

# **DISEÑO Y CONSTRUCCIÓN DE DATA WAREHOUSE**

**Proyecto Encuesta Continua de Hogares**  
**Grupo 16**

Martín Rodríguez 4.123.589-6

Marcelo Casiraghi 4.703.481-6

Montevideo, 27 de Junio de 2012, InCo – Fing - UDeLaR

# Índice

1. Objetivos del proyecto.....	3
2. Diseño Conceptual.....	3
2.1. Requerimiento funcional 1.....	3
2.2. Requerimiento funcional 2.....	6
2.3. Requerimiento funcional 3.....	8
2.4. Requerimiento funcional a.....	10
2.4. Requerimiento funcional adicional.....	10
3. Diseño Lógico.....	11
3.1. Requerimiento funcional 1.....	11
3.2. Requerimiento funcional 2.....	12
3.3. Requerimiento funcional 3.....	13
3.4. Requerimiento adicional.....	14
4. Implementación de relaciones dimensionales y dimensiones.....	15
5. Documentación del proceso de carga.....	19
5.1. Algoritmos de carga.....	20
5.1.1. Carga de las tablas de cada dimensión.....	20
5.1.2. Carga de la tabla Información de Hogares.....	21
5.1.2. Carga de la tabla Información de Personas.....	22
5.1.3. Carga de la tabla Información de Tecnologías.....	23
5.1.4. Carga de la tabla Información Ingreso Racial.....	23
6. Reportes.....	24
7. Testing de la solución.....	25
8. Conclusiones y dificultades encontradas.....	26
Bibliografía.....	28

## **1. Objetivos del proyecto**

El objetivo principal del proyecto es realizar un análisis multidimensional sobre algunos aspectos de los microdatos disponibles de la ECH (Encuesta Continua de Hogares) publicada por el INE (Instituto Nacional de Estadística) de los años 2009, 2010 y 2011.

Para alcanzar estos objetivos se debió:

1. Realizar un modelo conceptual dimensional de las dimensiones y relaciones dimensionales que surgen del análisis de los requerimientos planteados.
2. Diseñar e implementar un modelo lógico relacional que de soporte al modelo conceptual desarrollado en el punto anterior, teniendo en cuenta las restricciones impuestas por las herramientas utilizadas.
3. Diseñar e implementar los procesos de carga del modelo lógico del punto anterior utilizando la herramienta open source Pentaho Data Integration.
4. Implementar las dimensiones y relaciones dimensionales diseñadas en el punto 1 mediante la generación de cubos utilizando la herramienta Schema Workbench de Pentaho.
5. Publicar los cubos generados en un servidor ROLAP para poder visualizar las posibles consultas al DataWarehous construido. Se requirió utilizar el servidor Mondrian de Pentaho.

## **2. Diseño Conceptual**

Para poder realizar el diseño conceptual de la realidad planteada, debimos previamente estudiar lo más en detalle posible los metadatos de la ECH de los años 2009, 2010 y 2011, de forma tal de poder representar las dimensiones y relaciones dimensionales lo más acorde posible a la realidad planteada.

Notamos que la representación de los datos que requeríamos es casi idéntica para los años 2009 y 2010, pero difiere sustancialmente para el año 2011. Sobre este punto nos explayaremos cuando expliquemos la carga de los datos, donde detallamos en forma concisa los algoritmos utilizados.

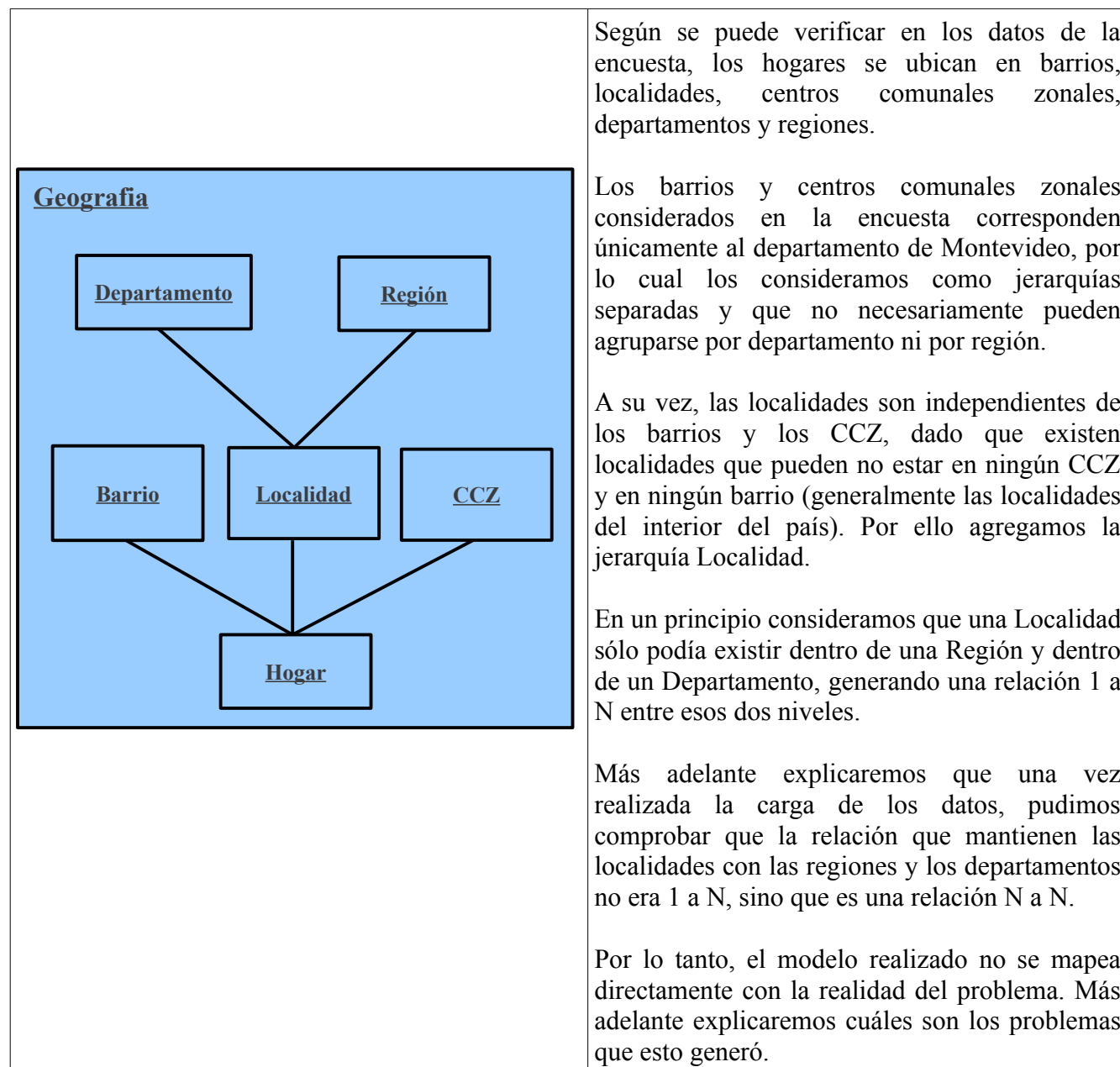
Las dimensiones y relaciones dimensionales las definimos para cada uno de los requerimientos funcionales presentados en la letra del obligatorio, habiéndose presentando el caso de dimensiones comunes. Por ello creemos que la mejor forma de explicar cómo realizamos el diseño conceptual de la realidad planteada es dividiendo por requerimientos.

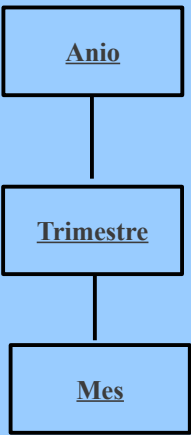


### **2.1. Requerimiento funcional 1**

En este requerimiento se plantea obtener la información de los hogares según el tipo de la vivienda, el nivel de confort de la vivienda, la ubicación geográfica y el tiempo. Para ello identificamos las siguientes dimensiones:

1. Tiempo
2. Geografía
3. Tipo de Vivienda
4. Nivel de Confort de la Vivienda

A continuación presentamos una representación gráfica de las dimensiones con sus respectivas jerarquías y niveles, junto con una explicación de su construcción:

