto se desprenden dos posibilidades o flujos, si la transformación finalizó correctamente continúa por el Step Success, caso contrario, finaliza con un Abort Job.

Un job sigue flujos que están determinados por flechas de dirección que unen a cada uno de los steps. Estos flujos se ven de manera gráfica y se pueden identificar de la siguiente manera:

- El camino que se recorre siguiendo las flechas verdes y con una tilde es cuando cada paso finaliza con éxito.
- El rojo con una cruz significa que el paso falló y por lo tanto se tiene que desviar el recorrido para realizar otra acción.



- Y por último cuando hay un candado en el salto significa que independientemente del resultado del paso anterior, se seguirá el recorrido por ese camino.



**Fig.4.9.** Ejemplo de un job

De todos los steps que dispone el Kettle para la creación de un trabajo, se utilizan en este caso los que se muestran en la Tabla 4.1.

Nombre	Descripción	Icono
Start	Se utiliza como el primer paso para identificar desde donde debe comenzar el job	
SQL	Ejecuta una secuencia SQL en una base de datos conectada	
Transformation (Job Entry)	Ejecuta una transformación ya creada y guardada	
Job	Ejecuta un job ya creado y almacenado. Esto permite	

	dividir los jobs en otros más pequeños y manejables	
Success	Fuerza a un estado de éxito el job, eliminando cualquier estado de error	
Abort job	Aborta el job que se está ejecutando	

**Tabla 4.1.** Descripción de los pasos o step utilizados en los jobs.

Para la creación y definición de las transformaciones se utiliza otro tipo de steps, dado que la finalidad de la misma es diferente a la de un trabajo. Para el presente proyecto se utilizan los pasos que se listan en la Tabla 4.2.










Nombre	Descripción	Icono
Table input	Este paso lee información de una base de datos mediante sentencias SQL	
Microsoft Access input	Lee los datos desde un archivo Microsoft Access	
Stream Lookup	Busca valores en otro flujo de la transformación y los une a la misma	
Select Values	Permite realizar diversas acciones en los campos del flujo de datos. Como pueden ser seleccionar, eliminar o cambiar los nombres, longitud y tipo de dato de los valores	
Replace in string	Reemplaza todas las apariciones de una palabra por otra	

Table output	Inserta información en una tabla de una base de datos conectada	
Dimension lookup/update	Se utiliza para cargar o actualizar una dimensión	
Sort rows	Ordena todo el flujo de información por el o los datos que se le indiquen, de forma ascendente o descendente	
Unique rows	Elimina las filas duplicadas y devuelve como salida ese filtro realizado. Tiene como prerequisite que los datos por los que se realice el filtro estén ordenados	

**Tabla 4.2.** Descripción de los pasos o steps utilizados en las transformaciones

#### 4.5.5.1. Preparación de los datos

Los datos necesarios para poder llevar adelante el proyecto se toman de dos orígenes distintos, el primero es una base de datos que sirve como repositorio del sistema de legajo electrónico. Éste es un sistema web para la gestión de los empleados en donde se registra y administra la información, movimientos y actividades que llevan a cabo los trabajadores. Esta base de datos, reside en un sistema de gestión Oracle. Para el presente proyecto solo se consideran las tablas pertenecientes al esquema llamado LEDGR, que contiene el núcleo del modelo de datos de la aplicación.

El segundo lugar de origen desde donde se consumen los datos para abastecer el Data Mart es una base de datos gestionada mediante la herramienta Microsoft Access. Dicha estructura de datos es gestionada desde el área de RRHH de la Agencia y en ella se almacena información adicional que no se encuentra en la BD del legajo electrónico mencionada anteriormente. Esta estructura de datos se nombrará de ahora en más como Base de Planta de Personal.



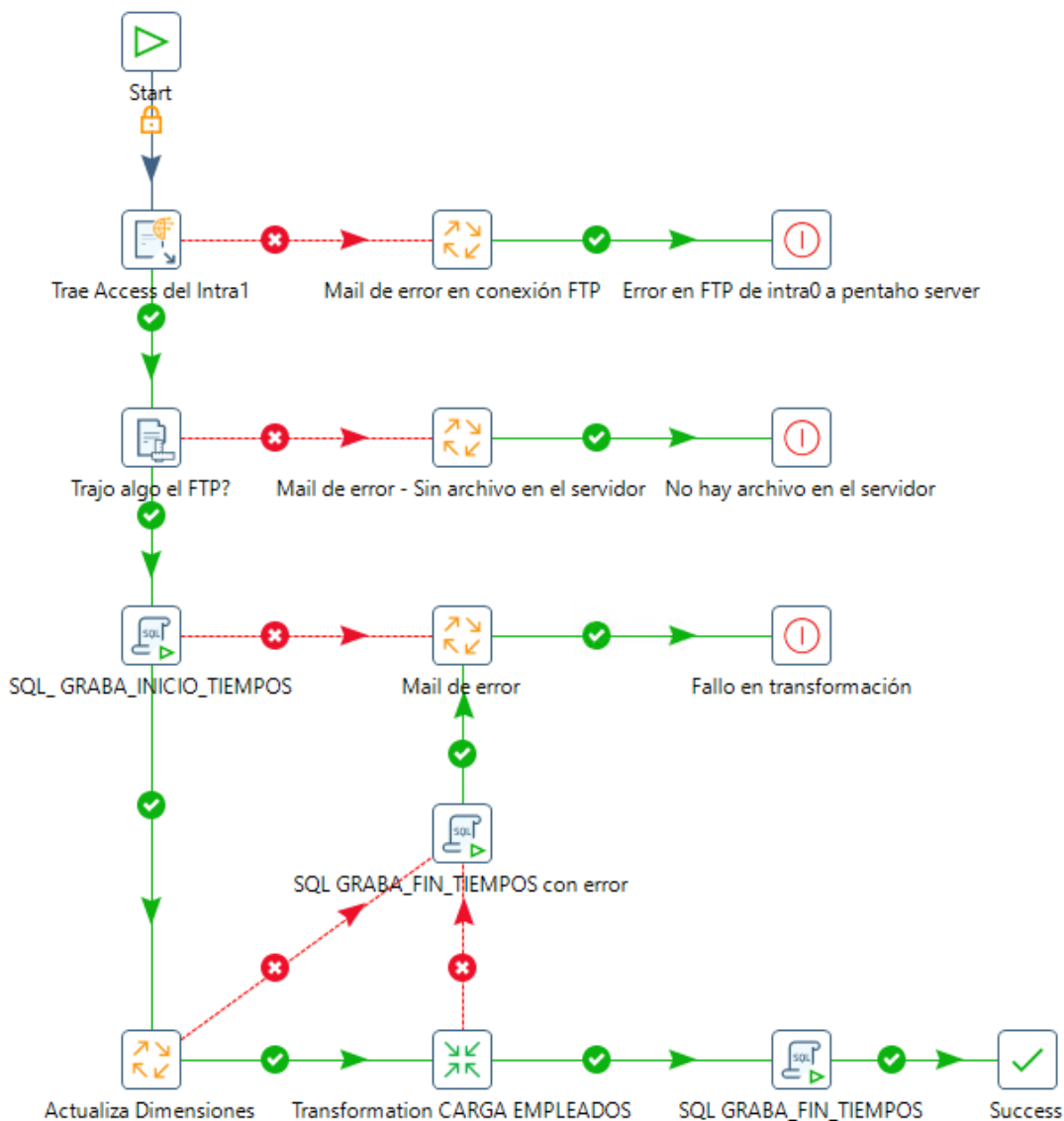
Para poder obtener los datos desde la BD LEDGR y pasarla al área de trabajo del Data Mart, se crean vistas en el área temporal o staging. Las mismas acceden a las tablas de origen, y son las siguientes:

- **VW\_LE\_EMPLEADOS:** Con esta vista se obtienen los datos de los empleados, entre los que se encuentran: ID, Número de legajo, Apellido y Nombre, CUIL, DNI y los datos del padre y/o de la madre (Nombre y DNI de cada uno)
- **VW\_LE\_AGRUP\_CAT:** Se obtienen los datos de los Regímenes (por ejemplo, Funcionarios o Ley 1844), de los Agrupamientos (como por ejemplo Ley 1844 Informáticos) y de todas las Categorías (Categoría 1, Categoría 2) que conforman el dominio de la dimensión.
- **VW\_LE\_ESTADOS\_CIVILES:** El resultado de ejecutar esta vista es el listado de los estados civiles que puede tener una persona, como por ejemplo, soltero, casado, viudo, etc., junto a cada uno de sus ID.
- **VW\_LE\_NACIONALIDADES:** Se obtienen las posibles nacionalidades de los empleados de la Agencia y sus ID.
- **VW\_LE\_ORGANIZACIONES:** Consulta una tabla del esquema LEDGR y trae la descripción de las organizaciones dentro de la ARTRN, o el área específica donde desarrolla las tareas un empleado. Por ejemplo: Dirección ejecutiva, Atención al contribuyente, Desarrollo, Inteligencia de Negocios y Bases de Datos, etc. También se obtiene el ID de cada uno.
- **VW\_LE\_REG\_JUBILATORIOS:** Los resultados que muestra esta vista hacen referencia al tipo de régimen jubilatorio del agente y además el ID. Dentro de los valores posibles se pueden encontrar, No aporta, Reparto, Retirado-jubilado, entre otros.
- **VW\_LE\_SIT\_REVISTA:** Lista las posibles situaciones de revista del cargo que posee el empleado, un ejemplo de ella es Contratado o Planta permanente. También se recupera el ID.
- **VW\_LE\_TITULO:** Esta vista consulta varias tablas para obtener el listado de grados académicos y el tipo (secundario, terciario, entre otros), junto a la institución que lo certifica y el ID de cada uno de ellos.
- **VW\_ GENERO:** Esta vista obtiene el listado de los distintos tipos de género.

- VW\_LE\_LOCALIDADES: La vista obtiene un listado de localidades y la provincia a la que pertenece y el ID de cada una.
- VW\_ESTADOS: Esta vista retorna el valor de los posibles estados de un agente junto a su ID.

#### 4.5.5.2. Job principal

Para definir el job principal, que es el que organiza y estructura todo el proceso ETL, se creó un archivo con el nombre job\_empleados.ktr y la estructura es la que se muestra en la Figura 4.10.



**Fig.4.10.** Pasos del job principal (job\_empleados.ktr)

Cada uno de los pasos utilizados en el job tienen dos posibles caminos, finalizar correctamente y seguir con el siguiente, o bien terminar con algún error, por lo que el flujo cambia y continúa por uno alternativo que consiste en enviar un mail a la casilla de correos especificada indicando el fallo y abortando el job.

En rasgos generales, se puede dividir este proceso en tres etapas y son: Búsqueda del archivo Access, Actualización de dimensiones y Carga de la información en el cubo. A continuación se detalla cada una de ellas:

### **Búsqueda del archivo Access**

La primera parte se encarga de buscar el archivo en Access, Base Planta de Personal con un acceso FTP en la carpeta donde está ubicado y subir una copia al servidor de Pentaho para el tratamiento de sus datos.

Durante esta parte del proceso se realizan dos controles, el primero es sobre la conexión FTP (Protocolo de Transferencia de Archivos), en caso que esté activa transfiere el archivo al lugar indicado dentro del servidor de Pentaho y continúa con la siguiente verificación. En cambio, si ocurre algún fallo, se envía un mail avisando de este error y finaliza el proceso.

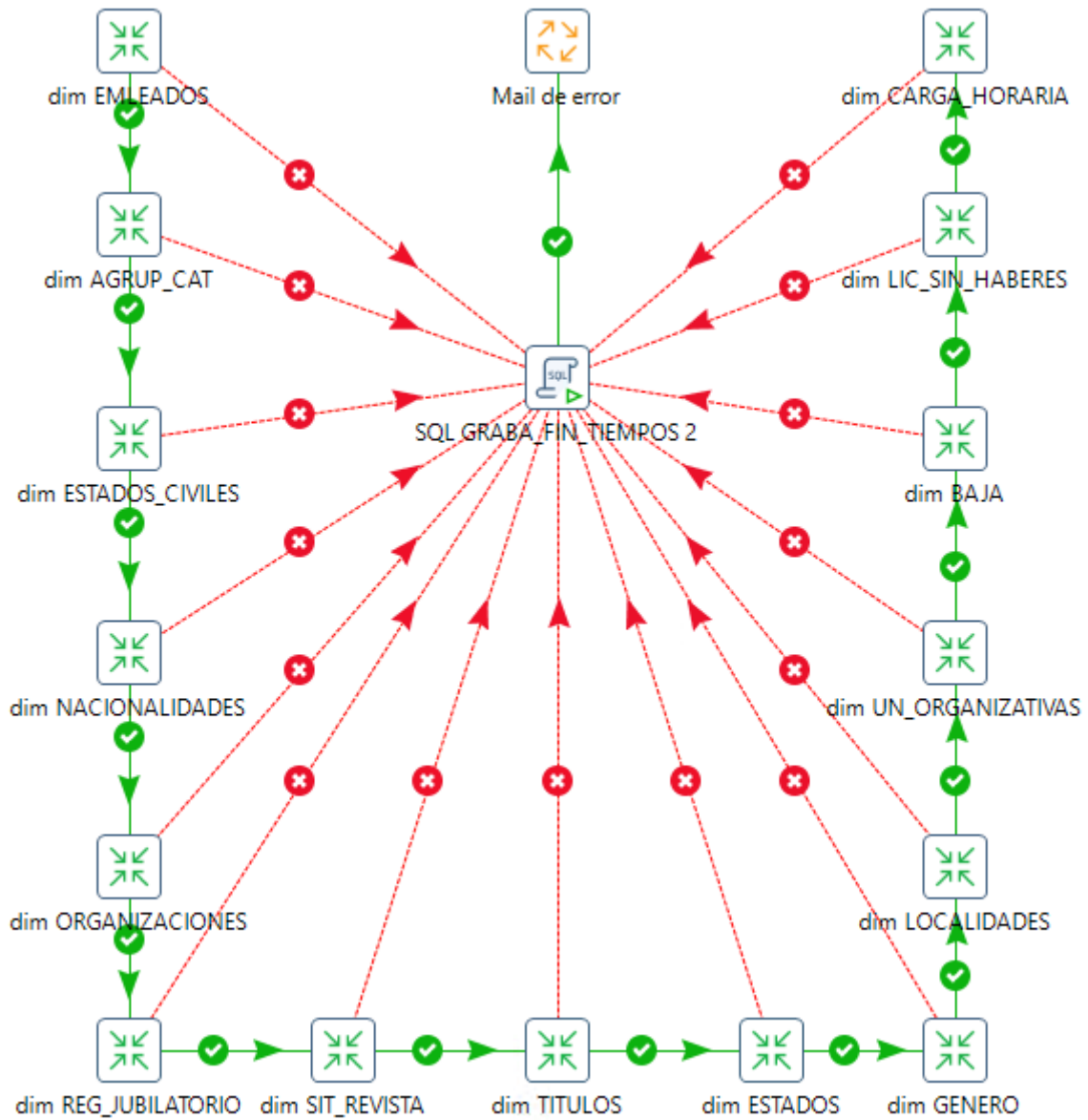
La segunda verificación se hace controlando que efectivamente hubo un archivo transferido. De esta manera si el archivo existe en el servidor, continúa con la siguiente etapa del job.

