Universidad Nacional Autónoma de México

FACULTAD DE CIENCIAS

“Introducción a la minería de datos en la plataforma KNIME”

T E S I S

Para obtener el título de Actuario

P r e s e n t a:

FRANCISCO JARAMILLO REYES

Director: Dr. Miguel Ehécatl Morales Trujillo

CDMX **2016**

Datos del Jurado:

1. Dra. Hanna Jadwiga Oktaba

2. M. en I. Gerardo Avilés Rosas

3. Dr. Miguel Ehécatl Morales Trujillo (tutor)

4. M. en C. Eréndira Miriam Jiménez Hernández

5. M. en C. María Guadalupe Elena Ibargüengoitia González

Tabla de contenido

Índice de Figuras

Índice de Tablas

1. Introducción

1.1 Minería de Datos

La Minería de Datos o *Data Mining (DM)* es considerada una disciplina científica en crecimiento y de interés internacional que integra áreas como la Estadística, Inteligencia Artificial, Bases de Datos, Aprendizaje de Máquinas, Reconocimiento de Patrones y Visualización de datos.

> *El primer taller especializado en minería de datos, KDD-1989, se realizó durante la Onceava Conferencia Internacional Conjunta en Inteligencia Artificial en Detroit, Michigan, en agosto de 1989 (las siglas KDD significan Knowledge Discovery in Databases). El taller tuvo una gran respuesta a nivel mundial (se recibieron propuestas de doce países en cuatro continentes) y una gran asistencia, formada por investigadores, así como representantes del gobierno y de la industria. El reporte del taller se puede consultar en la revista de inteligencia artificial AI Magazine (volumen 11, número 5).* [1]

Actualmente la asociación responsable del taller se conoce como la *Association for the Advancement of Artificial Intelligence* (AAAI).

La popularidad del tema creció significativamente en los años 90´s, periodo en el que se integra el grupo *SIGKDD (Special Interest Group on Knowledge Discovery in Data and Data Mining)*[2], el cual es uno de los 37 grupos de interés de *ACM (Association for Computing Machinery)* y cuyo congreso anual es reconocido como el más importante en el presente:

- *ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*

[1] Tesis Posgrado, Martinez María del Rosario Cruz. "Minería de Datos Multiperspectiva". México, D.F. , 2007.
[2] *SIGKDD,* Junio 2015. [En línea]. Disponible: http://www.kdd.org/

A medida que pasaba el tiempo, creció el interés por parte de grupos de investigadores, profesionales y estudiantes que buscan integrarse a esta área multidisciplinaria. Existen otros congresos y reuniones vigentes en diferentes partes del mundo que contribuyen a la divulgación, desarrollo e investigación de la Minería de Datos[3], entre ellos podemos mencionar las siguientes:

- *ICMLA International Conference on Machine Learning and Aplications*
- *MLDM International Conference on Machine Learning and Data Mining*
- *SIAM International Conference on Data Mining*
- *ICDE International Conference on Data Engineering*
- *CIKM International Conference on Information and Knowledge Management*
- *PAKDD Pacific-Asia Conference on Knowledge Discovery and Data Mining*
- *WSDM Web Search and Data Mining*
- *AusDM Australasian Data Mining Conference*
- *CIARP Iberoamerican Congress on Pattern Recognition*
- *ECML PKDD European Conference on Machine Learning & European Conference on Principles and Practice of Knowledge Discovery in Databases*
- *SIGMOD-SIGACT-SIGART Symposium on Principles of Database Systems*
- *IEEE Symposium on Computational Intelligence and Data Mining*
- *IEEE Symposium on Visual Analytics Science and Technology*
- *IEEE International Conference on Data Mining*
- *ASA American Statistical Association*
- *SAI The Science and Information Organization*
- *ACIT Arab Conference on Information Technology*

La Minería de Datos nace tras la capacidad de almacenamiento masivo en bases de datos, por lo que es usual que se entienda como sinónimo de *Knowledge Discovery in Databases* (KDD). Minería de Datos y KDD son conceptos enfocados al propósito

[3] *KD Nuggets,* Junio 2015. [En Línea]. Disponible: http://www.kdnuggets.com/2013/11/top-conferences-data-mining-data-science.html

común de extraer información oculta en los datos. Sin embargo, Minería de Datos es una fase o subproceso del KDD. Así lo declaran Fayyad, Piatetsky-Shapiro y Smyth, miembros de *SIGKDD,* en su artículo titulado *The KDD process for extracting useful knowledge from volumes of data* en 1996[4].