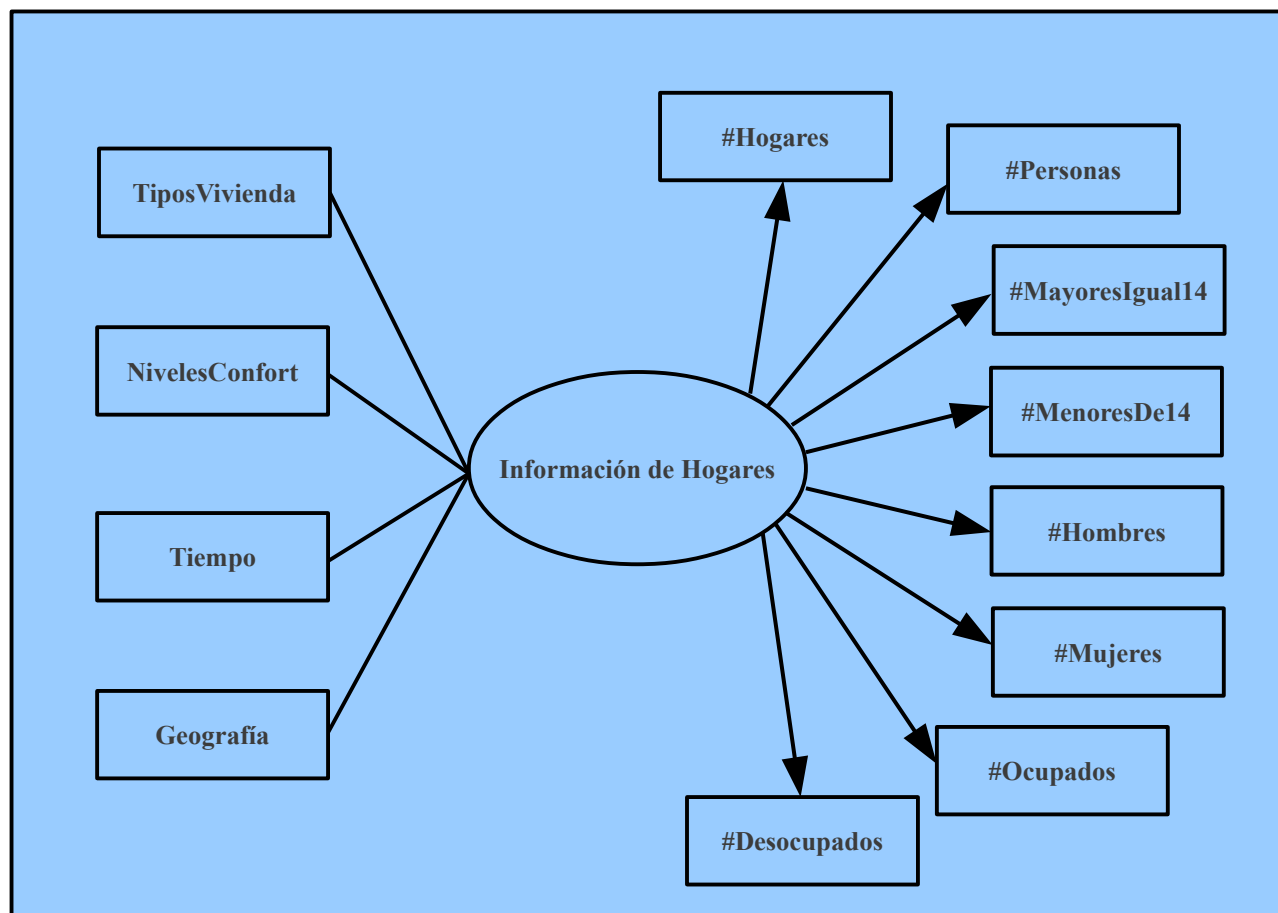


<p><b><u>Tiempo</u></b></p>  <pre> graph TD   A[Año] --- B[Trimestre]   B --- C[Mes] </pre>	<p>La información de los hogares interesa ser analizada por mes, trimestre y por año, por lo que se consideraron estos tres conceptos como niveles distintos de una misma jerarquía en la dimensión Tiempo.</p>
<p><b><u>Niveles de Confort</u></b></p>  <pre> graph TD   A[Tipo] </pre>	<p>Los niveles de confort de una vivienda se dividen en cuatro tipos, y su cálculo depende de tres factores que serán explicados en la etapa del diseño lógico: cantidad de electrodomésticos que cuenta la vivienda, el origen del agua y si la vivienda cuenta con servicio doméstico o no.</p> <p>Se define una dimensión Niveles de Confort con un único nivel.</p>
<p><b><u>Tipos de Vivienda</u></b></p>  <pre> graph TD   A[Tipo] </pre>	<p>Se define la dimensión Tipos de Vivienda que comprende los tipos de vivienda que describe el requerimiento: casa, apartamento y otros.</p>

Una vez definimos las cuatro dimensiones anteriores, identificamos las medidas requeridas:

1. Cantidad de hogares en cada situación.
2. Cantidad de personas en total.
3. Cantidad de personas de 14 años o más.
4. Cantidad de personas menores de 14 años.
5. Cantidad de hombres.
6. Cantidad de mujeres.
7. Cantidad de ocupados.
8. Cantidad de desocupados.

A continuación se presenta un esquema de la relación dimensional Información de Hogares, que modela el requerimiento funcional:



## 2.2. Requerimiento funcional 2

En este requerimiento se pide analizar a las personas según su ocupación, educación cursada y finalizada, salud, sexo, ubicación geográfica, edad y tiempo y la cantidad de personas que se encuentran en cada situación. También se pide la suma de los ingresos de cada persona. Se identificaron entonces las siguientes dimensiones:

1. Ocupaciones
2. Educación (cursada y finalizada)
3. Salud
4. Sexos
5. Geografía
6. Edades
7. Tiempo

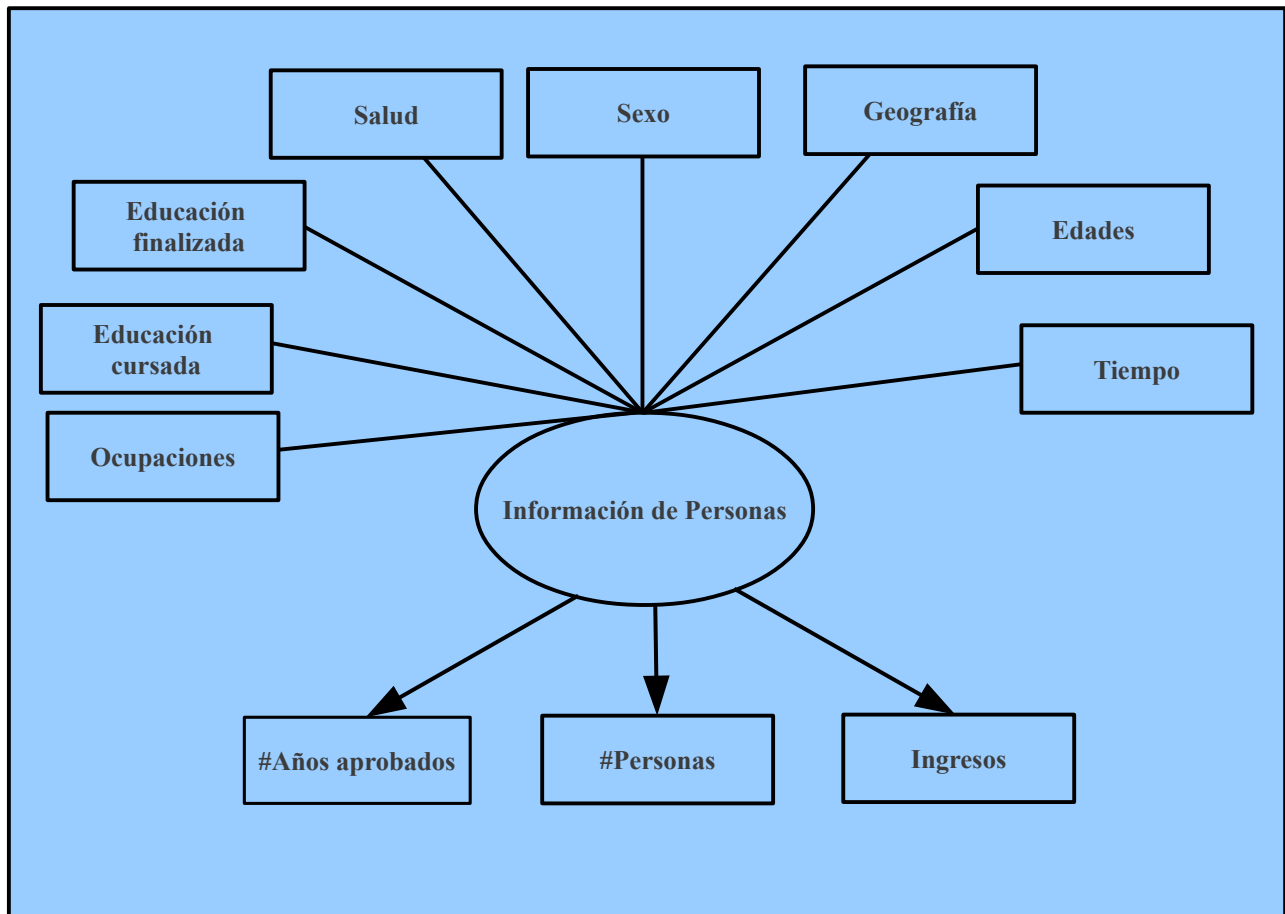
Las medidas identificadas son la cantidad de personas, la cantidad de años aprobados de la misma y la

suma de sus ingresos.

Cabe destacar, las dimensiones Geografía y Tiempo son las mismas que las definidas en el requerimiento funcional anterior. Por lo tanto, a continuación exponemos una representación gráfica únicamente de las nuevas dimensiones, junto con una breve explicación de su definición:

<div><b><u>Ocupaciones</u></b><div><b><u>Ocupacion</u></b></div></div>	La dimensión Ocupaciones considera todos los valores mencionados en el requerimiento. Se mencionan los mismos en la sección correspondiente al diseño lógico.
<div><b><u>Educación</u></b><div><b><u>Tipo</u></b></div></div>	Para realizar el análisis de las personas según su educación cursada y educación aprobada, se define la dimensión Educación. Se utilizarán dos instancias de esta dimensión para la relación dimensional que representa este requerimiento.
<div><b><u>Salud</u></b><div><b><u>Salud</u></b></div></div>	La información de la salud requerida consiste en si la persona se atiende en MSP, IAMC, Privado o en otro tipo de sistema.  Se define la dimensión Salud con una sola jerarquía de un nivel.
<div><b><u>Sexos</u></b><div><b><u>Sexo</u></b></div></div>	Se define la dimensión Sexos con una jerarquía de un solo nivel que contempla los tipos Masculino y Femenino.

A continuación se presenta un esquema de la relación dimensional Información de Personas, que modela el requerimiento funcional:



Como se mencionó anteriormente, en la relación dimensional anterior se utilizan dos instancias de la dimensión Educación: Una que representa la educación cursada por la persona, y otra que indica la educación finalizada por la misma.

### 2.3. Requerimiento funcional 3

Se pide analizar a las personas según su ubicación geográfica, el tiempo, el sexo, la edad, la educación y la utilización de las TICs (Tecnologías de la Información), viendo los años aprobados en el sistema educativo y la cantidad de personas que se encuentran en cada situación.

Se identificaron las siguientes dimensiones:

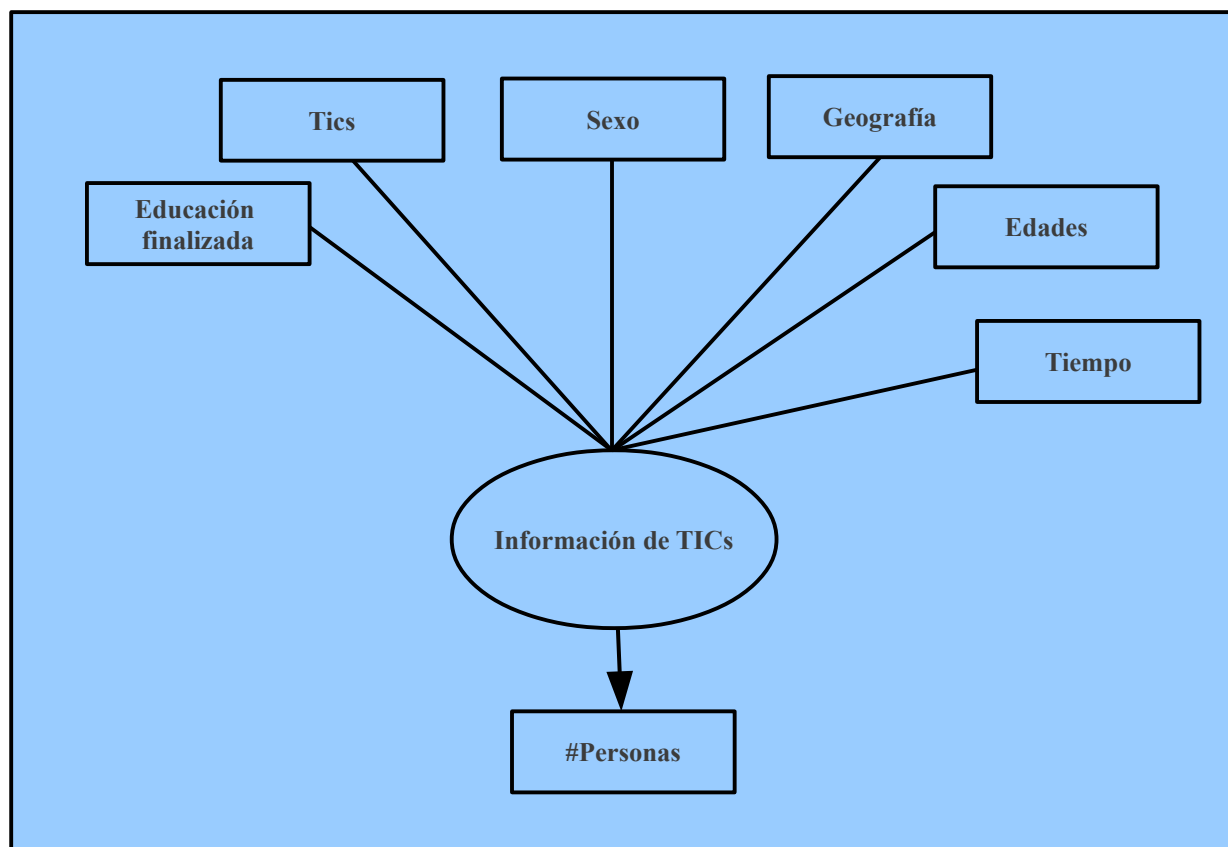
1. Geografía
2. Tiempo
3. Sexo
4. Edad
5. Educación (finalizada)
6. Utilización de TICs