### **Actualización de dimensiones**

La segunda parte del job es la que carga y actualiza todas las dimensiones que se utilizan en el cubo, esto se realiza desde un step del job principal que invoca a otro job denominado `actualiza_dimensiones.kjb`. Este último, se encarga de ejecutar todas las transformaciones que realizan esta tarea de actualización. En la Figura 4.11 se puede observar dicho proceso.

Las transformaciones de actualización pueden tener dos tipos de estructuras, dependiendo si la información se obtiene desde la base operacional LEDGR o desde la base de datos Planta de Personal.

Internamente, todas las transformaciones que se actualizan desde la BD operacional LEDGR, tienen la misma estructura, y se puede observar en la Figura 4.12.



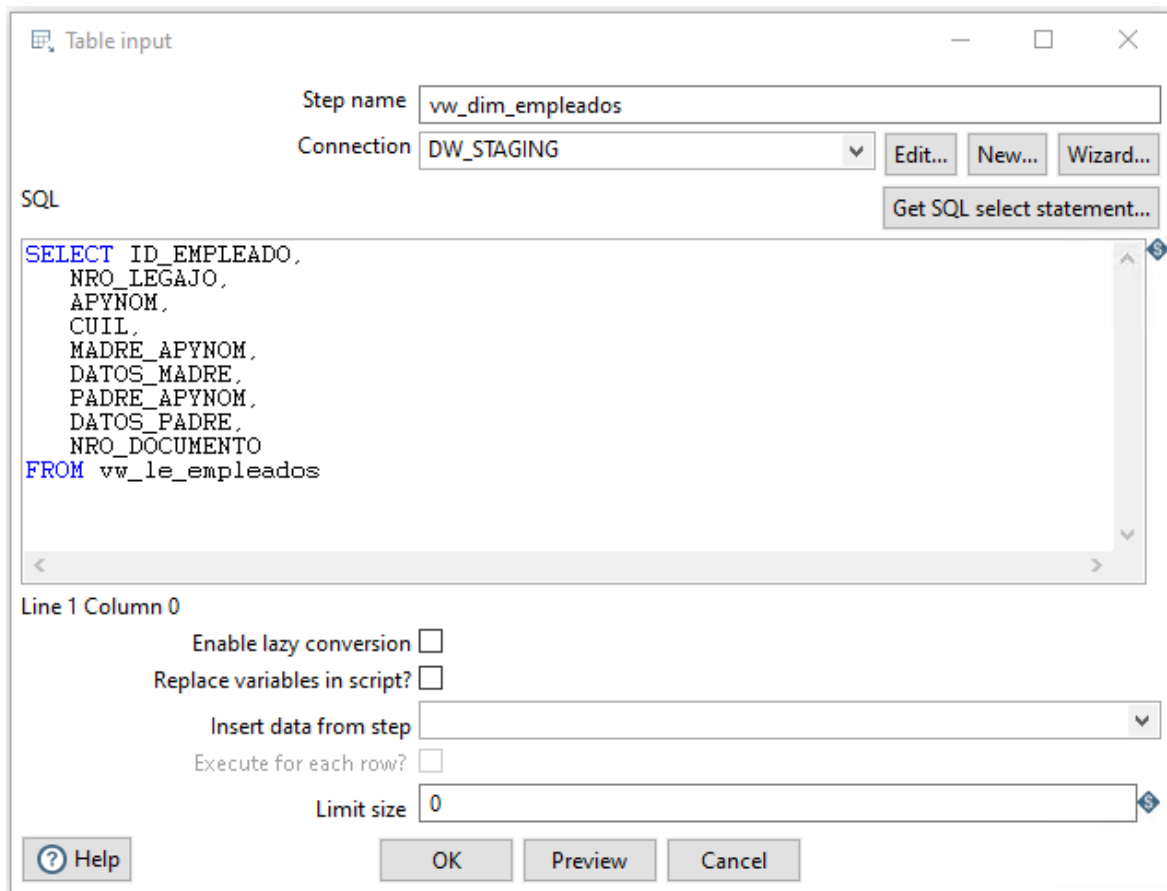
**Fig.4.11.** Estructura del job actualiza\_dimensiones.kjb



**Fig.4.12.** Estructura de una transformación de carga de dimensión desde la base de datos operacional

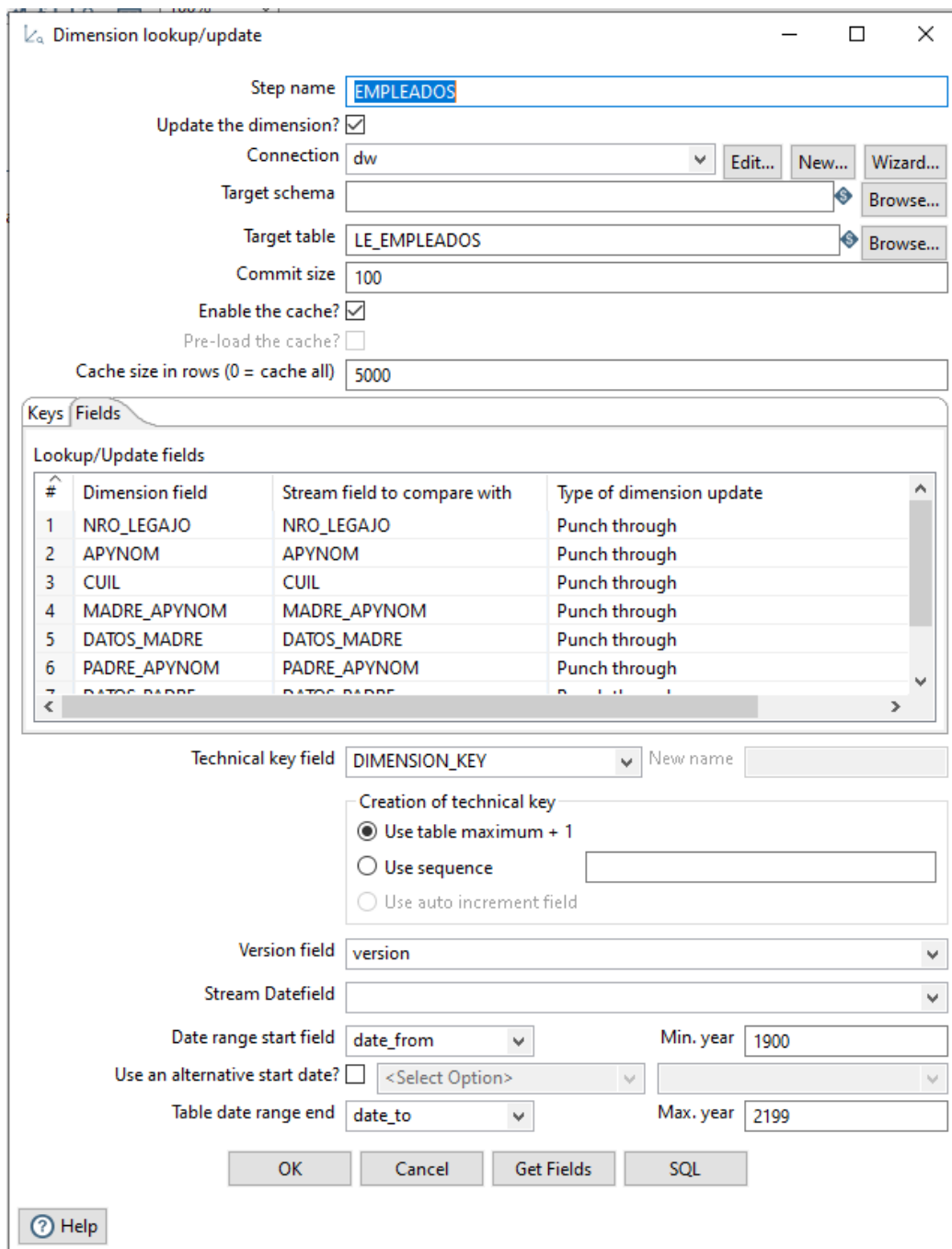
Para estas cargas de dimensiones, en primera instancia con el step 'Table input' se accede a las vistas creadas en la Staging Area para obtener los datos del

legajo electrónico. De esta manera, con una sentencia de SELECT de SQL se selecciona la información que se desea obtener de la vista (Figura 4.13).



**Fig.4.13.** Sentencia SQL en el paso Table Input de PDI.

Una vez que se obtiene la información desde la base de datos, el proceso continúa con el siguiente paso que se denomina 'Dimension lookup/update' (Figura 4.14) que permite implementar los tipos de dimensiones lentamente cambiantes dando la posibilidad de actualizar la dimensión o realizar una búsqueda. En este caso se utiliza en modo actualización por lo que la dimensión se actualiza cada vez que se ejecuta la transformación.



**Fig.4.14.** Paso de actualización de dimensión.

Como se mencionó precedentemente en este caso se utiliza la opción Punch through. Esto implica que en caso que haya un nuevo dato para cargar en la dimensión, se agrega como un nuevo registro; en cambio si un dato debe ser modificado porque sufrió un cambio en algún valor en la fuente de datos de origen, directamente en la dimensión se modifica la fila anterior, es decir, se reemplaza el valor que tenía.

Para el caso de las dimensiones que se cargan desde la base de datos Planta de Personal, el proceso para actualizarlas se describe a continuación y se muestra en la Figura 4.15.



**Fig.4.15.** Actualización de una dimensión desde la base de datos Planta de Personal.

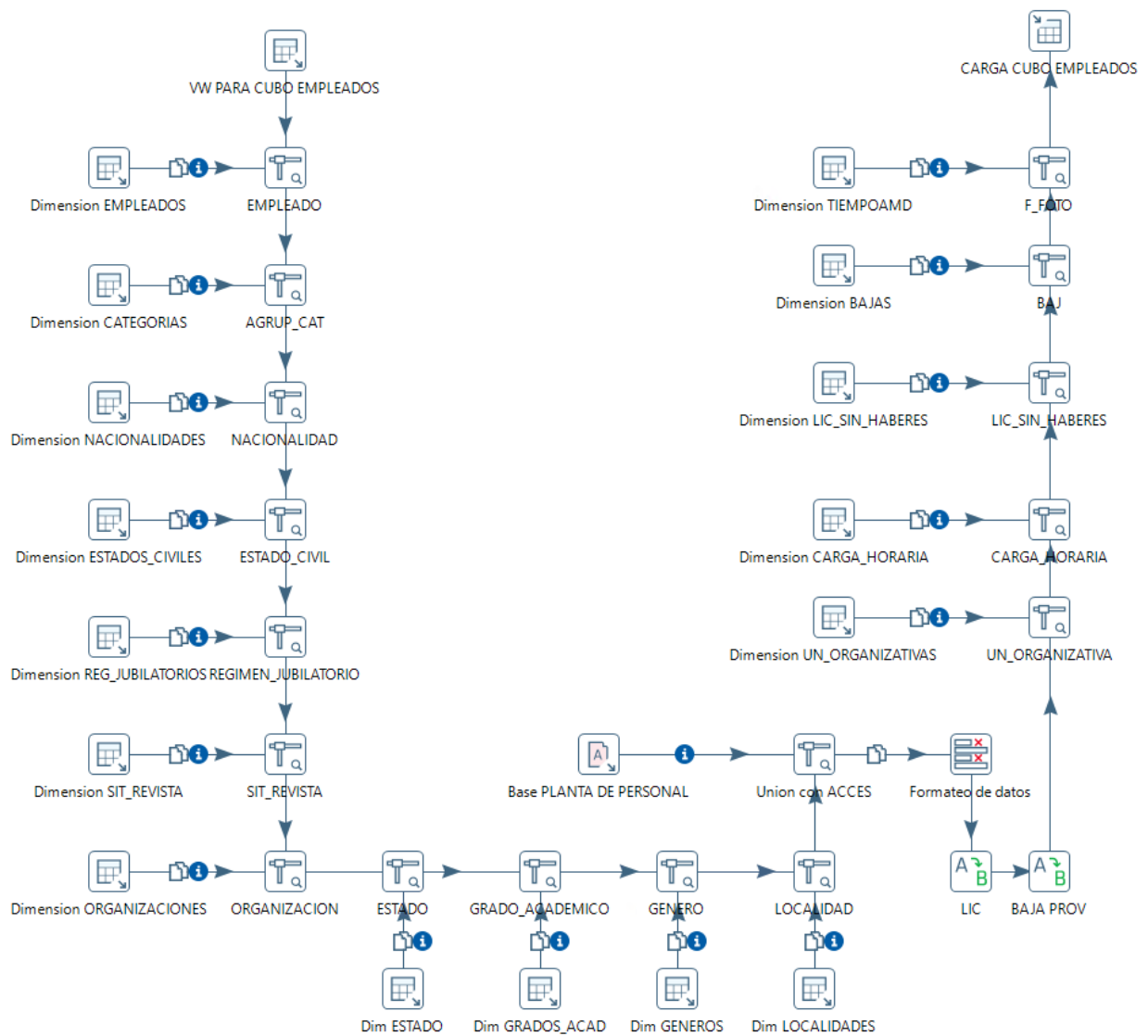
En primer lugar, se utiliza el step Microsoft Access Input que obtiene los datos solicitados de la base Planta de Personal. Si se toma como ejemplo la carga de la dimensión Unidades Organizativas desde el step mencionado se toman los siguientes datos: `codigo_uni_org` y `unidad_organizativa`, que se corresponden, el primero con el código identificador del dato y el segundo con la descripción del mismo.

Luego de esto se ordenan los datos, siguiendo con el ejemplo, el ordenamiento es de forma ascendente y por el dato `codigo_uni_org`. Esto se realiza para poder filtrar la información en el siguiente step: Unique Rows, que en caso que esté repetido el valor elimina del flujo ese dato dejando de esta manera una sola fila para cada una de las unidades organizativas.

Finalmente, con el step Dimension Lookup/Update se realiza la carga y actualización de los datos, utilizando en este caso también la opción Punch through para sobrescribir los valores en caso que se haya modificado algún dato en el origen.

### **Carga de la información en el cubo**

La última etapa del job principal es la que se encarga de la actualización del cubo tomando los datos de las fuentes e insertando la información en la tabla destinada para ello. Toda esta tarea se realiza con una transformación que es invocada para su ejecución desde el job principal, y se denomina `prod_carga_empleados.ktr`. Su estructura se puede visualizar en la Figura 4.16.



**Fig.4.16.** Estructura de la transformación de carga (prod\_carga\_empleados.ktr)

Dentro de esta transformación de carga, el primer paso que se realiza es la lectura de una vista creada en el área de Staging denominada VW\_PARA\_CUBO\_EMPLEADOS. La misma contiene información sobre las medidas y todas las claves foráneas necesarias para que se pueda relacionar a cada empleado con las dimensiones existentes.