En la Figura 1.1 se muestra el proceso KDD que consta de nueve pasos principales:

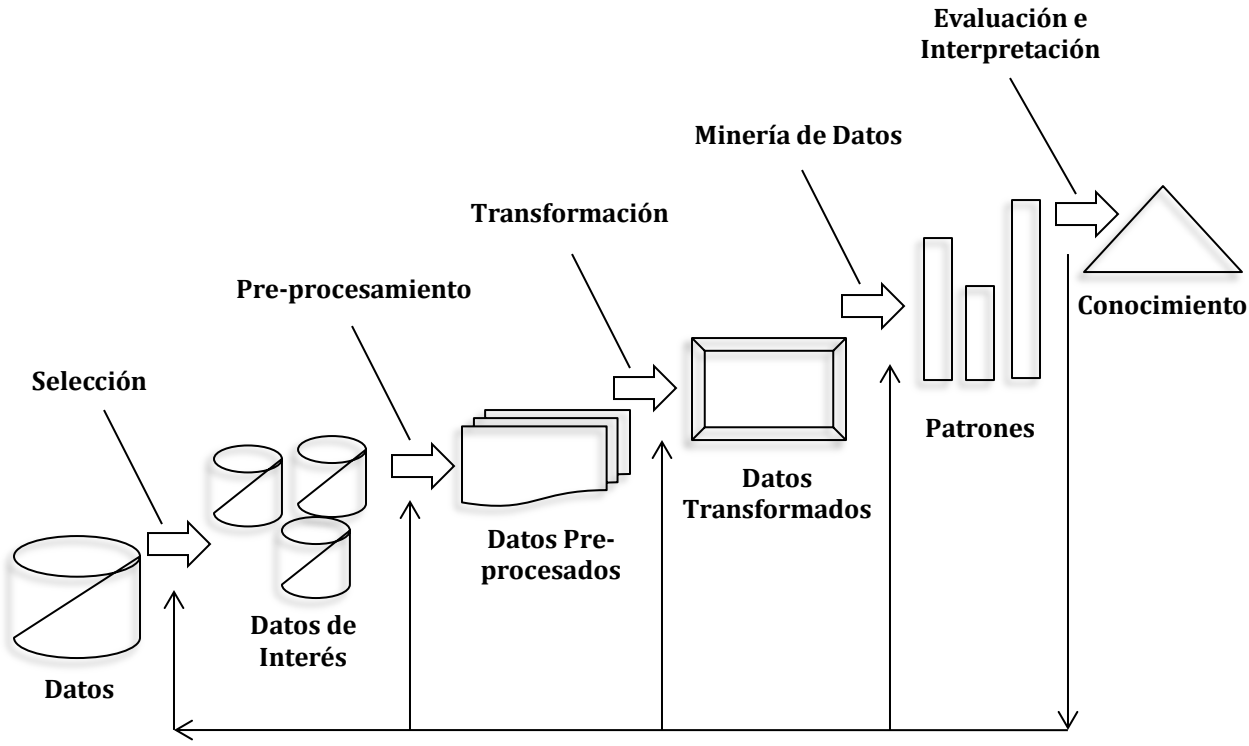


Fig. 1.1 Proceso KDD

1. **Desarrollar y comprender la aplicación**: este paso se refiere al conocimiento a priori de los objetivos de la aplicación.
2. **Crear un conjunto de datos objetivo:** se seleccionan las variables y/o muestras que participarán en las tareas de extracción de información.
3. **Limpiar y pre-procesar:** consiste en remover datos atípicos, ruido y completar los valores faltantes.

[4] U. Fayyad, G. Piatetsky-Shapiro y P. Smyth. *The KDD process for extracting useful knowledge from volumes of data. Volume 39 Issue 11, Nov. 1996 Pages 27-34. Magazine Communications of ACM.*

4. **Reducir y transformar los datos:** consiste en encontrar variables relevantes aplicando métodos de reducción y transformación de los datos.
5. **Elegir la tarea de minería de datos:** se empatan los objetivos previstos en el paso 1 con el tipo de técnica según su enfoque (clasificación, regresión, agrupamiento, etc.)
6. **Elegir el algoritmo de minería de datos:** se define el modelo y los parámetros apropiados.
7. **Realizar la minería de datos:** en este paso es donde se generan los patrones para el caso del particular en cuestión (reglas de clasificación, árboles de decisión, modelos de regresión, tendencias, etc.).
8. **Interpretar los patrones minados:** se lleva a cabo el análisis de lo obtenido y visualización de la información.
9. **Integrar el conocimiento descubierto:** en este paso se incorpora el nuevo conocimiento; documenta, reporta y/o evalúa el desempeño del sistema con los cambios y las soluciones o ventajas que se generaron.