Las dimensiones Geografía, Tiempo, Sexo, Edad y Educación fueron ya definidas y explicadas en los dos requerimientos anteriores, por lo que explicaremos a continuación únicamente la dimensión Utilización de TICs:

<p><b><u>Tics</u></b></p> <div data-bbox="380 401 613 535"> <p><b><u>Tics</u></b> celular usoInternet</p> </div>	<p>Interesa saber si la persona cuenta o no con teléfono celular, y si utiliza Internet (y su frecuencia de uso).</p> <p>Se define entonces la dimensión Tics que contempla estos puntos.</p>
--	---

La única medida identificada es la cantidad de personas. A continuación se presenta un esquema de la relación dimensional Información de Tecnologías, que modela el requerimiento funcional:



## 2.4. Requerimiento funcional adicional

Definimos un requerimiento adicional que creemos aporta información interesante para el análisis de la encuesta. El mismo consiste en analizar a las personas según su ubicación geográfica, el tiempo, su ascendencia racial, sus ingresos y su ocupación. Interesa saber la cantidad de personas en cada situación.

Para la ascendencia racial se consideran los siguientes tipos de raza: Afro o negra, Asiática o amarilla, Blanca, Indígena, otra. Las franjas de ingreso las agrupamos a consideración nuestra, se detallarán las franjas en la sección del diseño lógico.

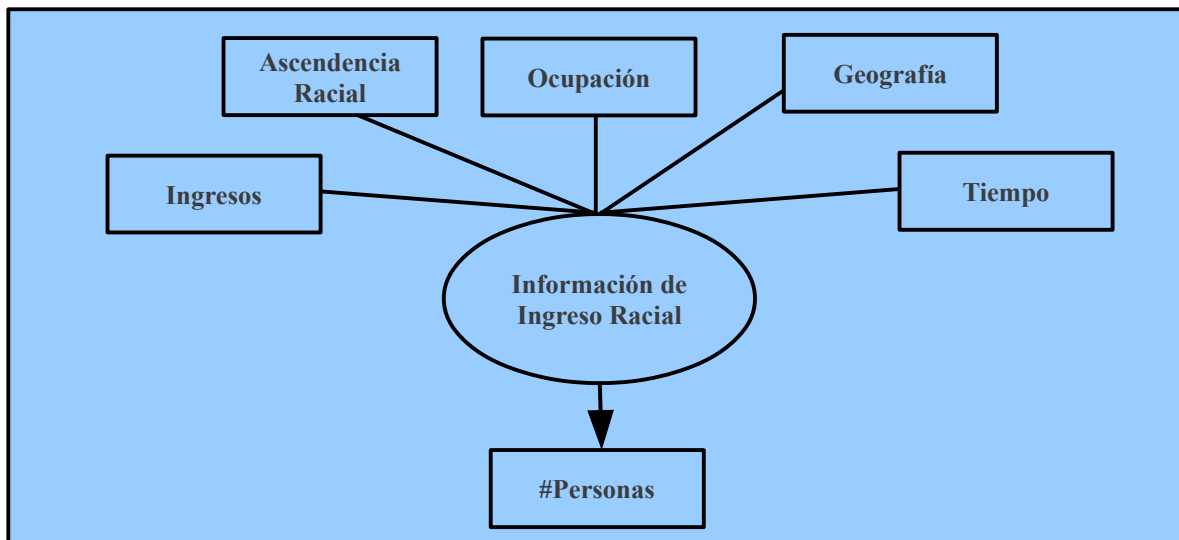
Por lo tanto, se identifica como única medida la cantidad de personas, y las siguientes dimensiones:

- 1.- Geografía
- 2.- Tiempo
- 3.- Ocupación
- 4.- Ascendencia Racial
- 5.- Ingresos

Las dimensiones Geografía, Tiempo y Ocupación son las mismas que se definieron en los requerimientos anteriores, por lo que nos encargamos de explicar a continuación únicamente las dimensiones Ascendencia Racial e Ingresos:

<div><u><b>Ascendencia Racial</b></u><div><u>AscendenciaRacial</u></div></div>	Se define la dimensión Ascendencia Racial de una sola jerarquía con un solo nivel, la cual considera los tipos de raza mencionados anteriormente.
<div><u><b>Ingresos</b></u><div><u>Rango de ingresos</u></div></div>	Se define la dimensión Ingresos que abarca nueve franjas de ingreso.

A continuación se presenta un esquema de la relación dimensional Información de Ingreso Racial, que modela el requerimiento funcional:



### 3. Diseño Lógico

Para el diseño lógico del sistema, además de considerar los metadatos y los datos de las ECH, estudiamos los requerimientos no funcionales que se derivan de las herramientas open source utilizadas. Con respecto a este punto, la única restricción que encontramos es que el servidor OLAP Mondrian provisto por Pentaho es de tipo ROLAP, por lo que nuestro diseño se restringe hacia un sistema data warehouse en base de datos relacional.

Se utilizó la herramienta MySQL Workbench para crear los diagramas de las tablas. Este programa nos fue de gran ayuda, ya que a partir de las tablas definidas y sus relaciones, Workbench genera un script SQL que crea la base de datos, tablas, etc, lo que supimos aprovechar al momento de realizar la carga de los datos.

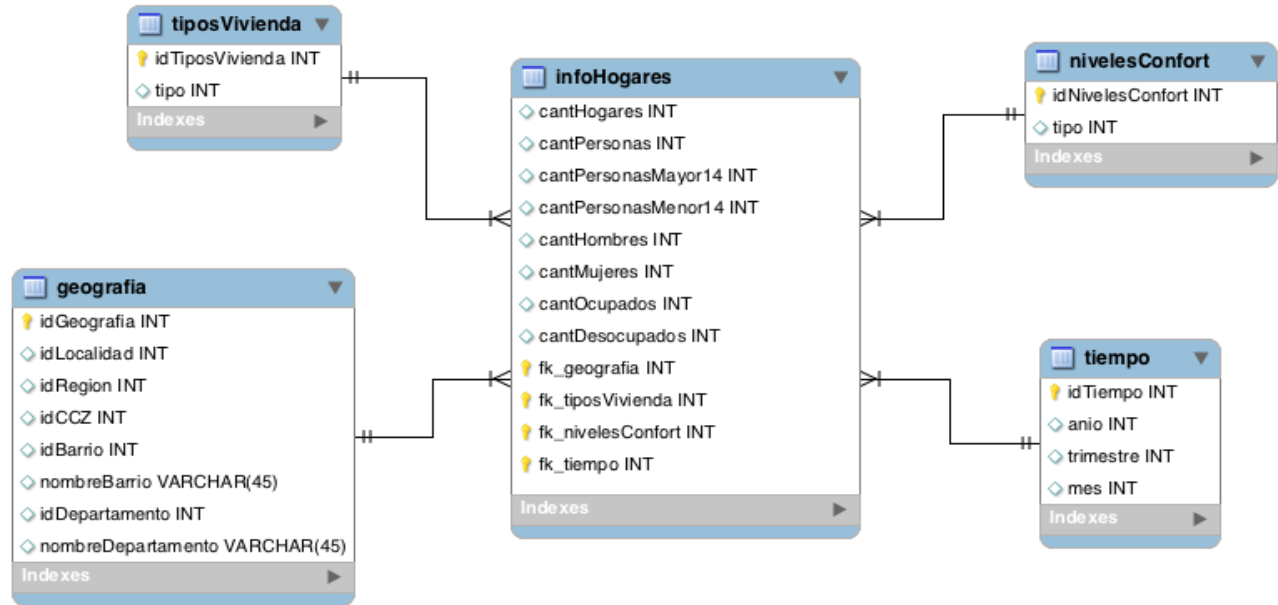
No se realizó fragmentación vertical ni fragmentación horizontal de dimensiones. A continuación describimos para cada requerimiento la materialización de las relaciones dimensionales realizadas. Se utilizó la estructura de esquema estrella, por lo que explicamos cómo se construyó cada tabla de hechos.

#### 3.1. Requerimiento funcional 1

Para este requerimiento se tienen las dimensiones Tiempo, Geografía, Tipo de Vivienda y Nivel de Vivienda y las medidas: cantidad de hogares en cada situación, cantidad de personas en total, cantidad de personas de 14 años o más, cantidad de personas menores de 14 años, cantidad de hombres, cantidad de mujeres, cantidad de ocupados y cantidad de desocupados. La relación dimensional que modela el requerimiento es Información de Hogares.