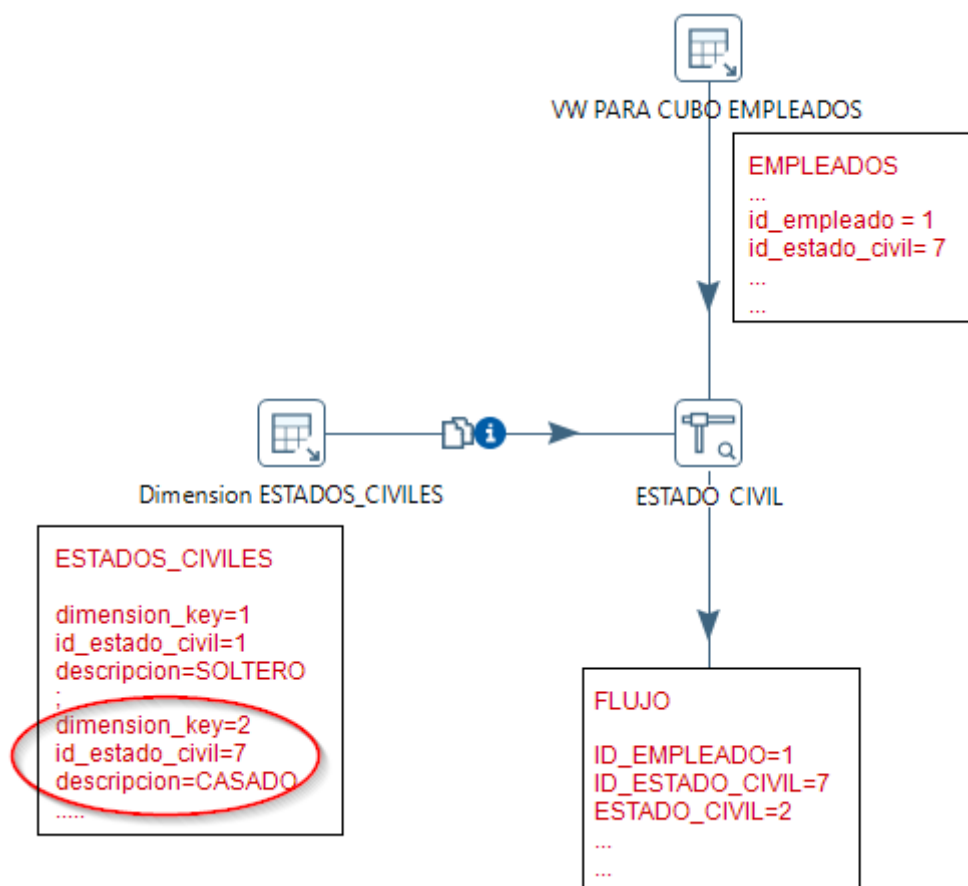
Para poder cruzar los datos de la vista del área de staging con cada una de las dimensiones se utilizan dos pasos que son: Table Input y Stream lookup.

Con el Table Input se obtienen todos los posibles valores de la dimensión en cuestión.

Con el paso Stream lookup se realiza el cruce de información entre los datos del flujo principal y los de la dimensión que se extraen con el step anterior. Un



ejemplo de este cruce es con la dimensión Estado\_Civil y se puede ver gráficamente en la Figura 4.17. En este caso, desde la vista para\_cubo\_empleados se obtiene el listado de empleados, cada uno con el ID del Estado Civil que le corresponde entre otros datos. Por otro lado, desde el Table Input se obtienen todos los Estados Civiles posibles junto con el ID y el dato de DIMENSION\_KEY (PK) almacenados en la tabla de la dimensión correspondiente. Teniendo esa información, con el Stream Lookup se cruzan por un lado los datos del empleado incluyendo el ID del Estado Civil, con los datos que provienen de la dimensión.

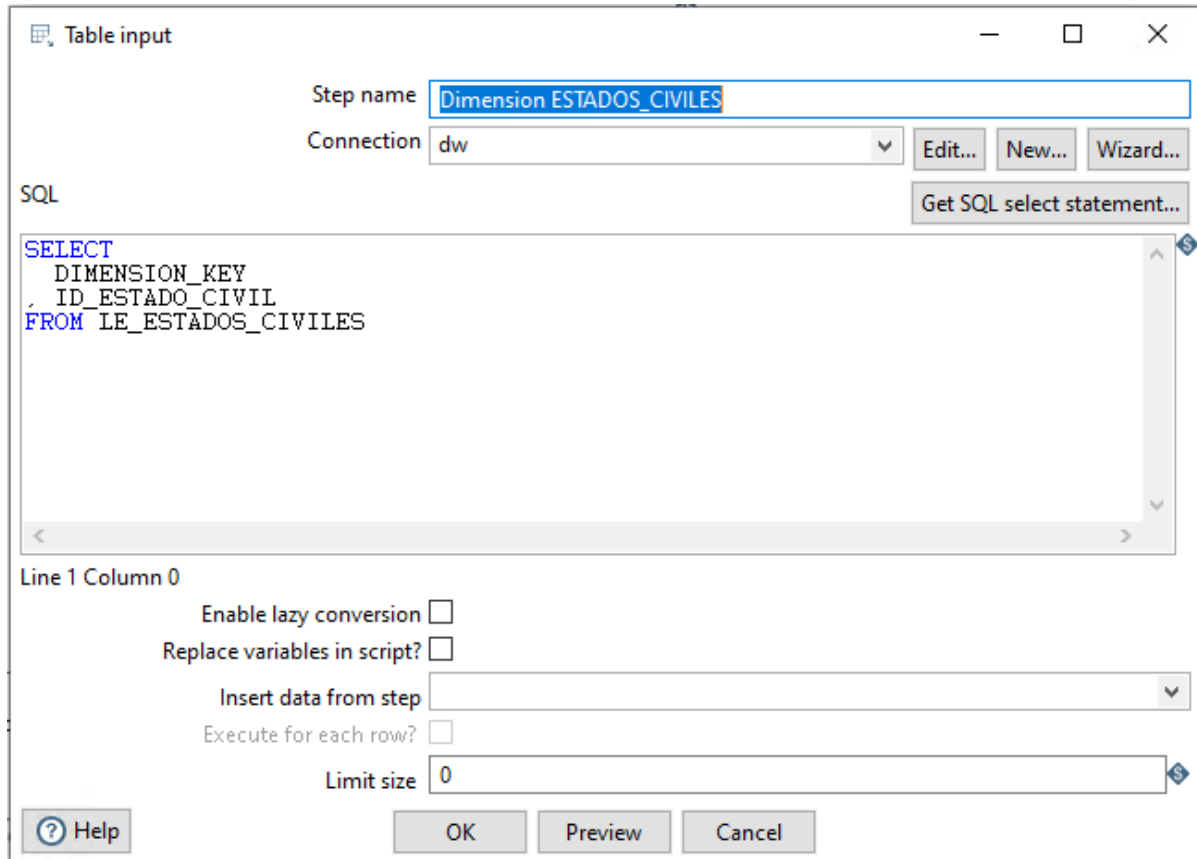


**Fig.4.17.** Ejemplo de cruce de información entre el flujo principal y una dimensión

Una vez finalizado el cruce, el flujo de información principal continúa con los datos que ya traía, y además se agrega el valor de la DIMENSION\_KEY que le corresponde al empleado.

Cada Table Input de esta serie de pasos para cruzar la información con las dimensiones realiza una consulta SQL sobre las tablas de dimensiones creadas y

obtiene todos los valores posibles de la misma. Siguiendo con el ejemplo anterior, para la dimensión Estado Civil se obtiene la lista de posibles estados civiles guardados en la dimensión de la forma que muestra la Figura 4.18.



**Fig.4.18.** Paso de Table Input para obtener los valores de la dimensión Estados Civiles

Finalmente, mediante el paso Stream Lookup se hace el cruce de información descrito anteriormente. Entonces al flujo principal de datos se le suma el valor del identificador correspondiente de la dimensión Estado Civil del Data Mart. Se puede observar la estructura de este paso en la Figura 4.19.

Al terminar el cruce de datos del flujo principal con todas las dimensiones relacionadas a la información que se extrae de la base de datos operacional, se realiza la unión de los datos con los que provienen de la segunda fuente origen de información, la base en Access Planta de Personal. Para ello se utiliza un paso de la herramienta PDI llamado Access Input, que busca un archivo específico en el servidor y extrae del mismo la información que se necesite.

Stream lookup

Step name: ESTADO\_CIVIL

Lookup step: Dimension ESTADOS\_CIVILES

The key(s) to look up the value(s):

#	Field	LookupField
1	ID_ESTADO_CIVIL	ID_ESTADO_CIVIL

Specify the fields to retrieve :

#	Field	New name	Default	Type
1	DIMENSION_KEY	ESTADO_CIVIL	0	Integer

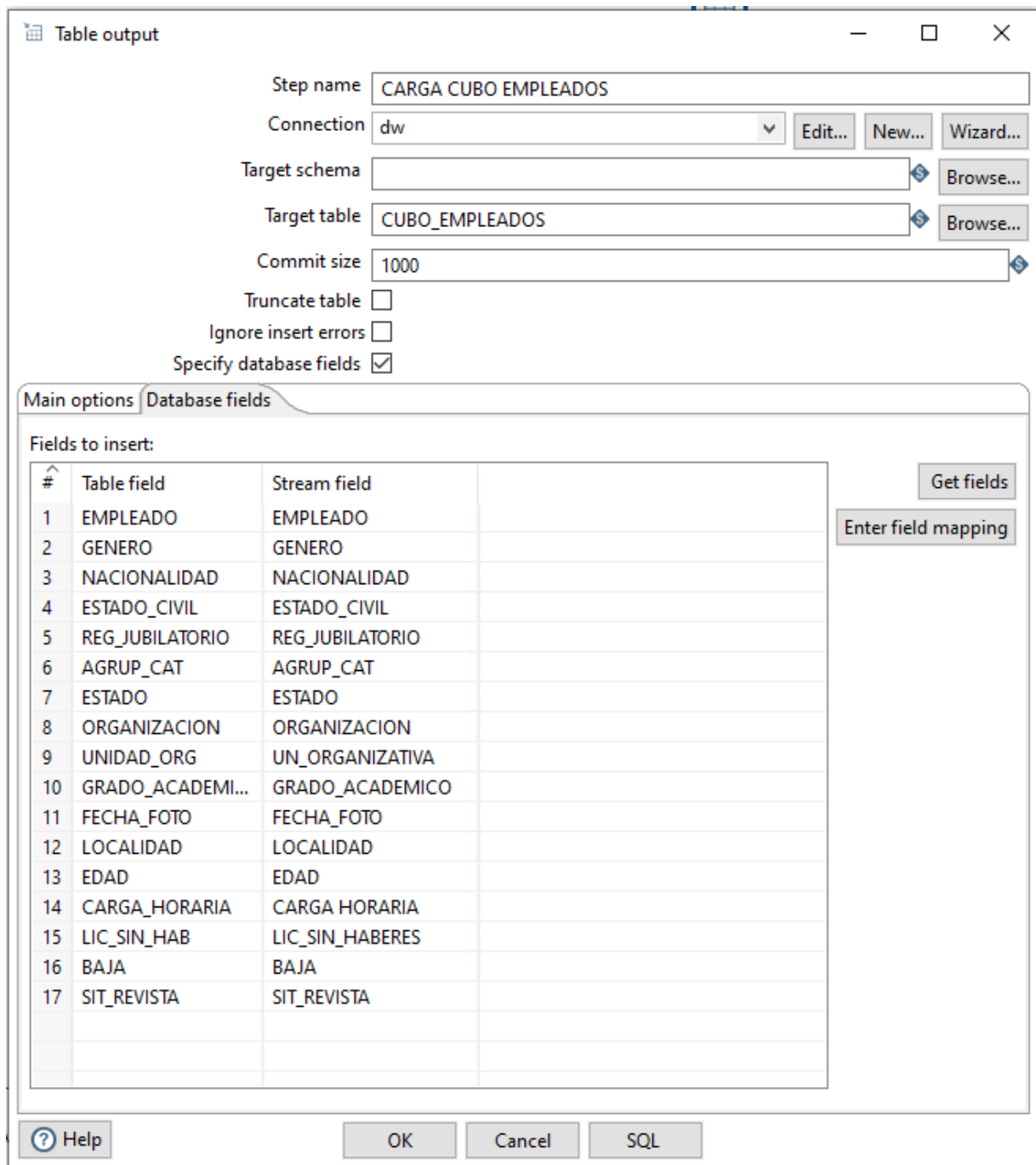
Preserve memory (costs CPU)  
 Key and value are exactly one integer  
 Use sorted list (i.s.o. hashtable)

**Fig.4.19.** Paso de Stream lookup

Para unir al flujo de información principal, la información que se extrae desde este archivo Access, se utiliza un step de Stream Lookup y el cruce se realiza mediante un campo que ambos flujos tienen en común que es el CUIT/CUIL del empleado.

La transformación de carga continúa estandarizando algunos datos y se los formatea para unificarlos. Para cruzar la información obtenida desde la estructura Planta de Personal con las dimensiones que se cargan a partir de ese mismo origen, se hace de la misma manera que las anteriores, mediante el step Table Input y el de Stream Lookup.

Finalmente, luego de transformar y estandarizar todos los datos se procede con la carga de la tabla de hechos donde es almacenada la información dentro del DW. Esto se realiza mediante el paso Table Output y se puede visualizar en la Figura 4.20.



**Fig.4.20.** Paso de Table Output para insertar la información en la tabla de hechos.

La tabla donde se insertan estos datos se denomina CUBO\_EMPLEADOS y está alojada en el área de trabajo del DW junto a las dimensiones y demás tablas de hechos creadas para los distintos DM. Esta tabla en este caso contiene los hechos y el valor a de la DIMENSION\_KEY de cada una de las dimensiones.

Dado que la información se guarda en formato de fotos, no se realiza un borrado de los datos de la tabla antes de comenzar, sino que directamente se procede a la inserción de la información.

#### **4.5.5.3. Definición del cubo y análisis**

Al finalizar con el proceso ETL, se continúa con la última etapa de la creación del DM que consiste en definir el cubo, sus atributos y la estructura con la que podrá ser utilizado en la herramienta de usuario final.

Esto se realiza con la herramienta de Pentaho, Schema Workbench. Con ella se define el cubo para permitir la explotación de la información del DM junto a toda la documentación necesaria.