Es importante resaltar que los autores expresan que en cada uno de los pasos del modelo es posible volver atrás los pasos necesarios para que el proceso sea exitoso.

1.1.1 Dominio del proceso

KDD comenzó como un proceso orientado a fuentes de información estructurada como lo son las bases de datos relacionales. Sin embargo, debido a la gran variedad de formatos y estructuras que existen, ha sido renombrado por algunos autores *como Knowledge Discovery Process* (KDP) para generalizar y estandarizar el domino moderno del proceso. Se extiende a conjuntos de datos (*Data Sets*) semi-estructurados y no-estructurados [1] [2].

El sentido de categorizar a los conjuntos de datos proviene de haberles dado estructura, orden e integridad cuando se almacenaron en bases, por tal motivo, se les clasifica como datos estructurados a las bases de datos relacionales y a los modelos multidimensionales. Por otra parte, los datos semi-estructurados son archivos como XLS, CSV, TXT, XML, ARFF, entre otros. Además, se les denomina no-estructurados (o

de estructura compleja) a colecciones de documentos de texto, multimedia y los datos "regados" en diversas plataformas de gestión (redes sociales, ERP, CRM, etc.).

1.1.2 Técnicas y definición

El material de Minería de Datos más difundido, estudiado y aplicado ha sido particularmente el que concierne a rutinas que se enfocan a datos estructurados y semi-estructurados. Para este tipo de dominio los métodos básicos utilizados se dividen en seis principales tareas a realizar en el descubrimiento de información[5]:

- Asociación

En términos generales consiste en encontrar o descartar correlaciones entre dos o más elementos u objetos contenidos en las bases de datos con la finalidad de predecir un patrón de comportamiento. Un ejemplo es la relación de compra entre productos en un supermercado. Los métodos son identificados como Minados de Reglas de Asociación.

- Clasificación

Es probablemente la técnica más popular y se refiere a la tarea de marcar o predecir un tipo de etiqueta usualmente llamada clase para un conjunto de instancias. Los principales métodos de clasificación son Redes Bayesianas, Vecino Más Cercano, Máquinas de Soporte Vectorial y Árboles de Decisión (siendo el algoritmo más reconocido el árbol C4.5)[6].

- Agrupamiento

Es una tarea que consiste en aplicar una regla de asociación a una población (o conjunto de datos) con la finalidad de determinar grupos cuyos individuos tienen

[5]IBM, *Data Mining Techniques.* Junio 2015 [En línea]. Disponible: http://www.ibm.com/developerworks/library/ba-data-mining-techniques/

[6] *International Journal of Engineering Trends and Technology (IJETT). Performance Evaluation of clustering Algorithms. Volume4 Issue7.* Julio 2013.

mayor similitud entre sí. Los métodos más conocidos son K-Medias, K-Medoids, CLARA, Jerárquicos Aglomerativos y Divisivos.

- Reconocimiento de patrones

Esta tarea es usada para encontrar tendencias o identificar sucesos recurrentes en eventos similares. Se basa en las **reglas de asociación** valiéndose de estas para definir prototipos provenientes de una muestra de entrenamiento donde la categoría ya es conocida, y así, poder clasificar un comportamiento en una clase existente o bien asignarla a una nueva. Una aplicación es el comportamiento de compra de un cliente en el supermercado.

- Predicción

Se refiere esencialmente la observación de las fluctuaciones que sufre la variable de estudio a razón del tiempo. Un tipo de modelo muy claro es una serie de tiempo (Procesos Autorregresivo Promedios Móviles, Autorregresivo Integrado de Promedios Móviles y Autorregresivo Integrado Promedios Móviles Estacional, etc.), así mismo, el modelo de regresión en el caso bi-variado. La diferencia con las técnicas de clasificación es que estas últimas tienen un carácter de estudio transversal, es decir, se realizan en una ventana de tiempo determinada y no existe predictibilidad explicita de la variabilidad a través del tiempo.

- Visualización

Esta tarea recurre a las características topológicas como regiones de predictibilidad o funciones que presenta la información, depende en gran parte de la capacidad del software para proyectar los datos en gráficos y/o tablas.