A partir de lo anterior, se creó una tabla para cada una de las cuatro dimensiones anteriores y la tabla de hechos. Esta última contiene una clave foránea que referencia a cada una de las otras tablas. A continuación se incluye una imagen con las tablas para la relación dimensional Información de

Hogares:



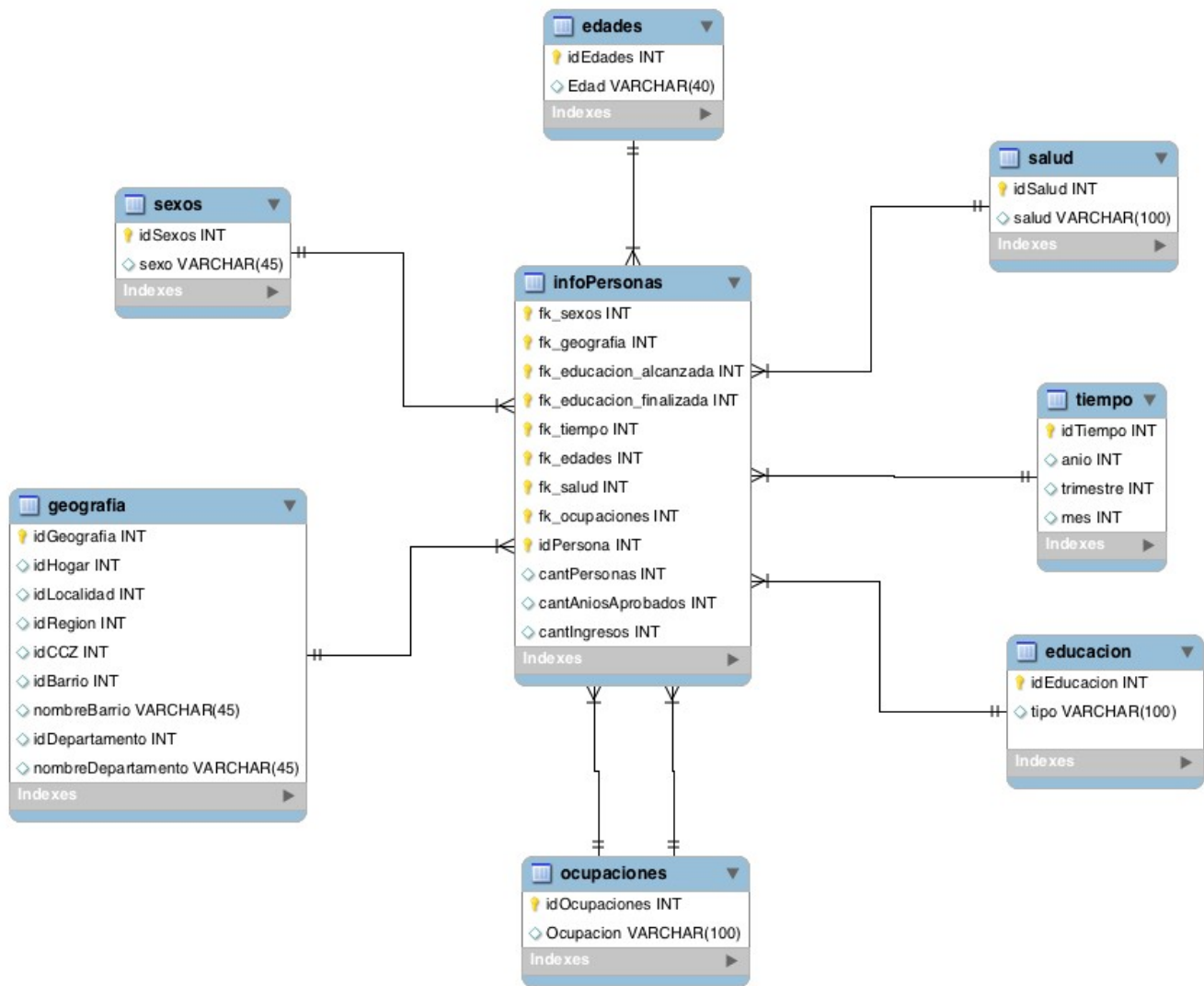
Como se aprecia en la figura, las tablas se encuentran desnormalizadas: la tabla de la dimensión Geografía contiene los atributos de todas sus jerarquías, así como también la tabla de la dimensión Tiempo.

Debido a que en los datos de la ECH de los años 2009, 2010 y 2011 se repiten hogares, la clave primaria de los hogares consiste en la concatenación del año de la encuesta con el número de hogar proveniente de los datos de la encuesta. A su vez, la clave primaria de la tabla de hechos es una clave compuesta por las claves foráneas de cada una de las tablas de dimensiones con que se relaciona.

### 3.2. Requerimiento funcional 2

Para este requerimiento se tienen las dimensiones: Ocupaciones, Educación (cursada y finalizada), Salud, Sexos, Geografía, Edades y Tiempo. Se tiene la medida Cantidad de Personas, y la relación dimensional Información de las Personas que modela el requerimiento.

A continuación se presenta el diagrama de tablas y sus relaciones:

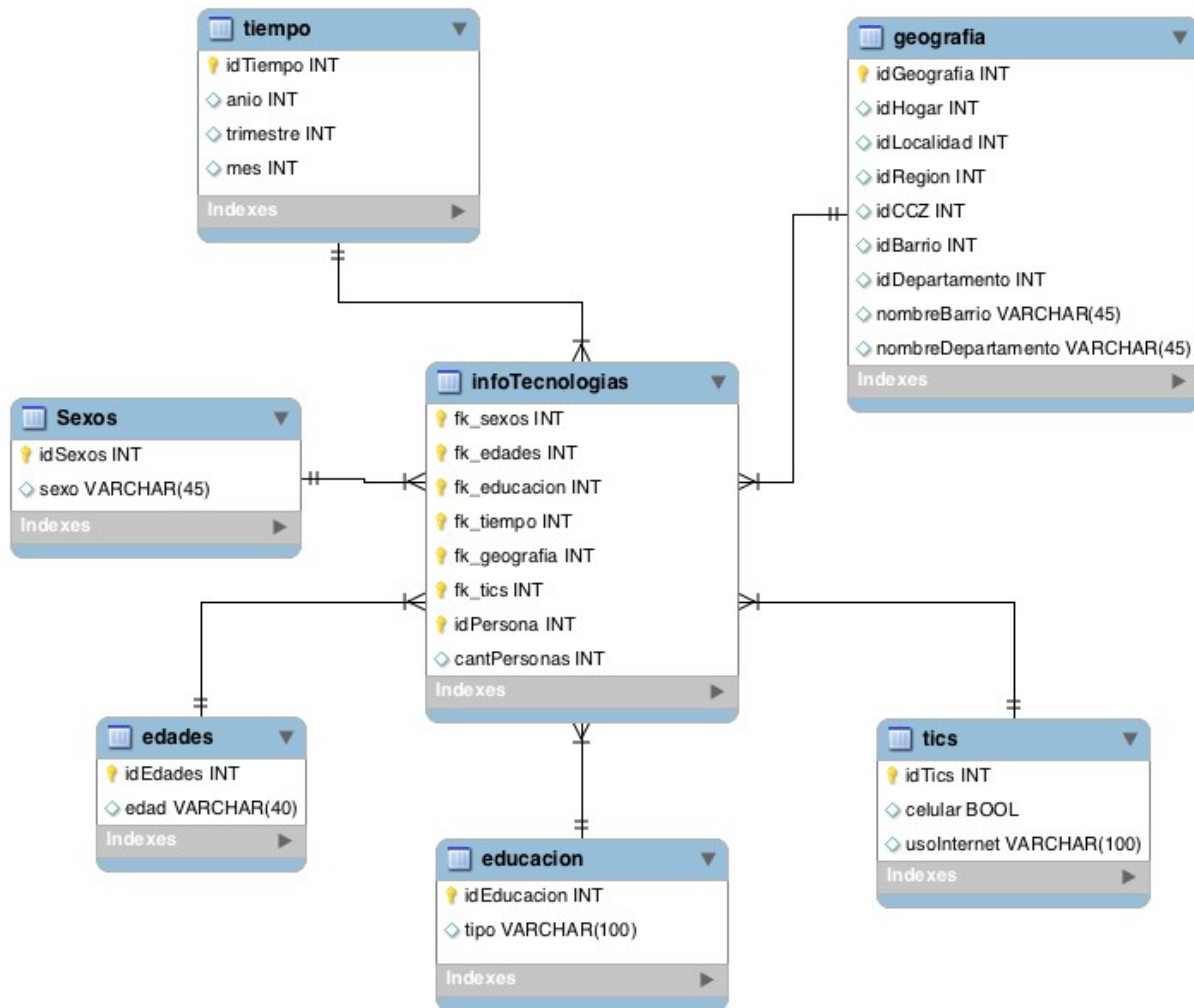


Cabe destacar que se agregó el identificador de cada persona en la tabla de hechos, ya que la clave primaria compuesta se puede repetir para personas de distintos hogares. Agregando el id de la persona (que es relativo al hogar), se soluciona este problema.

### 3.3. Requerimiento funcional 3

Para este requerimiento se definieron las dimensiones Geografía, Tiempo, Sexo, Edad, Educación, (finalizada) y Utilización de TICs. Se tiene la medida cantidad de personas y la relación dimensional Información de TICs que modela el requerimiento.

A continuación se presenta el diagrama de tablas y sus relaciones:

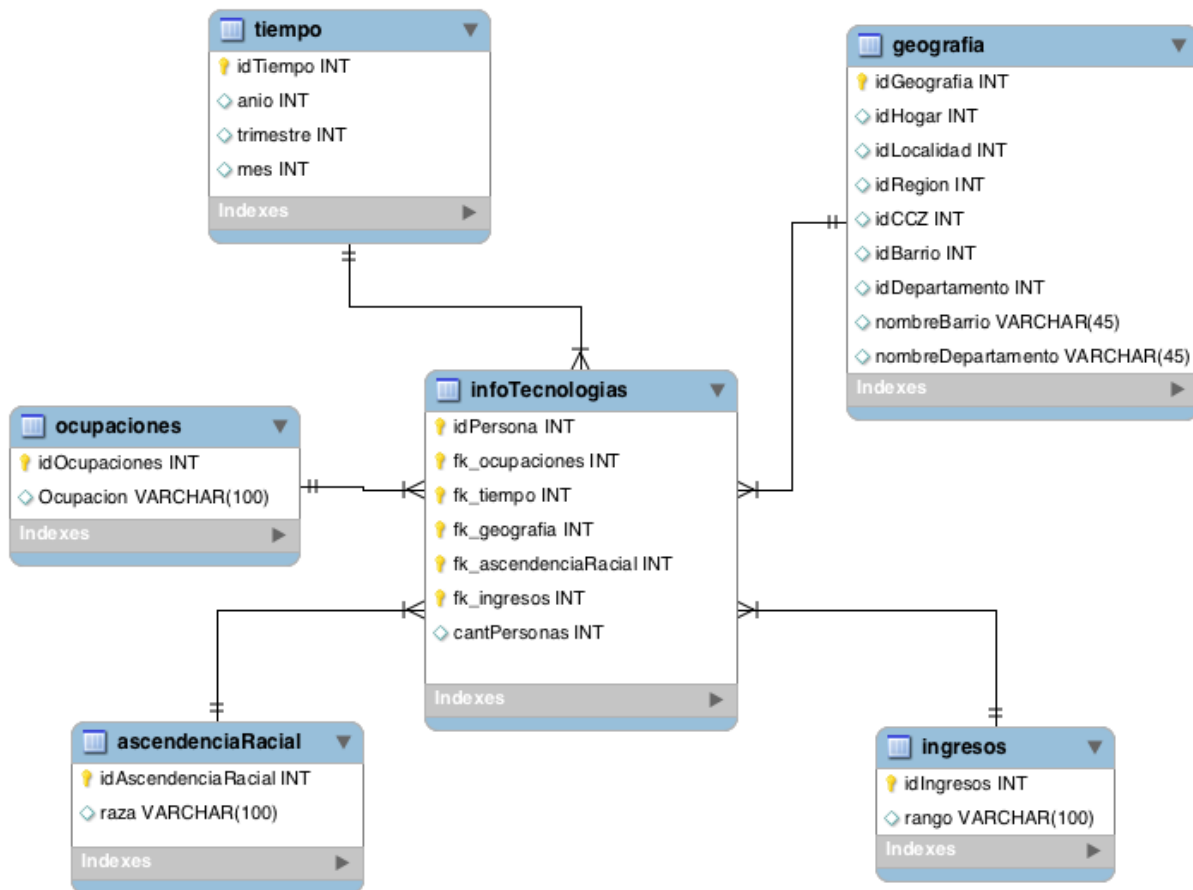


Al igual que para el la tabla de hechos del requerimiento funcional anterior, se debió incluir el identificador de la persona en la tabla, ya que se puede presentar el mismo problema.