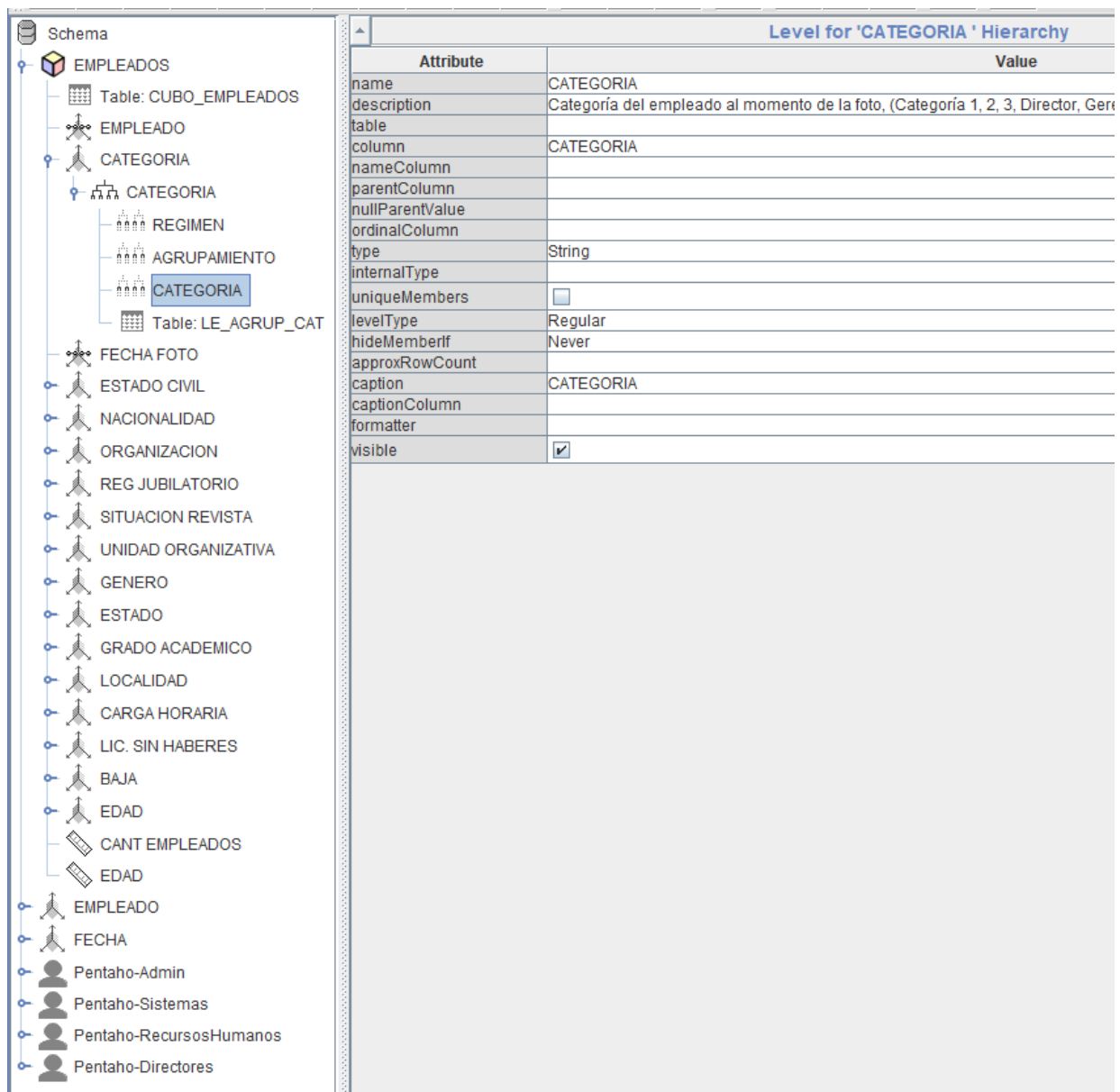
Con la herramienta lo que se termina creando es un archivo xml que es leído e interpretado por el motor Mondrian de Pentaho para el análisis. El archivo completo de definición del cubo utilizado se ha transcrito en el Anexo A.

Desde la herramienta se realiza una conexión a la base de datos del DW, para permitirle acceder a la tabla de hechos principal y a la de todas las dimensiones utilizadas.

Es en este punto donde se arman las jerarquías que tienen definidas las dimensiones, los nombres con los que se muestra al usuario final, el formato de las medidas, su tipo de dato, la forma de agrupamiento (por ejemplo suma o promedio), etc. También es donde se plasma la auto documentación del cubo, tanto la descripción general como la de cada una de las medidas y dimensiones. Esto permite a los usuarios finales desde la herramienta web poder visualizar toda la información necesaria sobre los datos que tienen disponibles sin necesidad de buscarlos en otro lugar. Parte de estas definiciones desde la herramienta se pueden ver en la Figura 4.21.

En última instancia se asignan los permisos del cubo a los usuarios o grupos para que puedan visualizar la información desde la herramienta web.

Desde la herramienta Schema Workbench se publica directamente el archivo con toda la información del nuevo cubo al servidor donde se encuentra el resto de los Data Marts de la ARTRN.

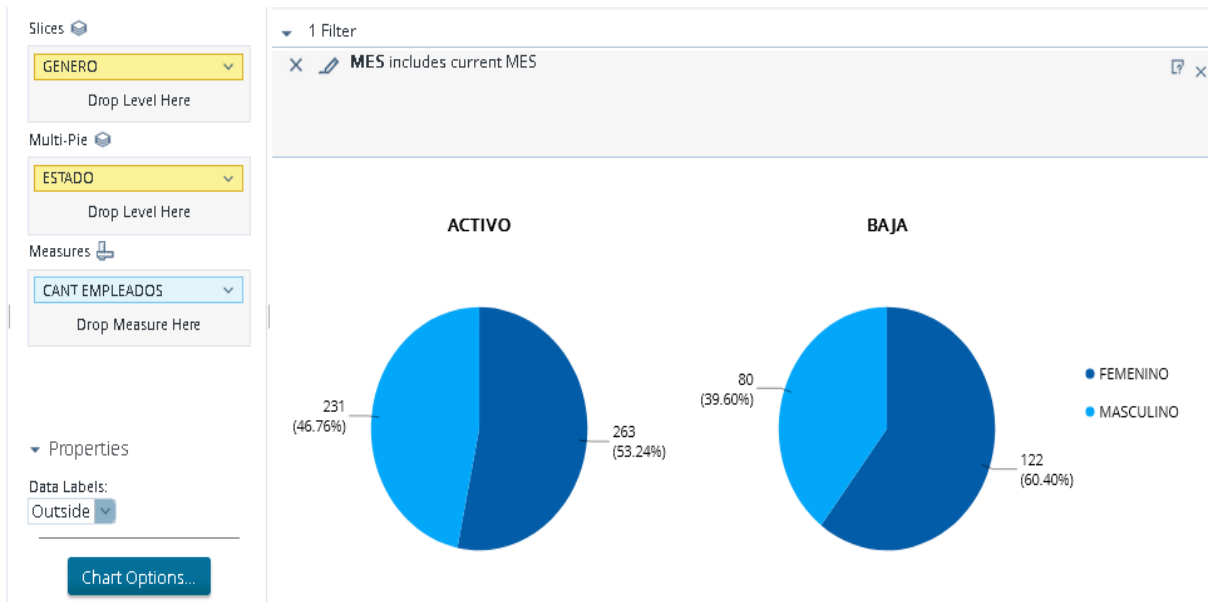


**Fig.4.21.** Estructura del cubo Empleados definida con Schema Workbench

#### 4.5.6. Visualización del cubo desde la herramienta de usuario final

Para poder dar solución a las necesidades del área de RRHH, se crearon diferentes reportes en Pentaho que responden a los objetivos identificados en las reuniones. Ejemplos de estos casos son:

Ver el número de empleados por género: El reporte que se ilustra en la Figura 4.22. muestra dos gráficos, uno para los agentes activos y otro para lo que tienen una baja, junto a la cantidad de empleados por género de la ARTRN.



**Fig.4.22.** Reporte de empleados por género y estado

Ver promedio de edades de los empleados por departamento o área: el reporte que se muestra en la Figura 4.23., muestra la cantidad de empleados por unidades organizativas junto al promedio de edades de los mismos.

UNIDAD ORG PTA	EDAD	CANT EMPLEADOS
Not Available	45.429	7
ALLEN (Subdelegación)	48.667	6
ASESORIA TECNICA	54	2
AUDITORIA INTERNA	37.5	6
BAHIA BLANCA (Subdelegación)	62.667	3
BARILOCHE (Receptoría)	49	2
CAPITAL FEDERAL (Delegación Zonal)	50	6
CATRIEL (Subdelegación)	56.5	4
CERVANTES (Receptoría)	60	1
CHICHINALES (Receptoría)	46	1
CHIMPAY - CNEL. BELISLE (Receptoría)	53	2
CHDÉLE CHDÉL (Delegación Zonal)	48.5	4
CINCO SALTOS (Subdelegación)	43.125	8
CIPOLLETTI (Delegación Zonal)	46.909	22
COMALLO (Receptoría)	54	1
DINA HUAPI (Subdelegación)	55	2
EL BOLSON (Subdelegación)	56	4
FERNANDEZ ORD (Receptoría)	38	1
GENERAL CONESA (Receptoría)	55	2
GENERAL GODOY (Receptoría)	57	1
GENERAL ROCA (Delegación Zonal)	51.821	28
GERENCIA DE ADMINISTRACION	47.414	29
GERENCIA DE ASUNTOS LEGALES	43.932	44
GERENCIA DE FACTOS	47.881	41

**Fig.4.23.** Promedio de edades por Unidad Organizativa

Con la herramienta de Pentaho Dashboard que brinda la interfaz web, se crea también un Tablero de Control con una serie de reportes para visualizar de manera más amigable cierta información, en este caso, se arman dos reportes de tablas que muestran la cantidad de empleados en condición de jubilarse, uno para cada género, y dos gráfico que indican: uno la cantidad de agentes activos por Agrupamiento y otro que muestra la evolución de la planta del personal. Esto se puede ver en la Figura 4.24.

Estos Tableros de Control se crean juntando dos o más reportes ya guardados y se muestran todos juntos en una misma ventana para poder visualizar toda la información junta.

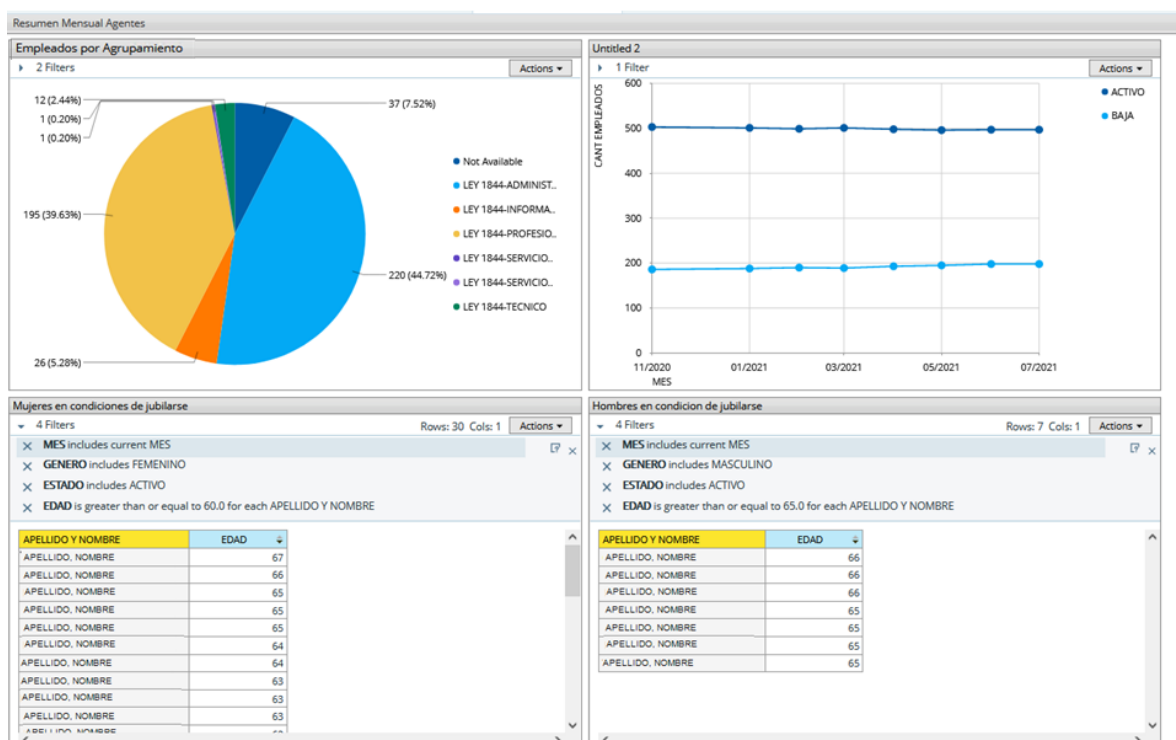


Fig.4.24. Tablero de control

#### 4.5.7. Programación y actualización del cubo

La programación del cubo se plantea de manera mensual ya que se guarda una foto de la información con dicha periodicidad. Para esto se crea en el servidor de Pentaho un script que contiene los comandos necesarios para la ejecución del mismo. Este archivo ejecuta la herramienta Kitchen de Pentaho, que es la encargada de ejecutar los jobs y le pasa como parámetro el archivo a ejecutar, en este caso el job principal job\_carga\_cubo\_empleados.kjb. Además, el script se



encarga de generar archivos de logs que son guardados para su revisión en caso que ocurra algún fallo durante la ejecución del proceso. El script mencionado se encuentra transcrito completo en el Anexo B.

Los cubos existentes en el DW se encuentran programados con el administrador de tareas CRON del servidor Linux donde se encuentra instalado Pentaho. A la lista de estas programaciones se sumó la del nuevo cubo, quedando su ejecución programada para el primer día de cada mes 19:00 hs. Esto se puede visualizar en la Figura 4.25.

```
#  
# ==> Cubo de fotos de RRHH de empleados de la agencia. El 1 de cada mes 19hs  
00 19 1 * * /home/pentaho/util/etl-empleados.sh  
#
```

**Fig.4.25.** Programación del script que ejecuta el job ETL

## 5. VERIFICACIÓN Y ANÁLISIS DE LOS RESULTADOS

En este capítulo se describe cómo se llevan a cabo las pruebas correspondientes para la verificación del resultado final obtenido en el capítulo anterior.

Las pruebas que se realizan consisten en comprobar la calidad del proceso ETL, lo que se lleva a cabo mediante casos de pruebas sobre datos de origen, pruebas sobre el mismo proceso ETL y sobre los datos de salida o finales.

También se realizan pruebas desde el cubo de análisis mediante la herramienta que utilizará el usuario para corroborar los datos cargados.

Los casos de prueba fueron analizados en un ambiente de testing donde se encuentra instalada una base de datos origen desde donde se tomaron los datos, llamada LEDGR, una base de datos Intermedia denominada DW\_STAGING y una BD con un DW con información de prueba que tiene el nombre DW. Todas las estructuras de estas bases de datos están iguales que en producción. Además se cuenta con un servidor de Pentaho también de pruebas que accede y consume los datos del DW del ambiente de test.

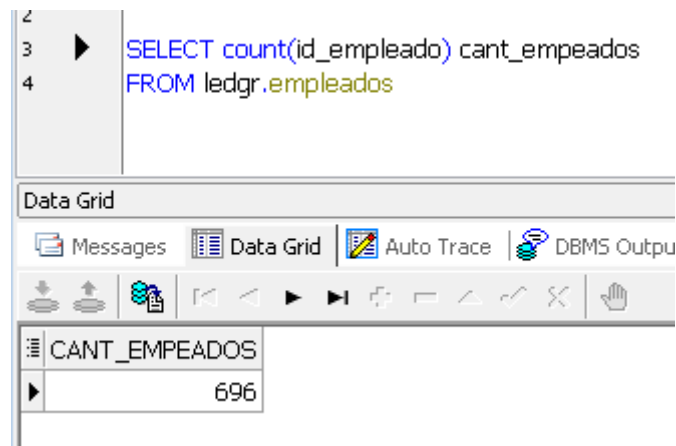
### 5.1. Casos de prueba

#### 5.1.1. Caso de prueba 1

- Objeto a analizar: Proceso de extracción de datos desde el área intermedia.
- Objetivo: Verificar que los datos que se obtienen en el área intermedia sean correctos.
- Datos de entrada: Tablas de la BD del Legajo Digital: EMPLEADOS, REGÍMENES, AGRUPAMIENTOS, CATEGORÍAS, ESTADOS\_CIVILES, NACIONALIDADES, ORGANIZACIONES, REGIMENES\_JUBILATORIOS, LOCALIDADES, GENEROS, SITUACIONES\_DE\_REVISTA y GRADO\_ACADEMICO.
- Funcionalidad a probar: Que las vistas involucradas en el proceso de extracción estén obteniendo los datos correctos desde la BD fuente.

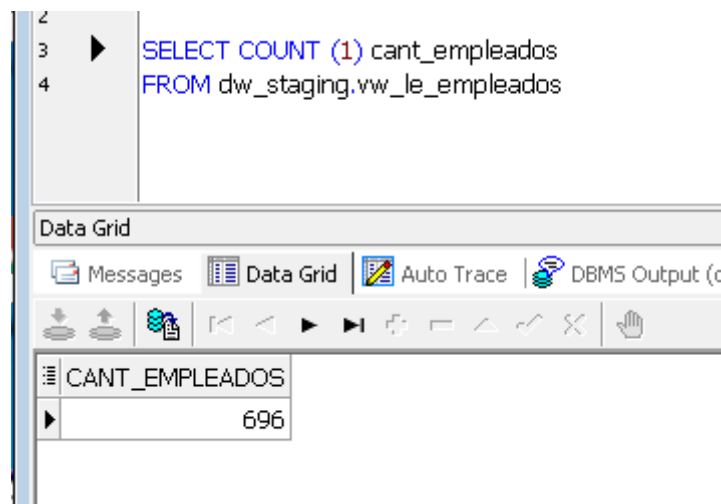
En primera instancia se analiza la vista VW\_LE\_EMPLEADOS: Esta vista toma los datos de la tabla denominada EMPLEADOS del esquema LEDGR (base de datos del aplicativo de legajo electrónico). Realizando una consulta a esta tabla se

obtiene que hay un total de 696 empleados de la agencia cargados, resultado que se puede observar en la Figura 5.1.



**Fig.5.1.** Cantidad de empleados en la base origen

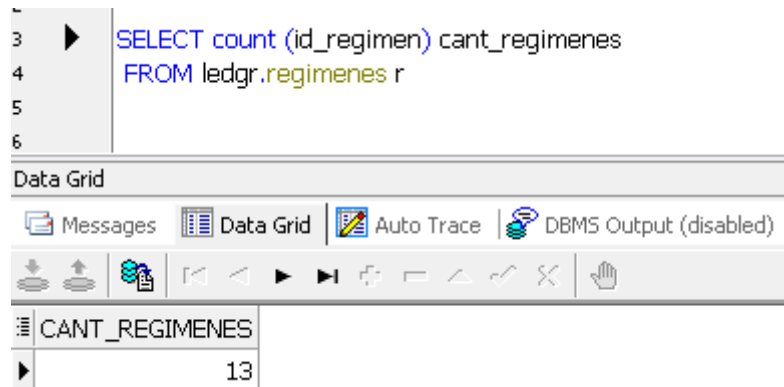
Se realiza en el área de staging una consulta similar para contar la cantidad de registros que se obtienen al ejecutar la vista, entendiendo que cada registro de la misma se corresponde con los datos de un empleado. El resultado de dicha consulta se puede visualizar en la Figura 5.2



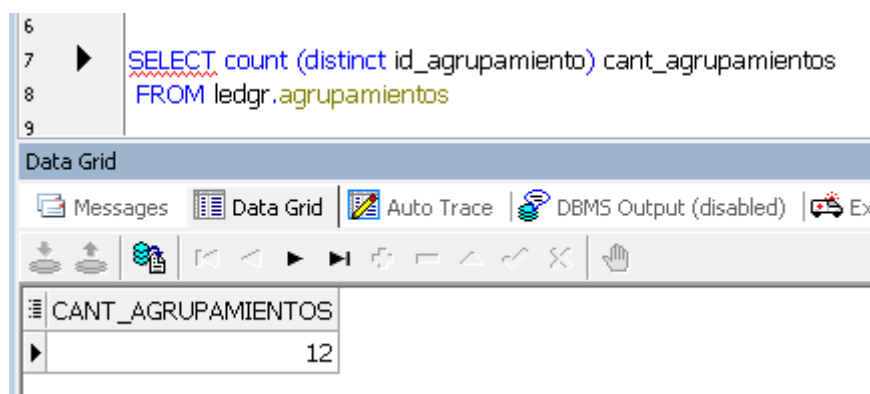
**Fig.5.2.** Cantidad de registros de empleados que se obtienen con la vista de la Staging.

Otra de las vistas analizadas es la que se denomina VW\_LE\_AGRUP\_CAT del área de staging, que se arma obteniendo los datos de 3 tablas de la BD del aplicativo de legajo electrónico estas son: REGÍMENES, AGRUPAMIENTOS y CATEGORÍAS.

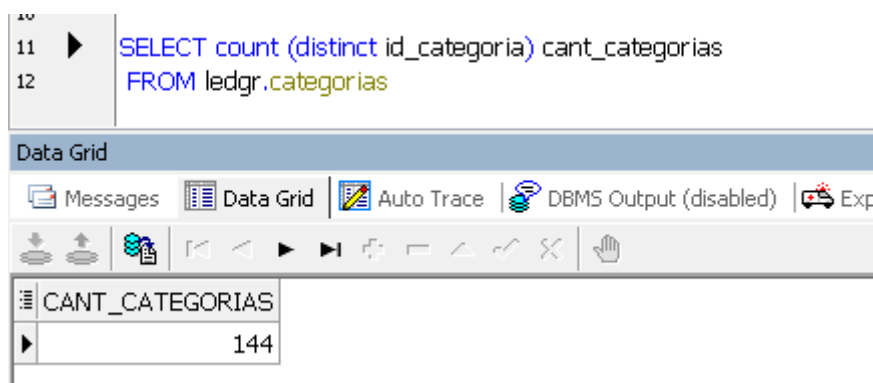
Para controlar esos datos, se realizan consultas contando la cantidad de registros en cada una de esas tablas, como se observa en las Figuras 5.3, 5.4 y en la Figura 5.5.



**Fig.5.3.** Cantidad de regímenes cargados en la base de datos origen



**Fig.5.4.** Cantidad de agrupamientos cargados en la base de datos origen

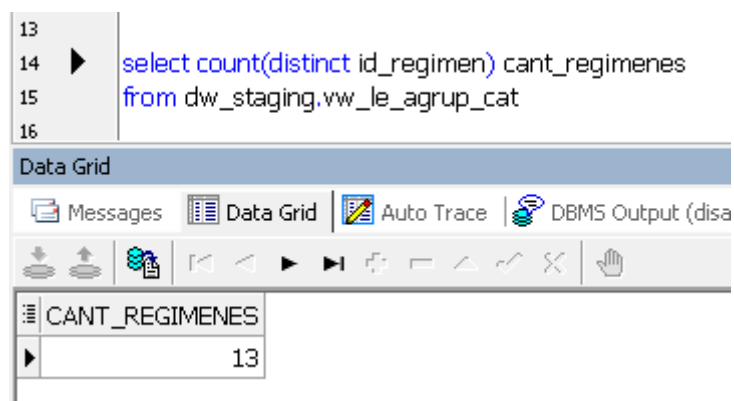


**Fig.5.5.** Cantidad de categorías cargadas en la base de datos origen

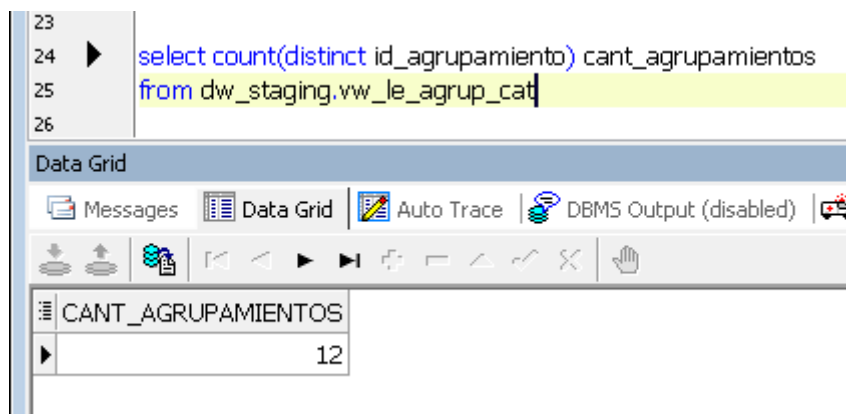
En el área intermedia, la vista VW\_LE\_AGRUP\_CAT devuelve un total de 1884 filas. Esto es porque cada registro es una combinación de regímenes, agrupamientos y categorías. Entonces, para un régimen, hay tantas filas como

combinaciones de agrupamientos y categorías posibles. Por ejemplo, para el Régimen: LEY1844, hay 408 filas. De esa selección, 34 filas tienen como valor Agrupamiento: LEY 1844-INFORMATICOS; y a su vez, en un nivel más bajo, cada uno de esos 34 registros pertenecen a una categoría distinta cada uno.

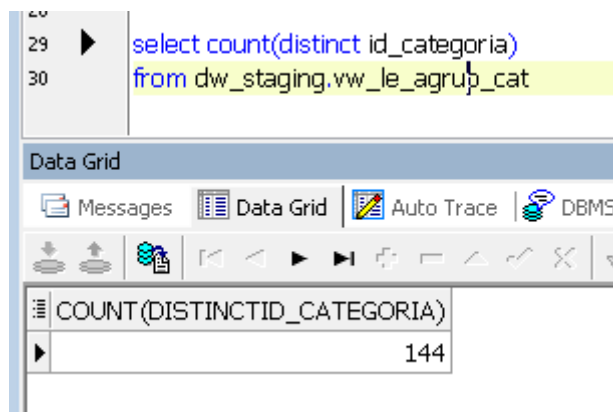
Para poder controlar entonces la cantidad de filas que se obtienen desde la base de origen se realizan tres consultas contando la cantidad de filas distintas para cada uno de los casos, de la manera que lo indican las ilustraciones a continuación. (Fig. 5.6, Fig 5.7 y Fig 5.8).



**Fig.5.6.** Cantidad de regímenes distintos en la vista del área intermedia



**Fig.5.7.** Cantidad de agrupamientos distintos en la vista del área intermedia



**Fig.5.8.** Cantidad de categorías distintas en la vista del área intermedia

Para las demás tablas y vistas, se realizaron los mismos pasos para contar los datos que se obtienen desde las tablas de la Base de Datos origen y desde las vistas del área de Staging. El procedimiento consiste en ejecutar una consulta en SQL que cuente la cantidad de registros de una tabla origen y otra consulta que cuente los resultados que devuelve la vista del área intermedia.

A continuación, en la Tabla 5.1, se presentan los datos obtenidos con dichas sentencias SQL:

Tablas de Base de Datos LEDGR	Cantidad de registros	Vista creada en el área de Staging	Cantidad de registros
ESTADOS_CIVILES	8	VW_LE_ESTADOS_CIVILES	8
NACIONALIDADES	7	VW_LE_NACIONALIDADES	7
ORGANIZACIONES	235	VW_LE_ORGANIZACIONES	235
REGIMENES_JUBILATO RIOS	7	VW_LE_REG_JUBILATORIO S	7
SITUACIONES_DE_REVI STA	4	VW_LE_SIT_REVISTA	4
GRADOS_ACADEMICOS	632	VW_LE_ GRADOS_ACADEMICOS	632

LOCALIDAD	126	VW_LE_LOCALIDADES	126
-----------	-----	-------------------	-----

**Tabla 5.1.** Comparación entre la cantidad de datos de la base de datos de origen y lo obtenido en el área intermedia.

En el caso de ESTADOS y GÉNEROS, los datos que se extraen desde la fuente origen no provienen de tablas particulares creadas para guardar dicha información, sino que son atributos de la tabla LEDGR.EMPLEADOS. Por lo tanto, las vistas creadas para obtener dichos datos realizan una sentencia SQL consultando toda esta tabla, agrupando y seleccionando sólo los valores distintos.

Para la dimensión GÉNERO, de esta forma se obtienen 2 valores desde la base de datos origen y la misma cantidad es la que se recupera desde la vista del área de Staging (VW\_LE\_GENEROS). En el caso de ESTADOS, se cuenta la misma cantidad (dos registros) en la tabla LEDGR.EMPLEADOS que desde el área intermedia con la vista VW\_LE\_ESTADOS.

### 5.1.2 Caso de prueba 2

- Objeto a analizar: Proceso de carga de las dimensiones desde el área intermedia al área del DW
- Objetivo: Verificar que los datos que se obtienen con las vistas desde el área intermedia se inserten de manera correcta y completa en las tablas correspondientes.
- Funcionalidad a probar: Que los procesos que se ejecutan desde la herramienta PDI mediante las transformaciones creadas carguen correctamente los datos.

Desde el job principal se invocan a todas las transformaciones de carga de las dimensiones. Al ejecutar cada una de estas dimensiones la herramienta crea un log que indica la cantidad de registros leídos, y grabados en la tabla correspondiente. Este log se puede observar en la Figura 5.9.



**tion Results**

[Execution History](#)
[Step Metrics](#)
[Performance Graph](#)
[Metrics](#)
[Preview data](#)

1/25 11:50:35 - General - Logging plugin type found with ID: CheckpointLogTable  
 1/25 11:50:53 - Carte - Installing timer to purge stale objects after 1440 minutes.  
 1/25 12:02:41 - Spoon - Starting job...  
 1/25 12:02:45 - Spoon - Job has ended.  
 1/25 12:29:23 - Spoon - Using legacy execution engine  
 1/25 12:29:23 - Spoon - Transformation opened.  
 1/25 12:29:23 - Spoon - Launching transformation [dim\_le\_empleados]...  
 1/25 12:29:23 - Spoon - Started the transformation execution.  
 1/25 12:29:24 - Spoon - The transformation has finished!!  
 1/25 15:39:11 - Spoon - Using legacy execution engine  
 1/25 15:39:11 - Spoon - Transformation opened.  
 1/25 15:39:11 - Spoon - Launching transformation [dim\_le\_estados\_civiles]...  
 1/25 15:39:11 - Spoon - Started the transformation execution.  
 1/25 15:39:11 - dim\_le\_estados\_civiles - Dispatching started for transformation [dim\_le\_estados\_civiles]  
 1/25 15:39:11 - vw\_dim\_estados\_civiles.0 - Finished reading query, closing connection.  
 1/25 15:39:11 - vw\_dim\_estados\_civiles.0 - Finished processing (I=8, O=0, R=0, W=8, U=0, E=0)  
 1/25 15:39:11 - ESTADOS\_CIVILES.0 - Finished processing (I=8, O=8, R=8, W=8, U=0, E=0)  
 1/25 15:39:11 - Spoon - The transformation has finished!!

**Fig.5.9.** Pasos y log resultante al ejecutar la carga de la dimensión Empleados

Este proceso de validación en la carga se realizó con cada una de las transformaciones involucradas en el job principal con este fin. Además del log de la dimensión (como observa en la Figura 5.10), desde la herramienta PDI se pueden ver los datos que se cargaron en la tabla de la dimensión como está ilustrado en la Figura 5.11.