### 3.4. Requerimiento adicional

Para este requerimiento se definieron las dimensiones Geografía, Tiempo, Ocupación, Ascendencia Racial e Ingresos. La única medida que se tiene es la cantidad de personas y la relación dimensional que modela el requerimiento es la Información de la Ingreso Racial.

A continuación se presenta el diagrama de tablas y sus relaciones:



#### 4. Implementación de relaciones dimensionales y dimensiones

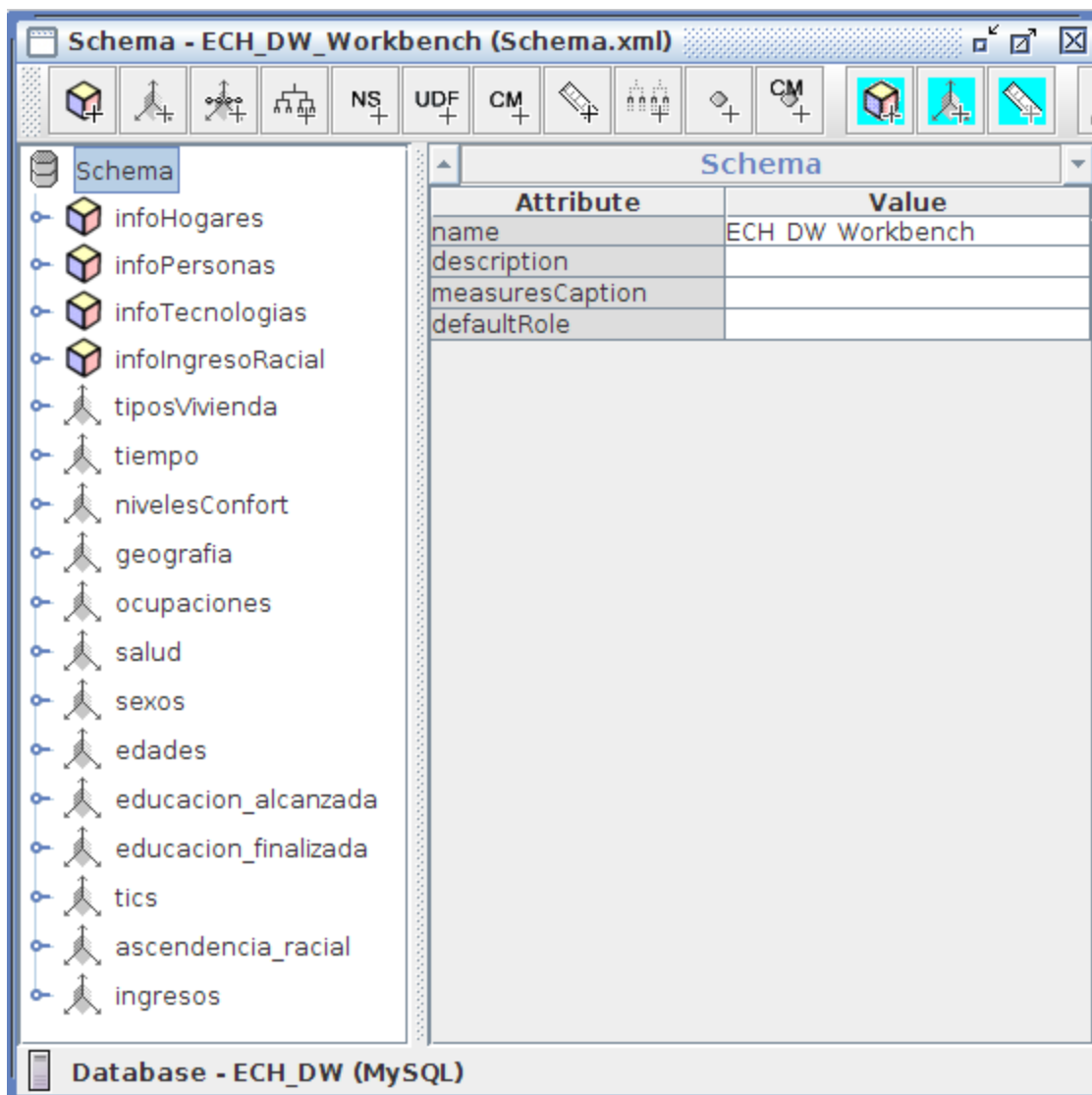
En primer lugar, previo a la carga debimos crear la base de datos que nombramos ECH\_DW en MySQL, utilizando los scripts generados por la herramienta MySQL Workbench como se dijo anteriormente. Esto es, a partir de los esquemas relacionales del diseño lógico, Workbench generó automáticamente un script SQL para cada relación dimensional que crea las tablas y relaciones en la base de datos.

Dichos scripts los modificamos y los juntamos en un solo archivo de nombre import.sql que se encarga de crear la base de datos, junto con todas las tablas del sistema.

Para la la carga de los datos, se utilizó en una primer instancia la herramienta de ETL Kettle. Para instalarlo simplemente se descomprimió el tar.gz correspondiente.

Luego, utilizando la herramienta Schema Workbench, generamos para cada relación dimensional, la representación de su cubo.

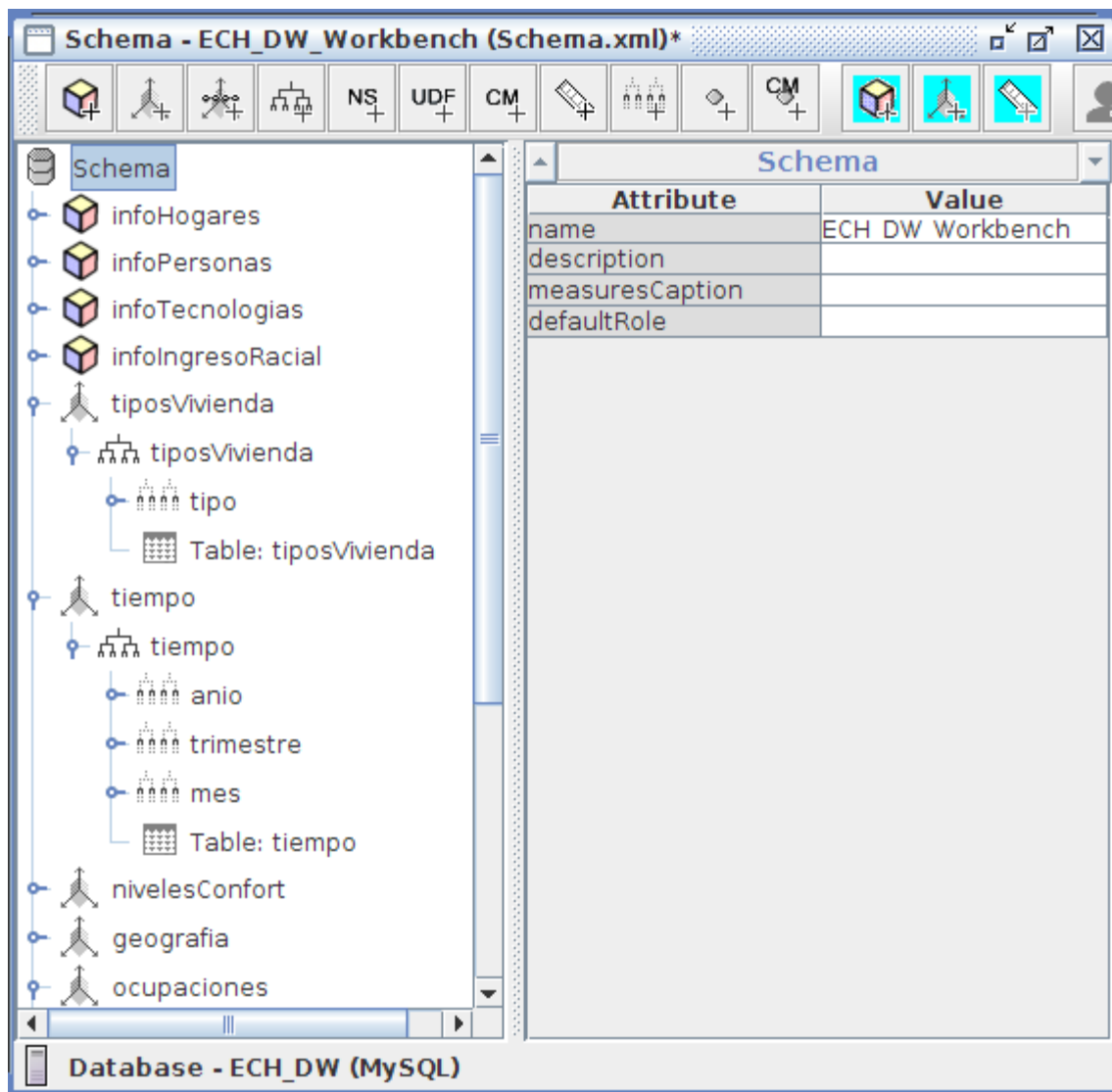
A continuación se muestra una imagen que representa cómo armamos el esqueleto del diseño de los cubos.



Como se puede apreciar en la imagen, utilizamos dimensiones compartidas. Dado que varios cubos utilizan las mismas dimensiones, optamos como decisión de diseño declarar todas las dimensiones como compartidas para que pudieran ser referenciadas desde cada cubo particular dependiendo si el mismo necesita utilizarla o no.

Para cada dimensión se configuraron sus jerarquías y sus niveles correspondientes según lo diseñado en el modelo conceptual. Por ejemplo, para la dimensión Tipos vivienda se generó una jerarquía con un sólo nivel. Mientras que para la dimensión Tiempo se generó otra jerarquía de tres niveles: Año, Trimestre y Mes. Esto puede verificarse en la imagen siguiente:





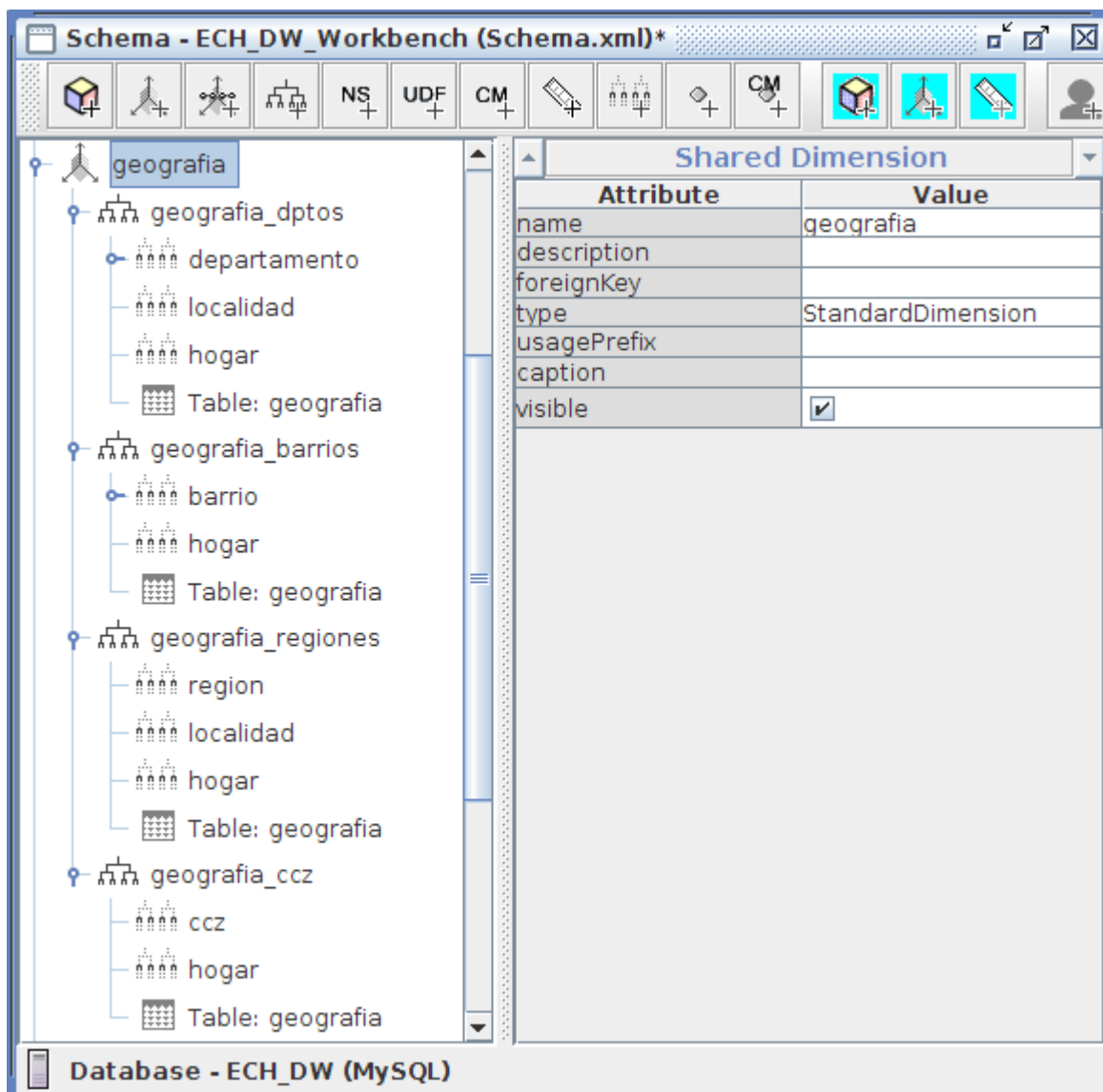
Como es de esperarse, el resto de las dimensiones fueron configuradas de manera análoga, respetando las decisiones de diseño tomadas en el modelo conceptual.

Para resolver el problema de modelado en lo que respecta a la dimensión Geografía y la relación N a N entre Localidades y Departamentos, y Localidades y Regiones, se tomó la decisión de generar varias jerarquías bajo la dimensión Geografía, donde cada jerarquía representa un camino desde el nivel más bajo (hogar) al nivel más alto alcanzable respetando relaciones N a 1. Entonces, se definieron las jerarquías geografia\_barrios, geografia\_ccz, geografia\_deptos, geografia\_regiones.

Si bien geografia\_dptos y geografia\_regiones no respetan las relaciones N a 1 al hacer roll up por Localidad, esta implementación nos permite ver de manera “limpia” las consultas sobre los barrios o los CCZ a la hora de realizarlas en el servidor ROLAP.

Para clarificar, a continuación se despliega una imagen que muestra lo diseñado con respecto a la

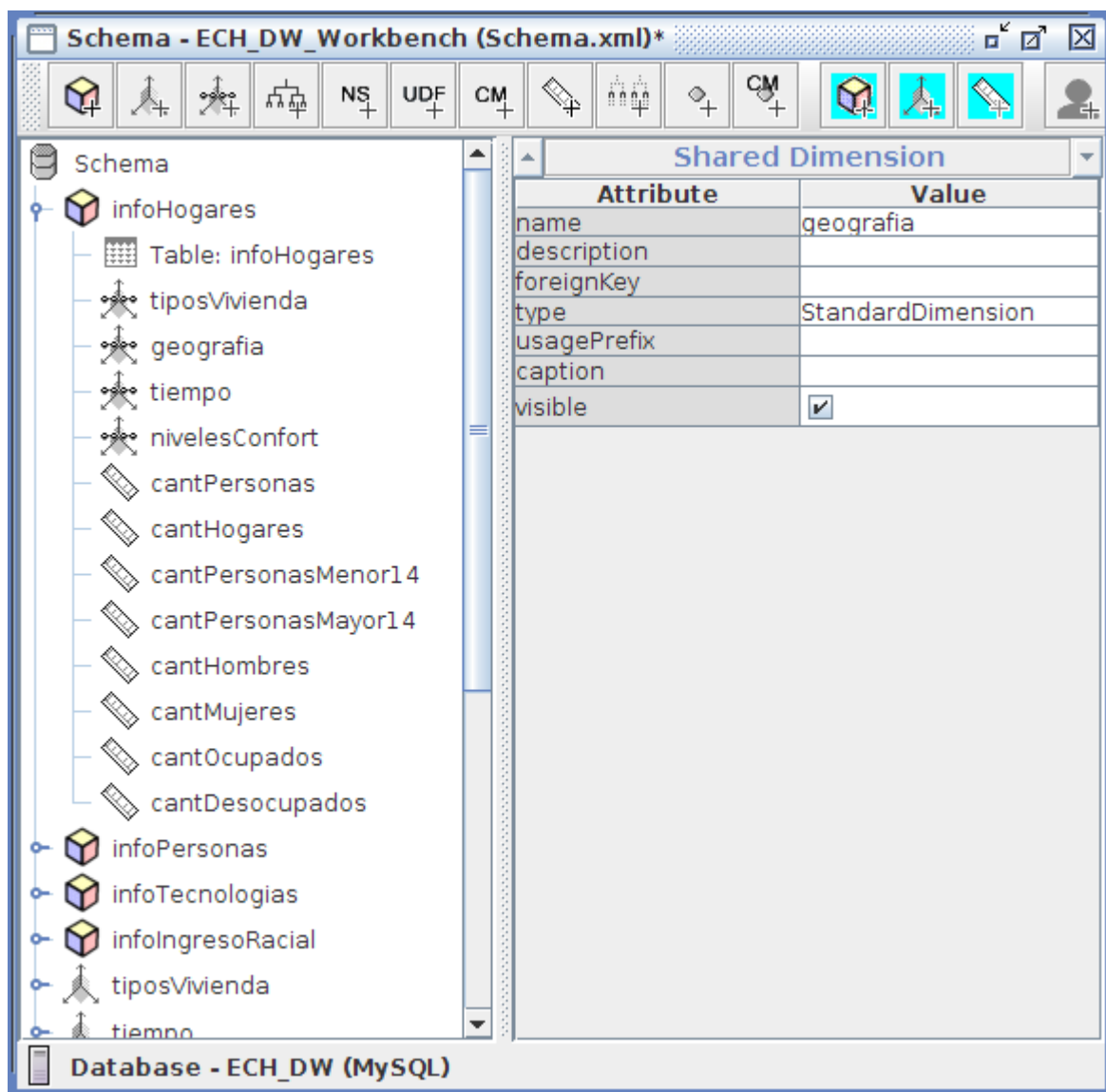
dimensión geografía:



Una vez diseñadas todas las dimensiones, se prosiguió a diseñar los cubos que las utilizan.

En cada cubo se definieron las medidas que debe soportar y las *Dimension Usages* correspondientes a las Dimensión Compartidas previamente definidas. Estas son básicamente referencias a las dimensiones compartidas.

A continuación se despliega otra imagen ejemplificando cómo luce la estructura de un cubo en Schema Workbench:



Para poder realizar consultas desde la interfaz de Pentaho, se publicó el esquema compuesto por los cubos definidos en el Schema Workbench.

## 5. Documentación del proceso de carga

Para la realización del proceso de carga se utilizó la herramienta Kettle de Pentaho.

Decidimos utilizar transformaciones para llevar a cabo los procesos de carga. Para llevar a cabo la carga de los datos involucrados el requerimiento 1, se realizaron varias transformaciones. En particular, realizamos una transformación por cada tabla pequeña y tres transformaciones para la fact table infoHogares y para la tabla geografia (una transformación por año – 2009, 2010 y 2011).

Para el resto de los requerimientos decidimos juntar algunas transformaciones en un sólo archivo por

comodidad de trabajo y para minimizar el volumen de archivos que se estaba generando.

Los steps más comunes que utilizamos para llevar a cabo las cargas fueron:

- CSV Input, para leer archivos CSV.
- Xbase Input, para leer archivos DBF.
- SQL File Output, para retornar scripts sql que implementen la carga de los datos a la base de datos.
- Merge Join, para hacer joins de archivos (fue necesario para los datos de las personas que venían en dos archivos separados).
- Modified JavaScript Value, para programar cálculos sobre los datos y la generación de nuevos atributos no existentes en los datos de entrada.
- Select Values, para seleccionar los atributos que nos interesaba retornar.
- Add Sequence, los datos del 2011 no tenían asignados ids para las personas, utilizamos este step para simular un identificador para cada persona.