### tion Results

Logging Execution History Step Metrics Performance Graph Metrics Preview data

9/25 11:50:35 - General - Logging plugin type found with ID: CheckpointLogTable  
 9/25 11:50:53 - Carte - Installing timer to purge stale objects after 1440 minutes.  
 9/25 12:02:41 - Spoon - Starting job...  
 9/25 12:02:45 - Spoon - Job has ended.  
 9/25 12:29:23 - Spoon - Using legacy execution engine  
 9/25 12:29:23 - Spoon - Transformation opened.  
 9/25 12:29:23 - Spoon - Launching transformation [dim\_le\_empleados]...  
 9/25 12:29:23 - Spoon - Started the transformation execution.  
 9/25 12:29:23 - dim\_le\_empleados - Dispatching started for transformation [dim\_le\_empleados]  
 9/25 12:29:24 - vw\_dim\_empleados.0 - Finished reading query, closing connection.  
 9/25 12:29:24 - vw\_dim\_empleados.0 - Finished processing (I= 696 O=0, R=0, W= 696 U=0, E=0)  
 9/25 12:29:24 - EMPLEADOS.0 - Finished processing (I=695, O= 696 R=695, W= 696 U=0, E=0)  
 9/25 12:29:24 - Spoon - The transformation has finished!!

**Fig.5.10.** Log de la transformación de carga de la dimensión Estados Civiles

**Execution Results**

Logging Execution History Step Metrics Performance Graph Metrics Preview data

First rows  Last rows  Off

#	ID_ESTADO_CIVIL	ESTADO_CIVIL
1	1	SOLTERO
2	2	CASADO
3	3	VIUDO
4	4	SEPARADO
5	5	DIVORCIADO
6	6	OTRO
7	7	NO INFORMADO
8	8	CONCUBINO

**Fig.5.11.** Datos cargados en la dimensión Estados Civiles

Este procedimiento se realiza con cada una de las dimensiones propuestas y descritas en el capítulo anterior.

### 5.1.3. Caso de prueba 3

- Objeto a analizar: Flujo en caso de éxito y flujos alternativos del job principal.
- Objetivo: Verificar el comportamiento del job principal en caso que todos los pasos que se ejecuten se hagan de manera correcta y comprobar el correcto comportamiento en caso que alguno de los pasos involucrados finalice con algún error.

Para llevar adelante esta prueba es necesario ejecutar el trabajo principal desde diferentes escenarios. En primer lugar se realiza la ejecución con todos los datos preparados para que el job termine correctamente. En este caso, el proceso finaliza con todas las dimensiones actualizadas y el cubo cargado completamente, resultado que se puede comprobar en el log final. Esto se ve como un ejemplo en la Figura 5.12.

Como segunda alternativa, se procede a ejecutar el job pero sin tener cargado el archivo fuente en Access, desde donde también se saca información para cargar el cubo. Por lo tanto, el paso que verifica la existencia del mismo en el servidor tiene que dar aviso mediante un mail y el flujo debe continuar por el camino alternativo para finalizar el proceso con el error.

El log resultante con el error se puede visualizar en la Figura 5.13.

**Execution Results**

Logging History Job metrics Metrics

```

2020 /09/25 11:50:35 - General - Logging plugin type found with ID: CheckpointLogTable
2020 /09/25 11:50:53 - Carte - Installing timer to purge stale objects after 1440 minutes.
2020 /09/25 12:02:41 - Spoon - Starting job...
2020 /09/25 12:02:42 - job_empleados - Start of job execution
2020 /09/25 12:02:42 - job_empleados - Starting entry [dim AGRUP_CAT]
2020 /09/25 12:02:42 - dim AGRUP_CAT - Using run configuration [Pentaho local]
2020 /09/25 12:02:42 - dim AGRUP_CAT - Using legacy execution engine
2020 /09/25 12:02:42 - dim_le_agrup_cat - Dispatching started for transformation [dim_le_agrup_cat]
2020 /09/25 12:02:42 - vw_dim_agrup_cat.0 - Finished reading query, closing connection.
2020 /09/25 12:02:42 - vw_dim_agrup_cat.0 - Finished processing (I=1878, O=0, R=0, W=1878, U=0, E=0)
2020 /09/25 12:02:43 - AGRUP_CAT.0 - Finished processing (I=1878, O=0, R=1878, W=1878, U=0, E=0)
2020 /09/25 12:02:43 - job_empleados - Starting entry [dim EMLEADOS]
2020 /09/25 12:02:43 - dim EMLEADOS - Using run configuration [Pentaho local]
2020 /09/25 12:02:43 - dim EMLEADOS - Using legacy execution engine
2020 /09/25 12:02:43 - dim_le_empleados - Dispatching started for transformation [dim_le_empleados]
2020 /09/25 12:02:43 - vw_dim_empleados.0 - Finished reading query, closing connection.
2020 /09/25 12:02:43 - vw_dim_empleados.0 - Finished processing (I=695, O=0, R=0, W=695, U=0, E=0)
2020 /09/25 12:02:44 - EMPLEADOS.0 - Finished processing (I=695, O=14, R=695, W=695, U=36, E=0)
2020 /09/25 12:02:44 - job_empleados - Starting entry [dim ESTADOS_CIVILES]
2020 /09/25 12:02:44 - dim ESTADOS_CIVILES - Using run configuration [Pentaho local]
2020 /09/25 12:02:44 - dim ESTADOS_CIVILES - Using legacy execution engine
2020 /09/25 12:02:44 - dim_le_estados_civiles - Dispatching started for transformation [dim_le_estados_civiles]
2020 /09/25 12:02:44 - vw_dim_estados_civiles.0 - Finished reading query, closing connection.
2020 /09/25 12:02:44 - vw_dim_estados_civiles.0 - Finished processing (I=8, O=0, R=0, W=8, U=0, E=0)
2020 /09/25 12:02:44 - ESTADOS_CIVILES.0 - Finished processing (I=8, O=0, R=8, W=8, U=0, E=0)
2020 /09/25 12:02:44 - job_empleados - Starting entry [dim NACIONALIDADES]
2020 /09/25 12:02:44 - dim NACIONALIDADES - Using run configuration [Pentaho local]
2020 /09/25 12:02:44 - dim NACIONALIDADES - Using legacy execution engine
2020 /09/25 12:02:44 - dim_le_nacionalidades - Dispatching started for transformation [dim_le_nacionalidades]
2020 /09/25 12:02:44 - vw_dim_nacionalidades.0 - Finished reading query, closing connection.
2020 /09/25 12:02:44 - vw_dim_nacionalidades.0 - Finished processing (I=7, O=0, R=0, W=7, U=0, E=0)
2020 /09/25 12:02:44 - NACIONALIDADES.0 - Finished processing (I=7, O=0, R=7, W=7, U=0, E=0)
2020 /09/25 12:02:44 - job_empleados - Starting entry [dim ORGANISMOS]
2020 /09/25 12:02:44 - dim ORGANISMOS - Using run configuration [Pentaho local]
2020 /09/25 12:02:44 - dim ORGANISMOS - Using legacy execution engine
2020 /09/25 12:02:44 - dim_le_organismos - Dispatching started for transformation [dim_le_organismos]
2020 /09/25 12:02:44 - vw_dim_organismos.0 - Finished reading query, closing connection.
2020 /09/25 12:02:44 - vw_dim_organismos.0 - Finished processing (I=5, O=0, R=0, W=5, U=0, E=0)
2020 /09/25 12:02:44 - ORGANISMOS.0 - Finished processing (I=5, O=0, R=5, W=5, U=0, E=0)
2020 /09/25 12:02:44 - job_empleados - Starting entry [dim ORGANIZACIONES]
2020 /09/25 12:02:44 - dim ORGANIZACIONES - Using run configuration [Pentaho local]
2020 /09/25 12:02:44 - dim ORGANIZACIONES - Using legacy execution engine
2020 /09/25 12:02:44 - dim_le_organizaciones - Dispatching started for transformation [dim_le_organizaciones]
2020 /09/25 12:02:45 - vw_dim_organizaciones.0 - Finished reading query, closing connection.
2020 /09/25 12:02:45 - vw_dim_organizaciones.0 - Finished processing (I=235, O=0, R=0, W=235, U=0, E=0)
2020 /09/25 12:02:45 - ORGANIZACIONES.0 - Finished processing (I=235, O=16, R=235, W=235, U=46, E=0)
2020 /09/25 12:02:45 - job_empleados - Starting entry [dim REG_JUBILATORIO]
2020 /09/25 12:02:45 - dim REG_JUBILATORIO - Using run configuration [Pentaho local]
2020 /09/25 12:02:45 - dim REG_JUBILATORIO - Using legacy execution engine
2020 /09/25 12:02:45 - dim_le_reg_jubilatorios - Dispatching started for transformation [dim_le_reg_jubilatorios]
2020 /09/25 12:02:45 - vw_dim_reg_jubilatorios.0 - Finished reading query, closing connection.
2020 /09/25 12:02:45 - vw_dim_reg_jubilatorios.0 - Finished processing (I=1, O=0, R=0, W=1, U=0, E=0)
2020 /09/25 12:02:45 - REG_JUBILATORIO.0 - Finished processing (I=1, O=0, R=1, W=1, U=0, E=0)

```

**Fig.5.12.** Log resultante al ejecutar el job principal sin errores

```

2021/05/11 15:16:35 - job_empleados - Start of job execution
log4j:ERROR No output stream or file set for the appender named [pdi-execution-appender].
2021/05/11 15:16:35 - job_empleados - Starting entry [Trae Access del Intranet]
2021/05/11 15:16:35 - Trae Access del Intranet - Started FTP job to intranet1
Sep 26, 2021 3:16:35 PM org.apache.cxf.endpoint.ServerImpl initDestination
INFO: Setting the server's publish address to be /lineage
2021/05/11 15:16:35 - job_empleados - Starting entry [Trajo algo el FTP?]
Sep 26, 2021 3:16:35 PM org.apache.cxf.endpoint.ServerImpl initDestination
INFO: Setting the server's publish address to be /il8n
2021/05/11 15:16:35 - job_empleados - Starting entry [Mail de error - Sin archivo en el servidor]
2021/05/11 15:16:35 - Mail de error - Sin archivo en el servidor - Using run configuration [Pentaho
local]
2021/05/11 15:16:35 - Carte - Installing timer to purge stale objects after 1440 minutes.
2021/05/11 15:16:35 - job_envia_mail - Starting entry [Mail de error]
2021/05/11 15:16:35 - job_envia_mail - Starting entry [Success]
2021/05/11 15:16:35 - job_envia_mail - Finished job entry [Success] (result=[true])
2021/05/11 15:16:35 - job_envia_mail - Finished job entry [Mail de error] (result=[true])
2021/05/11 15:16:35 - job_empleados - Starting entry [No hay archivo en el servidor]
2021/05/11 15:16:35 - No hay archivo en el servidor - ERROR (version 8.2.0.0-342, build 8.2.0.0-342
from 2018-11-14 10.30.55 by buildguy) : No se cargó el archivo al servidor
2021/05/11 15:16:35 - job_empleados - Finished job entry [No hay archivo en el servidor] (result=[false])
2021/05/11 15:16:35 - job_empleados - Finished job entry [Mail de error - Sin archivo en el servidor]
(result=[false])
2021/05/11 15:16:35 - job_empleados - Finished job entry [Trajo algo el FTP?] (result=[false])
2021/05/11 15:16:35 - job_empleados - Finished job entry [Trae Access del Intranet] (result=[false])
2021/05/11 15:16:35 - job_empleados - Job execution finished
2021/05/11 15:16:35 - Kitchen - Finished!
2021/05/11 15:16:35 - Kitchen - ERROR (version 8.2.0.0-342, build 8.2.0.0-342 from 2018-11-14 10.30.55
by buildguy) : Finished with errors
2021/05/11 15:16:35 - Kitchen - Start=2021/05/11 15:16:27.585, Stop=2021/05/11 15:16:35.641
2021/05/11 15:16:35 - Kitchen - Processing ended after 8 seconds.

```

**Fig.5.13.** Log al ejecutar el job sin haber cargado el archivo fuente necesario

Además de la posibilidad de buscar en el log el fallo ocurrido, el proceso envía un correo electrónico al sector correspondiente indicando que sucedió este fallo en la ejecución del proceso para que pueda ser revisado. En este caso, se comprueba que el mail llega correctamente e indica en el asunto del mismo cual es el error, tal como se muestra en la Figura 5.14

The screenshot shows an email client interface. At the top, it says 'De Pentaho Data Integration' with a redacted email address and a 'Responde' button. The subject line is 'Asunto ALERTA: RNTEST112 - Error en PDI - El archivo Access para la carga del cubo no se encontró en el servidor.' The recipient is 'A Inteligencia de Negocios y Datos, David Ponce'. The main body of the email contains the text: 'El Job falló...' followed by a 'Job:' header and a list of job details, including 'JobName : job envia mail'.

**Fig.5.14.** Asunto del mail enviado desde el PDI para informar un error en el flujo de ejecución.

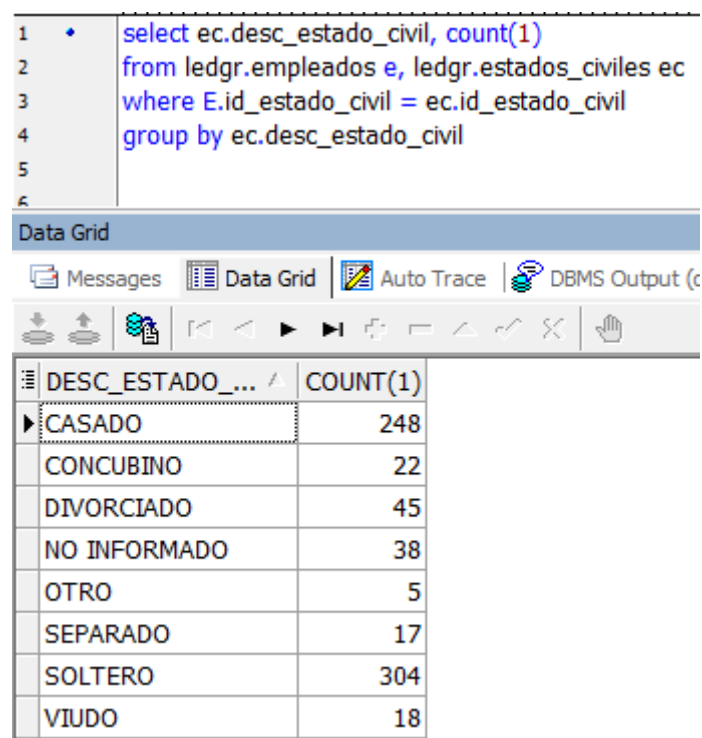
#### 5.1.4. Caso de prueba 4

- Objeto a analizar: Información cargada en la herramienta final de usuario y la visualización de la misma con Pentaho Analyzer
- Objetivo: Verificar que los datos que visualiza el usuario final sean correctos y completos.
- El usuario final del DM accede a la información cargada mediante la herramienta web Pentaho Analyzer. Desde ese lugar genera los reportes que necesita con los datos disponibles.

A continuación se muestra un resultado de ejemplo de las pruebas realizadas para verificar que la información que se encuentra en las fuentes de origen sea la misma que la que visualizan los usuarios del DM.

- Comprobar que la cantidad de empleados por estados civiles sea la correcta: Para eso se realizó una consulta a la BD fuente con la que se obtiene esta información. Se puede visualizar en la Figura 5.15.

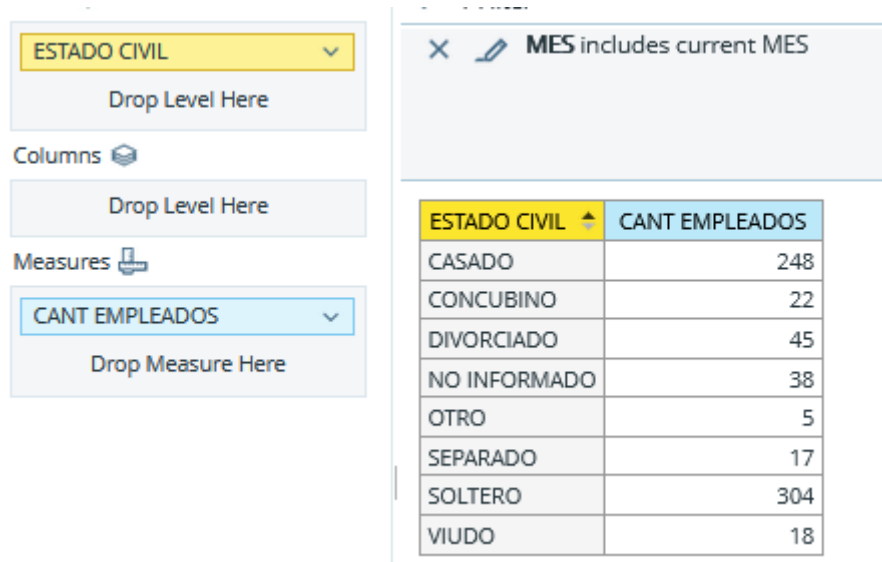
Desde Pentaho Analyzer se creó un reporte para que muestre esta misma información y las cantidades obtenidas fueron las mismas. Esto se puede observar en la Figura 5.16.



```
1 select ec.desc_estado_civil, count(1)
2 from ledgr.empleados e, ledgr.estados_civiles ec
3 where E.id_estado_civil = ec.id_estado_civil
4 group by ec.desc_estado_civil
```

DESC_ESTADO_...	COUNT(1)
CASADO	248
CONCUBINO	22
DIVORCIADO	45
NO INFORMADO	38
OTRO	5
SEPARADO	17
SOLTERO	304
VIUDO	18

Fig.5.15. Cantidad de empleados por estado civil desde la BD



**Fig.5.16.** Cantidad de empleados por estado civil desde Pentaho Analyser

Para cada una de las dimensiones se realizó esta verificación comprobando cantidades y también confirmando que los datos sean los que corresponden.

#### 5.1.5. Controles adicionales.

Además de los controles mencionados se realizan las siguientes verificaciones:

- Control de dimensiones en 0. Diariamente hay un job que se ejecuta a las 6:00hs de la mañana que comprueba en cada uno de los cubos si alguna de las dimensiones está con el valor igual a cero. En caso que una dimensión no se actualice y algún valor de dicha tabla no se cargue a la misma, al finalizar el cubo, puede suceder que si alguna de las filas de la tabla de hechos hace referencia al valor faltante, se cargue por defecto el valor 0. Se plantea el siguiente caso como ejemplo: en el momento en que se está actualizando el cubo, se crea un nuevo registro con una nueva nacionalidad desde el aplicativo de legajo electrónico y dicho valor no se alcanza a cargar en la tabla de la dimensión. Al momento de cargar los datos de un nuevo empleado que se asocia a la nueva nacionalidad, la relación a la dimensión en el cubo quedará con el valor cero ya que no se corresponde con ninguna cargada en ese momento. El proceso que se ejecuta diariamente identifica estos casos dentro de los cubos, por lo que se modificó para agregar el control al cubo Empleados.

- Grabado en la tabla Tiempos. Se agregó el código necesario en los jobs que conforman el ETL para el nuevo Data Mart, de esta forma, con una consulta sencilla en dicha tabla se pueden detectar problemas en la ejecución de los mismos. Al principio de cada job, se crea una fila en la tabla mencionada con los siguientes datos:
  - Hora de inicio: Se especifica el día y la hora en que comienza a ejecutarse el job.
  - Proceso: Se almacena el nombre del job que comenzó a ejecutarse
  - Error: En caso que el job finalice con algún error, antes de enviar el mail, se graba en esta columna el error que produjo el corte del job. En caso que no se produzca ninguno este dato queda vacío.
  - Hora de finalización: Tanto si el job termina de manera correcta o con algún error, se graba en esta columna la hora de finalización del mismo.

## 6. CONCLUSIONES

En este capítulo se presentan las conclusiones finales del Trabajo Final de Carrera y se muestran posibles líneas futuras de trabajo relacionadas con el presente proyecto.

La finalidad de este trabajo se puede resumir en brindar ayuda para la toma de decisiones de una Gerencia basadas en los datos con los que cuenta. Con esta premisa es necesario que dichos datos se encuentren disponibles y accesibles de una manera integrada, estructurada y completa.

Si bien el área de Recursos Humanos en particular no genera un volumen de información tan grande como otras dentro de la Agencia, la necesidad de gestión y análisis es igual de importante. La forma de lograr esto es a través de la implementación de soluciones de Inteligencia de Negocios, y en particular se optó en este caso por el desarrollo de un Data Mart debido a todos los beneficios que conlleva, entre los que podemos nombrar:

- Los datos se encuentran integrados y estandarizados en un único lugar.
- Todas las necesidades referidas al acceso a la información se encuentran resueltas por medio del uso de la herramienta seleccionada para ello.
- El acceso a la información se hace de manera rápida y se pueden generar y visualizar reportes en tiempo real.
- Los cambios, mejoras, adaptaciones y nuevas necesidades se pueden resolver de una manera rápida e integrada debido a la propiedad de escalabilidad que presenta.
- Se reduce la cantidad de solicitudes de creación de reportes e informes al área de Tecnologías de la Información.

El desarrollo se planteó siguiendo los lineamientos de la metodología Bottom-up, describiendo y documentando uno a uno los pasos realizados. En cuanto al diseño se optó por el modelo dimensional de estrella definiendo de esta manera la estructura física del DM.

Mediante la preparación de la información y la implementación de procesos de extracción, transformación y carga, se incorporaron un conjunto de datos relevantes, definidos previamente al Data Mart. La nueva estructura se incorporó al



resto de los Data Marts con los que ya cuenta la Organización, conformando de esta manera el Data Warehouse de la ARTRN.

Como resultado del desarrollo el área de RRHH podrá invertir mayor tiempo en el análisis de los datos y no en el armado de reportes, con todo lo que esto implica, como por ejemplo, integración y formateo de los datos, trabajo manual de control sobre la información obtenida, entre otros. Con esta implementación se logró también unificar la información que se encuentra en dos lugares diferentes, con distintas formas de administración y acceso, permitiendo tener una visión completa de la misma y posibilitando la realización de reportes de análisis con la totalidad de los datos.

## **6.1. Líneas Futuras**

Se pueden encontrar varios lineamientos para continuar con el desarrollo y realizar mejoras al Data Mart implementado.

Existe la posibilidad que los datos tomados desde el archivo en Access se incorporen al sistema operacional mediante una modificación en el desarrollo de este sistema. En este caso, se deberán modificar los procesos de extracción, transformación y carga para tomar los datos desde la nueva estructura.

Es posible definir la creación y estructuras de reportes para luego realizar programaciones automáticas de los mismos. De esta manera se puede automatizar el envío de información actualizada, de manera periódica, tanto para que lo reciba personal de la Gerencia de Recursos Humanos como para la distribución de reportes hacia diversas áreas de la Agencia e incluso hacia otros Organismos públicos.

En el proceso de evolución del Data Mart se podrá asociar además otros cubos con información relacionada con los empleados, como por ejemplo, información sobre sueldos, horarios de entrada y salida, licencias, etc., ampliando de esta forma las capacidades de análisis de la información para la toma de decisiones.

## 7. REFERENCIAS BIBLIOGRÁFICAS

- Adó, M., Rastelli, M. C., Smail, A., Bertone, R. (2015). Desarrollo de herramientas para warehousing en el Municipio de Junín. Recuperado a partir de <http://sedici.unlp.edu.ar/handle/10915/50535>
- Aimacaña Quilumba, D. E. (2013). ANÁLISIS, DISEÑO E IMPLEMENTACIÓN DE UN DATA MART ACADÉMICO USANDO TECNOLOGÍA DE BI PARA LA FACULTAD DE INGENIERÍA, CIENCIAS FÍSICAS Y MATEMÁTICA. [Trabajo de grado, UNIVERSIDAD CENTRAL DEL ECUADOR]. Recuperado a partir de <http://www.dspace.uce.edu.ec/handle/25000/999>
- Bernabeu, R. D. (2009). DATA WAREHOUSING: Investigación y Sistematización de Conceptos– HEFESTO: Metodología propia para la Construcción de un Data Warehouse.
- Castillo, J. Y. y Palomino Paniora, L. (2013). Implementación de un Datamart como una solución de Inteligencia de Negocios para el área de logística de T-Impulso. *REVISTA DE INVESTIGACIÓN DE SISTEMAS E INFORMÁTICA*, 10, 53–63.
- Corso, C.L., Luque, C., Ciceri, L., & Donnet, M. (2015). Diseño de almacén de datos para el análisis eficiente de la información de incidentes informáticos y mantenimientos.
- Domingues, M. A., Soares, C., Jorge, A. M. y Rezende, S. O. (2014). A data warehouse to support web site automation. *Journal of the Brazilian Computer Society*, 20, Artículo 11. <https://doi.org/10.1186/1678-4804-20-11>
- Formia, S., y Estévez, E. (2017). Implementación y maduración de un data warehouse – caso de estudio de la Agencia de Recaudación Tributaria de Río Negro (ARTRN).
- Inmon, W. H. (2002). *Building the Data Warehouse*. John Wiley & Sons, Ltd.
- Kemp De La Hoz, E. A. (2005). Desarrollo de una metodología que permita a empresas el desarrollo de un Data Warehouse y su integración con Sistemas Workflow utilizando herramientas de libre distribución y/o bajo costo. [Trabajo de grado, UNIVERSIDAD AUSTRAL DE CHILE]. Recuperado a partir de <http://cybertesis.uach.cl/tesis/uach/2005/bmfcik.32d/doc/bmfcik.32d.pdf>

- Mendez, A., Mártire, A., Britos, P. y Garcia-Martínez, R. (2003). Fundamentos de Data Warehouse - Reportes Técnicos en Ingeniería del Software, vol. 5, 19-26
- Minnaard C., Servetto D., Lobo Mirassón U., Pascal G. (2015). La información y la tecnología para la toma de decisiones: Aplicación Data WareHouse en instituciones universitarias. Recuperado a partir de <http://digital.cic.gba.gob.ar/handle/11746/4721>
- Orellana Sanchez, F. A. (2013). Propuesta de implementación de un Data Warehouse para el área de Soporte de Información, Rabie S.A [Trabajo de grado]. Universidad del Bio-Bio.
- Pentaho*. (2018). Hitachi Vantara Lumada and Pentaho Documentation. Sitio Oficial. Recuperado a partir de <https://help.hitachivantara.com/Documentation/Pentaho> [Consultado Agosto 2021]
- Pentaho Mondrian Documentation*. Sitio oficial. Recuperado a partir de <https://mondrian.pentaho.com/documentation/schema.php> [Consultado Agosto 2021]
- Rivadera, G. R. (2019). La metodología de Kimball para el diseño de almacenes de datos (Data warehouses). Cuadernos de la Facultad de Ingeniería e Informática, (5), p. 56-71 . Recuperado a partir de <https://revistas.ucasal.edu.ar/index.php/CI/article/view/169>
- Ross, M. y Kimball, R. (2002). The Data Warehouse Toolkit: The Complete Guide to Dimensional Modeling (Second Edition). Wiley.
- Silvers, F. (2008). Building and Maintaining a Data Warehouse. AUERBACH.
- Vaisman, A. y Zimányi, E. (2014). *Data Warehouse Systems*. Springer Berlin Heidelberg. <https://doi.org/10.1007/978-3-642-54655-6>
- Villarreal Rosero, R. X. (2013). Estudio de metodologías de Data Warehouse para la implementación de repositorios de información para la toma de decisiones gerenciales. [Trabajo de grado, UNIVERSIDAD TÉCNICA DEL NORTE]. Recuperado a partir de <http://repositorio.utn.edu.ec/bitstream/123456789/2660/1/04%20ISC%20279%20TESIS.pdf>.
- Zaldívar Rojas, Alejandro. (2014). IMPLEMENTACIÓN DE UN DATA MART COMO SOLUCIÓN DE INTELIGENCIA DE NEGOCIOS, BAJO LA METODOLOGÍA

DE RALPH KIMBALL PARA OPTIMIZAR LA TOMA DE DECISIONES EN EL DEPARTAMENTO DE FINANZAS DE LA CONTRALORÍA GENERAL DE LA REPÚBLICA [Trabajo de grado]. Universidad San Martín de Porres, Chiclayo, Perú.

## 8. ANEXOS

### ANEXO A: ARCHIVO XML CREADO PARA LA DEFINICIÓN DEL CUBO EMPLEADOS

```
<Schema name="RRHH" description="Cubos referentes a los RRHH de la
Agencia">
  <Dimension type="StandardDimension" visible="true" highCardinality="false"
name="EMPLEADO" description="Empleado">
    <Hierarchy name="EMPLEADO" visible="true" hasAll="true"
primaryKey="DIMENSION_KEY" description="Empleados de la ARTRN">
      <Table name="LE_EMPLEADOS" schema="DW"></Table>
      <Level name="APELLIDO Y NOMBRE" visible="true" column="APYNOM"
type="String" uniqueMembers="false" levelType="Regular" hideMemberIf="Never"
description="Apellido y nombre del empleado">
        <Property name="CUIL" column="CUIL" type="String"></Property>
        <Property name="NRO LEGAJO" column="NRO_LEGAJO"
type="String"></Property>
        <Property name="AP y NOM MADRE" column="MADRE_APYNOM"
type="String"></Property>
        <Property name="DATOS MADRE" column="DATOS_MADRE"
type="String"></Property>
        <Property name="AP y NOM PADRE" column="PADRE_APYNOM"
type="String"></Property>
        <Property name="DATOS PADRE" column="DATOS_PADRE"
type="String"></Property>
        <Property name="NRO DOCUMENTO" column="NRO_DOCUMENTO"
type="String"></Property>
      </Level>
    </Hierarchy>
  </Dimension>
  <Dimension type="TimeDimension" visible="true" highCardinality="false"
name="FECHA" caption="FECHA" description="Fecha">
    <Hierarchy name="FECHA" visible="true" hasAll="true"
primaryKey="DIMENSION_KEY" description="Fecha">
      <Table name="TIEMPOAMD" schema="DW"></Table>
      <Level name="CALENDAR YEAR CAL YEAR CODE" visible="true"
column="CALENDAR_YEAR_CAL_YEAR_CODE" type="Numeric"
uniqueMembers="false" levelType="TimeYears" hideMemberIf="Never"

```

caption="AÑO" description="Año. Nota: El DW muestra años desde 1980, para años anteriores a este se verá No disponible">

<Annotations>

<Annotation name="AnalyzerDateFormat">

<![CDATA[[yyyy]]]>

</Annotation>

</Annotations>

</Level>

<Level name="CALENDAR\_MONTH CAL MONTH CODE" visible="true"

column="CALENDAR\_MONTH\_CAL\_MONTH\_CODE"

ordinalColumn="CAL\_MONTH\_NUMBER" type="Numeric" uniqueMembers="false"

levelType="TimeMonths" hideMemberIf="Never" caption="MES"

description="Año - Mes">

<Annotations>

<Annotation name="AnalyzerDateFormat">

<![CDATA[[yyyy].[yyyyMM]]]>

</Annotation>

</Annotations>

<Property name="NOMBRE MES"

column="CALENDAR\_MONTH\_NAME" type="String" description="Por ejemplo May 2013"></Property>