Como el volumen de datos era tan importante, intentamos tomar decisiones que agilizaron lo más posible el proceso de carga. Por esto decidimos utilizar el step SQL File Output en vez de conectarnos directamente a la base de datos e ingresar los datos directamente desde el Kettle.

Esto nos permitía, mediante el análisis de los scripts generados, verificar que los datos fueran correctos antes de ingresarlos a la base.

Por la misma razón decidimos separar en transformaciones diferentes el proceso de carga de cada año en particular.

Los archivos que definen las transformaciones pueden encontrarse en el anexo bajo el directorio *transformaciones*.

## **5.1. Algoritmos de carga**

Los algoritmos de carga los realizamos separando por año. Es decir, nos encargamos primero de hacer los algoritmos que corresponden a la carga de los datos de la encuesta de 2009, siguiendo por los algoritmos para la carga de la encuesta de 2010 y finalmente de la encuesta de 2011.

Optamos por proceder de esta forma, ya que los metadatos de los tres años difieren en algunos puntos, particularmente los metadatos correspondientes al año 2011 varían sustancialmente en lo que refiere a los estudios y las ocupaciones de las personas encuestadas.

### **5.1.1. Carga de las tablas de cada dimensión**

Para cada dimensión, exceptuando la dimensión Geografía, debimos crear en primer lugar un archivo csv con los datos correspondientes.

En segundo lugar, creamos en Kettle archivos de transformación para cada carga, que consisten en dos steps básicamente: el primero que se encarga de leer los datos del csv, y el segundo que se encarga de generar un script SQL que crea las consultas de inserción para los datos leídos del primer step.

A continuación exponemos las consideraciones que tuvimos al momento de crear los archivos csv:

1. Para la tabla Tipos de Vivienda tuvimos en cuenta que tenemos solo tres tipos: casa, apartamento y otros (mediante strings).
2. Para la tabla Edades, definimos 10 rangos etáreos a nuestra consideración (mediante strings).
3. Para la tabla Niveles de Confort definimos cuatro tipos (mediante enteros del 1 al 4).
4. Para la tabla Educación definimos los 12 tipos que se encuentran en la letra del obligatorio mediante strings.
5. Para la tabla Ocupaciones definimos los 10 tipos de ocupaciones que se encuentran en la letra del obligatorio (mediante strings).
6. Para la tabla TICs definimos los 5 tipos de uso de Internet definidos en la letra, tanto para las personas que cuentan con celular como para las que no (10 filas en total).
7. Para la tabla de Salud tuvimos que considerar que, según los datos de las encuestas, las personas pueden estar afiliadas al mismo tiempo al MSP, a la salud privada, al IAMC u a otro tipo de sistema. Por lo tanto, tuvimos que considerar las 15 combinaciones posibles al momento de armar el archivo csv.
8. Para la tabla de Ingresos consideramos 9 rangos a nuestra consideración.

### 5.1.2. Carga de la tabla Información de Hogares

Para realizar la carga de esta tabla tuvimos en cuenta que los metadatos de la encuesta del año 2011 varían con los de las encuestas de 2009 y 2010 en lo que refiere al servicio doméstico. No existe en dicha encuesta la información referida al servicio doméstico que se pide en el requerimiento, por lo que para la carga de esta tabla no incluimos el servicio doméstico como factor determinante del nivel de confort de la vivienda para el año 2011.

Utilizamos el step *Modified JavaScript Value* para procesar los datos y generar atributos que contuvieran la información calculada de los valores que nos interesa aportar.

El algoritmo que calcula cada medida y las claves foráneas de la tabla no tiene más complicación que en el cálculo del nivel de confort de la vivienda. Para ello definimos los 4 niveles de confort según lo siguiente:

- Nivel 1 – Nivel de confort bajo
- Nivel 2 – Nivel de confort medio-bajo
- Nivel 3 – Nivel de confort medio-alto.
- Nivel 4 – Nivel de confort alto.

Entonces, para calcular el nivel de confort, se creó en el step de JavaScript un arreglo que contiene como entradas cada electrodoméstico definido en los metadatos. Por ende se tiene un arreglo que contiene únicamente los valores 1 y 2 (según los metadatos). Para calcular la cantidad de electrodomésticos del hogar recorreremos el arreglo preguntando por el valor 1 que se corresponde con contar con dicho electrodoméstico en el hogar.

Cabe destacar que, al momento del conteo de electrodomésticos, no tuvimos en cuenta ningún tipo de ponderación que agregue valor a un auto ante un lavarropa o un televisor, por ejemplo.

Consecuentemente, todos los electrodomésticos tienen la misma ponderación.

Una vez finalizado el conteo de electrodomésticos, pasamos a verificar el origen del agua y si se cuenta con servicio doméstico (y su frecuencia). En función de los valores obtenidos y de los niveles de confort previamente definidos seteamos el nivel de confort correspondiente.

El resto de las medidas se calcula en forma casi directa a partir de los datos que se obtienen de la salida del input inicial.

### **5.1.2. Carga de la tabla Información de Personas**

Para la carga de esta tabla se procedió de forma similar a la anterior.

Tuvimos en cuenta que los datos de las personas se encuentran en dos archivos separados para los tres años considerados, por lo que en Kettle tuvimos que incluir un step para cada transformación que se encargara de hacer un join de las tablas de ambos archivos. También utilizamos un step “Modified JavaScript Value” en el cual incluimos el algoritmo de carga de los datos obtenidos del join.

En el caso de la encuesta correspondiente al año 2011, los datos no aportan un atributo que identifique a la persona. Por esta razón, en el Kettle debimos agregar un step previo al step del join que se encarga de agregar un número de secuencia para cada fila de la tabla leída. De esta forma podemos hacer el join en forma exitosa entre las tablas de datos de personas de ambos archivos.

Este algoritmo no fue tan sencillo como el anterior, en particular porque los metadatos de la encuesta del año 2011 correspondientes a la educación varían significativamente con respecto a los de 2009 y 2010 y el modo que utilizamos para la carga de estos datos no fue el ideal a nuestro entender.

Empezamos por explicar cómo realizamos la carga de los datos del máximo nivel de educación alcanzada y finalizada de cada persona en Kettle.

Para ello definimos en el step de JavaScript un arreglo que tiene como entradas la cantidad de años cursados para cada nivel de educación, a partir de la salida del step de join de las tablas de información de personas. Destacamos que el primer elemento del arreglo es un 1, que se corresponde con que la persona no haya cursado nunca ningún nivel de educación.

Luego, recorreremos todo el arreglo preguntando si cada elemento es distinto de cero. En caso de ser distinto de cero, se almacena el índice actual (correspondiente al nivel de educación actual) en una variable auxiliar. De esta forma, al terminar de recorrer el arreglo, la variable auxiliar almacena el último nivel de educación cursado (en caso de ser igual a 1, significa que la persona nunca cursó ningún nivel de educación como se explicó anteriormente).

Para calcular el nivel de educación finalizada, se opero de manera similar. Pero se utilizó el dato que viene en la encuesta que dice si la personas finalizó el nivel de educación más alto indicado. Si lo finalizó, entonces es el mismo que el nivel alcanzado, sino, se recorre el arreglo desde el final hasta el comienzo para encontrar la segunda ocurrencia de educación realizada y se considera esa como el nivel más alto finalizado.

Para la carga de la suma de ingresos de cada persona creamos un arreglo cuyas entradas se corresponden con cada uno de los ingresos definidos en los metadatos de la encuesta. Luego, recorrimos el arreglo y acumulamos en cada iteración los ingresos de la persona.

En lo que refiere a las ocupaciones de las personas consideramos incluir, además de las ocupaciones definidas en la letra del obligatorio, aquellas personas cuya ocupación consiste en ser miembro del hogar no remunerado, así como también consideramos como personas inactivas aquellas que realizan quehaceres en el hogar, los estudiantes y los rentistas. Esto lo hicimos porque, al no considerar estos casos se nos generaba inconsistencias en la carga.

Finalmente, para la carga de los datos de la salud de cada persona, tuvimos que controlar que estuviera afiliado a más de un sistema al mismo tiempo.

### **5.1.3. Carga de la tabla Información de Tecnologías**

La información para cargar estas tablas se obtuvo de los datos de las personas, por lo tanto, se pudo reutilizar gran parte de los algoritmos de carga definidos para la carga de la tabla de información de Personas.

Primeramente se cargó la tabla *tics* que almacena los niveles de uso de internet posibles, discriminando también por la utilización o no de celular. Se utilizaron los mismos niveles de *uso de internet* que los definidos en la letra del obligatorio.

El cálculo del uso de internet y de celular para cada persona no tuvo mayores complicaciones, todos los datos se desprenden directamente de la encuesta, por lo que simplemente se generó un algoritmo que consultara si la persona utiliza o no internet y si utiliza con qué frecuencia lo hace, discriminando si utiliza también celular o no.

### **5.1.4. Carga de la tabla Información Ingreso Racial**

Al igual que para la tabla de información de tecnologías, la fact table de información de nivel de ingresos según ascendencia racial también se desprende de los datos arrojados por la información contenida en las encuestas de Personas.

Fue necesario generar dos tablas más: *ascendenciaRacial* e *ingresos*. La tabla *ascendenciaRacial* esta compuesta por los tipos de ascendencia que define la encuesta, los mismos son:

1. Afro o Negra
2. Asiatica o Amarilla
3. Blanca
4. Infigena
5. Otra

La tabla de ingresos se compone de franjas de ingresos que nosotros definimos.

El cálculo para cargar los datos de la fact table también fue prácticamente directo. Se utilizó la información de la encuesta para obtener la ascendencia racial de la persona y se reutilizó el cálculo de los ingresos. Dependiendo del monto del ingreso calculado, se seteó la foreign key correspondiente al nivel de franja de ingreso en el que la persona estaba incluida.

## 6. Reportes

Para la generación de reportes se utilizó la herramienta Report Designer de Pentaho.

Se realizaron cinco reportes distintos, consultando sobre diferentes tablas del Data Warehouse.

Para llevar a cabo los reportes se realizaron directamente consultas SQL que el Report Designer interpreta para generar el reporte.

### **Cantidad de Hogares y personas por Barrio**

Se realizó este reporte para contabilizar cuántos hogares fueron encuestados en cada barrio de Montevideo y cuántas personas en total fueron encuestadas en ese barrio. El reporte agrupa por barrio y despliega la información para cada uno de ellos.