</Level>

<Level name="DAY DAY CODE" visible="true" column="DAY\_DAY\_CODE"

ordinalColumn="DAY\_OF\_CAL\_MONTH" type="Numeric" uniqueMembers="false"

levelType="TimeDays" hideMemberIf="Never" caption="FECHA" description="Fecha AAAAMMDD">

<Annotations>

<Annotation name="AnalyzerDateFormat">

<![CDATA[[yyyy].[yyyyMM].[yyyyMMdd]]]>

</Annotation>

</Annotations>

<Property name="NOMBRE DIA" column="DAY\_NAME" type="String"

description="Por ejemplo 15-MAY-2013"></Property>

</Level>

</Hierarchy>

</Dimension>

<Cube name="EMPLEADOS" visible="true" description="Contiene los datos de todos los empleados de la agencia. Se guardan en concepto de Fotos por lo que hay que tener en cuenta la fecha de carga de la foto. Se puede filtrar por apellido y nombre del empleado, categoría, situación de revista, unidad organizativa, etc. Los

```

datos se toman de la base del legajo digital y de la base PLANTA PERSONAL
(access de RRHH)." cache="true" enabled="true">
  <Table name="CUBO_EMPLEADOS" schema="DW"></Table>
  <DimensionUsage source="EMPLEADO" name="EMPLEADO" visible="true"
foreignKey="EMPLEADO" highCardinality="false"></DimensionUsage>
  <Dimension type="StandardDimension" visible="true"
foreignKey="AGRUP_CAT" highCardinality="false" name="CATEGORIA">
  <Hierarchy name="CATEGORIA " visible="true" hasAll="true"
primaryKey="DIMENSION_KEY" caption="CATEGORIA " description="Categoría
del empleado al momento de la foto">
  <Table name="LE_AGRUP_CAT" schema="DW"></Table>
  <Level name="REGIMEN" visible="true" column="REGIMEN"
type="String" uniqueMembers="false" levelType="Regular" hideMemberIf="Never"
caption="REGIMEN" description="Regimen del empleado al momento de la foto (Ej:
Funcionarios, Ley 1844)"></Level>
  <Level name="AGRUPAMIENTO" visible="true"
column="AGRUPAMIENTO" type="String" uniqueMembers="false"
levelType="Regular" hideMemberIf="Never" caption="AGRUPAMIENTO"
description="Agrupamiento del empleado al momento de la foto (Ej: Ley 1844
administrativo, Ley 1844 informaticos)"></Level>
  <Level name="CATEGORIA" visible="true" column="CATEGORIA"
type="String" uniqueMembers="false" levelType="Regular" hideMemberIf="Never"
caption="CATEGORIA" description="Categoría del empleado al momento de la foto,
(Categoría 1, 2, 3, Director, Gerente, etc)"></Level>
  </Hierarchy>
</Dimension>
  <DimensionUsage source="FECHA" name="FECHA FOTO" visible="true"
foreignKey="FECHA_FOTO" highCardinality="false"></DimensionUsage>
  <Dimension type="StandardDimension" visible="true"
foreignKey="ESTADO_CIVIL" highCardinality="false" name="ESTADO CIVIL">
  <Hierarchy name="ESTADO_CIVIL" visible="true" hasAll="true"
primaryKey="DIMENSION_KEY">
  <Table name="LE_ESTADOS_CIVILES" schema="DW"></Table>
  <Level name="ESTADO CIVIL" visible="true" column="ESTADO_CIVIL"
type="String" uniqueMembers="false" levelType="Regular" hideMemberIf="Never"
description="Estado civil del empleado (ej: casado, soltero, concubino,
etc.)"></Level>
  </Hierarchy>
</Dimension>

```

```

    <Dimension type="StandardDimension" visible="true"
foreignKey="NACIONALIDAD" highCardinality="false" name="NACIONALIDAD">
    <Hierarchy name="NACIONALIDAD" visible="true" hasAll="true"
primaryKey="DIMENSION_KEY">
        <Table name="LE_NACIONALIDADES" schema="DW"></Table>
        <Level name="NACIONALIDAD" visible="true" column="NACIONALIDAD"
type="String" uniqueMembers="false" levelType="Regular" hideMemberIf="Never"
description="Nacionalidad del empleado"></Level>
    </Hierarchy>
</Dimension>
    <Dimension type="StandardDimension" visible="true"
foreignKey="ORGANIZACION" highCardinality="false" name="ORGANIZACION">
    <Hierarchy name="ORGANIZACION" visible="true" hasAll="true"
primaryKey="DIMENSION_KEY" description="Departamento, delegación o sede de
funciones del empleado (ej: Delegación Zonal Viedma, Asesoría legal, Patrimonio,
T&#233;cnico Operativo)">
        <Table name="LE_ORGANIZACIONES" schema="DW"></Table>
        <Level name="ORGANIZACION" visible="true"
column="ORGANIZACION" type="String" uniqueMembers="false"
levelType="Regular" hideMemberIf="Never"></Level>
    </Hierarchy>
</Dimension>
    <Dimension type="StandardDimension" visible="true"
foreignKey="REG_JUBILATORIO" highCardinality="false" name="REG
JUBILATORIO">
    <Hierarchy name="REG JUBILATORIO" visible="true" hasAll="true"
primaryKey="DIMENSION_KEY">
        <Table name="LE_REG_JUBILATORIOS" schema="DW"></Table>
        <Level name="REG JUBILATORIO" visible="true"
column="REG_JUBILATORIO" type="String" uniqueMembers="false"
levelType="Regular" hideMemberIf="Never" description="indica si cuenta con
alg&#250;n regimen jubilatorio (Capitalizacion - No aporta - Reparto - Retirado
jubilado)"></Level>
    </Hierarchy>
</Dimension>
    <Dimension type="StandardDimension" visible="true"
foreignKey="SITUACION_REVISTA" highCardinality="false" name="SITUACION
REVISTA">
    <Hierarchy name="SITUACION REVISTA" visible="true" hasAll="true"
primaryKey="DIMENSION_KEY">

```



```

    <Table name="LE_SIT_REVISTA" schema="DW"></Table>
    <Level name="SITUACION REVISTA" visible="true"
column="SITUACION_REVISTA" type="String" uniqueMembers="false"
levelType="Regular" hideMemberlf="Never" description="Indica la situación de
revista del empleado (Contratado - jornalizado - Planta permanente)"></Level>
    </Hierarchy>
</Dimension>
    <Dimension type="StandardDimension" visible="true"
foreignKey="UNIDAD_ORG" highCardinality="false" name="UNIDAD
ORGANIZATIVA">
    <Hierarchy name="UNIDAD ORGANIZATIVA" visible="true" hasAll="true"
primaryKey="DIMENSION_KEY">
    <Table name="LE_UN_ORGANIZATIVAS" schema="DW"></Table>
    <Level name="UNIDAD ORGANZATIVA" visible="true"
column="UN_ORGANIZATIVA" type="String" uniqueMembers="false"
levelType="Regular" hideMemberlf="Never" description="Sede de funciones del
empleado (delegación, zonal, receptoría, etc.)"></Level>
    </Hierarchy>
</Dimension>
    <Dimension type="StandardDimension" visible="true" foreignKey="GENERO"
highCardinality="false" name="GENERO">
    <Hierarchy name="GENERO" visible="true" hasAll="true"
primaryKey="DIMENSION_KEY">
    <Table name="LE_GENEROS" schema="DW"></Table>
    <Level name="GENERO" visible="true" column="GENERO" type="String"
uniqueMembers="false" levelType="Regular" hideMemberlf="Never"
description="G&#233;nero del empleado"></Level>
    </Hierarchy>
</Dimension>
    <Dimension type="StandardDimension" visible="true" foreignKey="ESTADO"
highCardinality="false" name="ESTADO">
    <Hierarchy name="ESTADO" visible="true" hasAll="true"
primaryKey="DIMENSION_KEY">
    <Table name="LE_ESTADOS" schema="DW"></Table>
    <Level name="ESTADO" visible="true" column="ESTADO" type="String"
uniqueMembers="false" levelType="Regular" hideMemberlf="Never"
description="Indica si el empleado está activo o con baja al momento de la
foto"></Level>
    </Hierarchy>
</Dimension>

```

```

    <Dimension type="StandardDimension" visible="true"
foreignKey="GRADO_ACADEMICO" highCardinality="false" name="GRADO
ACADEMICO">
        <Hierarchy name="GRADO_ACADEMICO" visible="true" hasAll="true"
primaryKey="DIMENSION_KEY">
            <Table name="LE_GRADOS_ACADEMICOS" schema="DW"></Table>
            <Level name="UNIVERSIDAD" visible="true" column="INSTITUCION"
type="String" uniqueMembers="false" levelType="Regular" hideMemberIf="Never"
description="Institución en la que obtuvo el máximo título (ej: CEM Provincial,
Instituto, Universidad Nacional de la Plata)"></Level>
            <Level name="TIPO TITULO" visible="true" column="TIPO_TITULO"
type="String" uniqueMembers="false" levelType="Regular" hideMemberIf="Never"
description="Máximo tipo de título obtenido por el empleado (Ej: Secundario
completo , Terciario, Posgrado, etc)"></Level>
            <Level name="TITULO" visible="true" column="TITULO" type="String"
uniqueMembers="false" levelType="Regular" hideMemberIf="Never"
description="Máximo título obtenido al momento de la foto"></Level>
        </Hierarchy>
    </Dimension>
    <Dimension type="StandardDimension" visible="true"
foreignKey="LOCALIDAD" highCardinality="false" name="LOCALIDAD">
        <Hierarchy name="LOCALIDAD" visible="true" hasAll="true"
primaryKey="DIMENSION_KEY">
            <Table name="LE_LOCALIDADES" schema="DW"></Table>
            <Level name="PROVINCIA" visible="true" column="PROVINCIA"
type="String" uniqueMembers="false" levelType="Regular" hideMemberIf="Never"
description="Localidad donde desarrolla sus funciones el empleado al momento de
la foto"></Level>
            <Level name="LOCALIDAD" visible="true" column="LOCALIDAD"
type="String" uniqueMembers="false" levelType="Regular" hideMemberIf="Never"
description="Localidad donde desarrolla sus funciones el empleado al momento de
la foto"></Level>
        </Hierarchy>
    </Dimension>
    <Dimension type="StandardDimension" visible="true"
foreignKey="CARGA_HORARIA" highCardinality="false" name="CARGA
HORARIA">
        <Hierarchy name="CARGA HORARIA" visible="true" hasAll="true"
primaryKey="DIMENSION_KEY">
            <Table name="LE_CARGAS_HORARIAS" schema="DW"></Table>

```

```

    <Level name="CARGA HORARIA" visible="true"
column="CARGA_HORARIA" type="String" uniqueMembers="false"
levelType="Regular" hideMemberIf="Never" description="Carga horaria adicional del
empleado al momento de la foto. Extraído de la Base Pta.Personal"></Level>
  </Hierarchy>
</Dimension>
  <Dimension type="StandardDimension" visible="true"
foreignKey="LIC_SIN_HAB" highCardinality="false" name="LIC. SIN HABERES">
  <Hierarchy name="LIC. SIN HABERES" visible="true" hasAll="true"
primaryKey="DIMENSION_KEY">
    <Table name="LE_LIC_SIN_HABERES" schema="DW"></Table>
    <Level name="LIC. SIN HABERES" visible="true"
column="LIC_SIN_HABERES" type="String" uniqueMembers="false"
levelType="Regular" hideMemberIf="Never" description="Indica si tiene licencia con
goce de haberes al momento de la carga de la foto. Extraído de la Base
Pta.Personal"></Level>
  </Hierarchy>
</Dimension>
  <Dimension type="StandardDimension" visible="true" foreignKey="BAJA"
highCardinality="false" name="BAJA">
  <Hierarchy name="BAJA" visible="true" hasAll="true"
primaryKey="DIMENSION_KEY">
    <Table name="LE_BAJAS" schema="DW"></Table>
    <Level name="BAJA" visible="true" column="BAJA" type="String"
uniqueMembers="false" levelType="Regular" hideMemberIf="Never"
description="Indica si tiene baja provisoria o definitiva al momento de la foto.
Extraído de la Base Pta.Personal"></Level>
  </Hierarchy>
</Dimension>
  <Dimension type="StandardDimension" visible="true" foreignKey="EDAD"
highCardinality="false" name="EDAD">
  <Hierarchy name="EDAD" visible="true" hasAll="true" primaryKey="EDAD">
    <Table name="CUBO_EMPLEADOS" schema="DW"></Table>
    <Level name="EDAD" visible="true" column="EDAD" type="String"
uniqueMembers="false" levelType="Regular" hideMemberIf="Never"
description="Edad del empleado (Tener en cuenta la fecha de la foto que se está
mirando)"></Level>
  </Hierarchy>
</Dimension>

```

```
<Measure name="CANT EMPLEADOS" column="EMPLEADO"
aggregator="distinct count" visible="true"></Measure>
  <Measure name="EDAD" column="EDAD" aggregator="avg"
visible="true"></Measure>
</Cube>
<Role name="Pentaho-Admin">
  <SchemaGrant access="all"></SchemaGrant>
</Role>
<Role name="Pentaho-Sistemas">
  <SchemaGrant access="all"></SchemaGrant>
</Role>
<Role name="Pentaho-RecursosHumanos">
  <SchemaGrant access="all"></SchemaGrant>
</Role>
<Role name="Pentaho-Directores">
  <SchemaGrant access="all"></SchemaGrant>
</Role>
</Schema>
```

## ANEXO B: SCRIPT PARA LA EJECUCIÓN DEL JOB DE ETL PRINCIPAL

```
#
#
# ==> Carga del Cubo de Empleados con Job (kettle)
#
set `date`
#
echo "- - - - -" > ../LOGS-TRAFOS/etl-empleados-`date`.dat
echo "Inicio de la CARGA: " `date` >> ../LOGS-TRAFOS/etl-empleados-`date`.dat
echo "- - - - -" >> ../LOGS-TRAFOS/etl-empleados-`date`.dat
echo " " >> ../LOGS-TRAFOS/etl-empleados-`date`.dat
#
#
# ==> Voy donde esta PDI y ejecuto
cd ../design-tools/data-integration
./kitchen.sh -file=/home/.../TRANSFORMACIONES/job_empleados.kjb 2>
../LOGS-TRAFOS/etl-empleados-`date`.err > /tmp/etl-empleados.log
#
# ==> cd ../LOGS-TRAFOS
cat /tmp/etl-empleados.log >> ../LOGS-TRAFOS/etl-empleados-`date`.dat
rm -f /tmp/etl-empleados.log
#
#
echo "- - - - -" >> ../LOGS-TRAFOS/etl-empleados-`date`.dat
echo "Fin de la CARGA...: " `date` >> ../LOGS-TRAFOS/etl-empleados-`date`.dat
echo "- - - - -" >> ../LOGS-TRAFOS/etl-empleados-`date`.dat
echo " " >> ../LOGS-TRAFOS/etl-empleados-`date`.dat
#
etl-empleados.sh (END)
```

**Fig.B.1.** Script etl-empleados.sh que ejecuta el job ETL y almacena archivos de log