La consulta SQL realizada fue la siguiente:

```
SELECT SUM(cantHogares), nombreBarrio, SUM(cantPersonas)
FROM infoHogares INNER JOIN geografia
ON ( infoHogares.fk_geografia = geografia.idGeografia )
WHERE ( geografia.nombreDepartamento = 'Montevideo' )
GROUP BY geografia.nombreBarrio;
```

### **Hogares, Hombres, Mujeres, Desocupados por Barrio**

Similar al anterior, este reporte distingue agrupando por Barrio de Montevideo, la cantidad de hombres, la cantidad de mujeres y la cantidad de desocupados que existen en cada barrio.

La consulta SQL utilizada fue la siguiente:

```
SELECT SUM(cantHogares), nombreBarrio, SUM(cantHombres), SUM(cantMujeres),
SUM(cantDesocupados)
FROM infoHogares INNER JOIN geografia
ON ( infoHogares.fk_geografia = geografia.idGeografia )
WHERE ( geografia.nombreDepartamento = 'Montevideo' )
GROUP BY geografia.nombreBarrio;
```

### **Nivel de Ingreso por Persona y Raza, según Ocupación y Hogar**

Se generó otro reporte para informar el tipo de ocupación de cada persona, su ascendencia racial y el rango de ingresos de la misma, agrupando por hogar.

La consulta SQL fue la siguiente:

```
SELECT infoIngresoRacial.fk_geografia, infoIngresoRacial.idPersona,
ascendenciaRacial.raza,
ocupaciones.ocupacion, ingresos.rango
```



```
FROM infoIngresoRacial INNER JOIN ascendenciaRacial ON
(infoIngresoRacial.fk_ascendencia_racial = ascendenciaRacial.idAscendenciaRacial)
INNER JOIN ocupaciones ON (infoIngresoRacial.fk_ocupaciones =
ocupaciones.idOcupaciones)
INNER JOIN ingresos ON (infoIngresoRacial.fk_ingresos = ingresos.idIngresos)
```

### **Promedio de ingresos por persona discriminando su nivel educativo**

Este reporte agrupa por nivel educativo finalizado de la persona, la cantidad de personas en esa situación, la cantidad de ingresos sumada de las personas en esa situación y a partir de eso el promedio de ingresos por persona en esa situación educacional.

La consulta SQL utilizada fue la siguiente:

```
SELECT SUM(infoPersonas.cantPersonas), SUM(infoPersonas.cantIngresos),
ROUND(AVG(infoPersonas.cantIngresos)), educacion.tipo, educacion.idEducacion
FROM infoPersonas
INNER JOIN educacion ON (infoPersonas.fk_educacion_finalizada =
educacion.idEducacion)
GROUP BY educacion.tipo ORDER BY educacion.idEducacion ASC;
```

### **Uso de Internet según rango de edades**

Por último, se generó un reporte que agrupa por franjas etarias y cuenta cuantas personas utilizan internet de determinadas maneras, considerando sólo personas que utilizan celular.

La consulta fue:

```
SELECT edades.idEdades, tics.idTics, edades.Edad, tics.usoInternet,
sum(infoTecnologias.cantPersonas)
from infoTecnologias INNER JOIN tics
ON (infoTecnologias.fk_tics = tics.idTics) INNER JOIN edades
ON (infoTecnologias.fk_edades = edades.idEdades)
where (tics.celular = 1)
group by edades.idEdades, tics.idTics
```

Nota: Todos los reportes se entregan en el anexo, por problemas con el tamaño de los mismos, solo se entrega, a modo de ejemplo, la primer carilla de cada uno.

## **7. Testing de la solución**

Encaramos el test de la solución en dos etapas. En primer lugar, durante el proceso de carga de los datos fue necesario testear que cada script SQL generado fuera correcto, y por otro lado que la solución y los reportes generados por la herramienta ROLAP fueran coherentes con los datos arrojados por la encuesta.

No nos fue posible generar tests automatizados para esto, tuvimos que realizarlos manualmente cada vez que se realizaba un cambio.

En la primer etapa fue necesario centrarse en el testeado de datos. Tanto los generados por nosotros a partir de la herramienta ETL como los descargados desde el INE (nos encontramos con algunas inconsistencias entre lo que se definía en los metadatos y lo que en definitiva estaba en los datos).

En los procesos de carga, se definieron algunas *flags de error* para verificar que todo estuviera bien. Esto es, inicializar un atributo que esperamos que tenga un valor válido con un valor inválido y fácilmente identificable, entonces si ese valor inválido aparece en algún momento en los datos generados, sabemos que algo anduvo mal.

Mediante el comando *grep* de Linux era muy sencillo buscar las flags de error en los scripts de inserción de datos generados por la herramienta ETL. De esta forma pudimos testear que los datos a insertar en el Data Warehouse eran correctos antes de insertarlos, solamente mirando el script.

Por otro lado, se tuvieron que realizar test funcionales (manuales) mediante la generación de consultas en Saiku o en JPivot. Simplemente se generaron consultas y se verificaron sus resultados realizando consultas análogas en la base de datos, o bien verificando que sumas realizadas mediante cruzamientos distintos daban lo mismo, en los casos que tuviesen que dar lo mismo.

## **8. Conclusiones y dificultades encontradas**

Creemos que el trabajo propuesto fue muy interesante, en el que nos tuvimos que enfrentar con problemas reales, volúmenes enormes de datos, presentados en diferentes formatos y distinta cantidad de archivos, generación de bases de datos desnormalizadas, aprendizaje de herramientas de todo tipo, comunicación entre distintos componentes de software, etc.

Descubrimos en las herramientas aplicadas una potencia muy importante, sobretodo en la herramienta ETL que puede ser utilizada para muchos otros tipos de problemas aparte de los relacionados con los Data Warehouses, y también las distintas variantes en los usos de la herramienta OLAP y su capacidad para responder ante las distintos conjuntos de datos para generar consultas dinámicamente.

Resulta muy interesante ver el producto finalizado y entender la importancia de la posibilidad de manipular los datos con los que se cuenta. La capacidad de obtener información de la índole de la que se manejó en el obligatorio y de cualquier tipo y tenerlo tan al alcance de la mano una vez finalizado el proceso de desarrollo, con la maleabilidad que es posible brindar utilizando los conceptos vistos, es realmente muy importante.

Durante el transcurso de la realización del proyecto nos encontramos con varias dificultades, por lo que creemos conveniente listar algunas de ellas a continuación.

En primer lugar tuvimos que prestar mucha atención al momento de realizar el diseño conceptual y lógico del sistema, ya que contábamos con un conjunto de datos muy grande y los metadatos de la encuesta no eran lo suficientemente claros por momentos. En particular, para la definición de la dimensión geografía, tuvimos que observar en detalle los datos para poder diferenciar en forma correcta las jerarquías y sus niveles.

Con respecto a los metadatos de las encuestas encontramos diferencias que nos complicaron al momento del diseño lógico y de la carga de datos. Para ejemplificar, en la encuesta del año 2011, nos enfrentamos a que la misma no incluía en sus metadatos las características relacionadas con el servicio doméstico referidas en el requerimiento 1.

A su vez, la misma encuesta tiene en sus metadatos información de la educación finalizada para cada nivel de educación a excepción de la educación pre-escolar. Tampoco indica la cantidad de años aprobados en para el este nivel educativo. Para los niveles de educación primaria y media se incluyen los años de aprobación para cada tipo dentro de educación primaria y media, sin embargo incluye un campo único que indica si finalizó el nivel pero no distingue por el tipo de cada nivel.

Por otro lado, surgieron también varias dificultades y problemas técnicos al momento de utilizar las herramientas open source provistas por Pentaho. En particular el uso de Mondrian y Kettle al momento de la carga de los datos y de la comunicación entre componentes. Si bien las herramientas demostraron ser muy efectivas para el propósito del obligatorio, la información disponible en la web de Pentaho no es muy abundante al momento de toparse con errores o dudas, y dicha información se encuentra muy dispersa por la web de las herramientas.

Por este motivo tuvimos que recurrir en reiteradas oportunidades a varios foros para resolver dudas muy particulares o consultar a compañeros de otros grupos si pasaron por alguna situación similar.

Para finalizar, deseamos expresar nuestra conformidad con el trabajo obligatorio. No hay dudas que logramos aprender durante el curso conceptos nuevos de data warehouses que no conocíamos, y la realización del obligatorio nos permitió reafirmarlos. En particular, las herramientas utilizadas nos parecieron adecuadas para desarrollar el proyecto, a pesar de no contar con mucha información al momento de enfrentarnos a errores o problemas técnicos.

Lamentamos no haber podido incorporar en el proyecto la realización de gráficas clickeables ni la integración de los datos con Google Maps, pero el poco tiempo que dispusimos para dedicarle al obligatorio nos limitó bastante al momento de intentar incorporar estas funcionalidades (ambos integrantes del obligatorio trabajamos full time y cursamos tres materias más además de la presente).

## Bibliografía

- [1] Instituto Nacional de Estadística, <http://www.ine.gub.uy>
- [2] Microdatos y datos de las ECH del INE, <http://www.ine.gub.uy/microdatos/microdatosnew2008.asp>
- [3] Diccionario y referencias de las ECH del INE, <http://www.ine.gub.uy/microdatos/diccionariosech2008.asp>
- [4] Principales resultados de la ECH de 2010, INE, [http://www.ine.gub.uy/biblioteca/Encuesta%20Continua%20de%20Hogar es/Publicaci%C3%B3n%20Principales%20Resultados%202010.pdf](http://www.ine.gub.uy/biblioteca/Encuesta%20Continua%20de%20Hogar%20es/Publicaci%C3%B3n%20Principales%20Resultados%202010.pdf)
- [5] Pentaho Business Intelligence Community, <http://community.pentaho.com/>
- [6] Configuración de Pentaho sobre otros RDBMS, <http://wiki.pentaho.com/display/ServerDoc2x/Configuring+the+Platform+for+Other+Databases>
- [7] MySQL Workbench, <http://www.mysql.com/downloads/workbench/>
- [8] Kettle, <http://sourceforge.net/projects/pentaho/files/Data%20Integration/4.2.1-stable>
- [9] Pentaho Report Designer, <http://sourceforge.net/projects/jfreereport/files/04.%20Report%20Designer/3.8.3-stable>
- [10] Saiku, <http://analytical-labs.com/>
- [11] Creación del esquema XML en Pentaho Mondrian, [http://mondrian.pentaho.com/documentation/schema.php#XML\\_Cube](http://mondrian.pentaho.com/documentation/schema.php#XML_Cube)
- [12] Creación de Reportes, <http://wiki.pentaho.com/display/ServerDoc1x/Creating+Reports+using+Adhoc+Reporting>
- [13] Material teórico de la materia, <http://www.fing.edu.uy/inco/cursos/disDW/>
- [14] Stack Overflow, <http://stackoverflow.com/>