Otra técnica aplicada a una estructura de datos compleja que es incluida en algunas plataformas de Minería de Datos es la **Minería de Texto** que consiste en extraer información en forma condensada de documentos de texto digitalizados, induciendo la información útil desde su estructura lingüística [3].

Por otro lado, la evolución del concepto de Minería de Datos ha alcanzado la idea de "minar" todo tipo de datos. Si bien ya existen una amplia gama reconocida y aplicada de técnicas para minar datos estructurados y semi-estructurados, todavía existen grandes retos en el desarrollo y mejora de modelos de **Minería de Imágenes y Video** los cuales tienen estructuras más complejas[7].

Tomando en cuenta lo anterior, la definición general para Minería de Datos la declararemos como una serie de mecanismos y técnicas reconocidos por un software con el objetivo de encontrar información oculta en grandes volúmenes de datos.

1.2 Estándares de Minería de Datos

Existen estándares definidos por expertos en el tema para llevar a cabo el proceso de Minería de Datos, con la finalidad de lograr las mejores prácticas y resultados exitosos. Los más populares son *Cross Industry Standard Process for Data Mining* (CRISP-DM) y SEMMA (acrónimo de las palabras *Sample*, *Explore*, *Modify*, *Model*, *Assess*).

La diferencia entre ellos es que son promovidos por diferentes instituciones. CRISP-DM nace en Europa y promovida principalmente por la empresa SPSS. Por otro lado, SEMMA es considerado un estándar por la empresa SAS Institute Inc. [8]

Son similares ya que ambos estándares apoyan la idea de dividir el proceso de minería en fases esenciales que van desde seleccionar los datos hasta evaluar los resultados del modelo; a continuación se describirán brevemente las fases de cada uno de ellos:

CRISP-DM

- **Comprensión del problema:** Se enfoca en entender los requerimientos desde la perspectiva del negocio y un establecer un plan inicial para alcanzar objetivos.

[7] *International Journal of Computer Graphics & Animation (IJCGA), Multimedia Mining Research - An Overview, Vol.5, No.1*. Enero 2015.
[8] Rohanizadeh, S. S. and Moghadam, M. B. A Proposed Data Mining Methodology and its Application to Industrial Procedures Journal of Industrial Engineering (2009) pp 37-50.

- **Comprensión de los datos:** Contempla una recolección inicial de datos y la familiarización con ellos para comenzar a identificar aspectos importantes que permitan plantear hipótesis.
- **Preparación de los datos:** Esta fase cubre todas las tareas de procesamiento para llevar el conjunto de datos inicial al conjunto que servirá como insumo para el modelo de minería de datos.
- **Modelado:** Las técnicas de minería de datos aquí son seleccionadas y aplicadas. Usualmente se prueba más de una para elegir la más factible. También, algunas veces es necesario volver al paso de preparación de datos.
- **Evaluación:** En esta etapa se valida que el modelo esté siguiendo los intereses del negocio o efectivamente esté atacando el problema por resolver. Se aprovecha para verificar que ningún aspecto fundamental esté siendo ignorado. Al final de esta etapa se decide sobre que se hará con los resultados del modelo.
- **Despliegue:** Una vez que se ha evaluado la utilidad de los resultados se deben organizar y presentar de manera comprensible al "cliente", ya que en muchas ocasiones es el "cliente" es quien llevará acabo las acciones que sugieren los resultados del modelo.[9]

SEMMA

- ***Sample:*** Refiere a la selección de los datos potencialmente útiles. Se necesita recolectar suficientes datos para obtener una muestra óptima.
- ***Explore:*** Esta fase cubre el descubrimiento de las relaciones entre las variables, anormalidades en los datos y comprensión de los mismos. Se apoya de la visualización de datos.
- ***Modify:*** Esta etapa contiene los métodos de selección, creación y transformación de las variables destinadas al modelo de minería.
- ***Assess:*** La última fase consiste validar lo que muestran los resultados del modelo; confiabilidad y utilidad de los mismos.

[9] Óscar Marbán, Gonzalo Javier Mariscal y Segovia (2009). Universidad Europea de Madrid. *A Data Mining & Knowledge Discovery Process Model*

Los modelos descritos sirven, como ya se mencionó, de guía para dar secuencia a los proyectos de Minería de Datos.

1.3 Software de Minería de Datos

En nuestros días existe una extensa gama de herramientas desarrolladas para el descubrimiento de información, de la cual se puede distinguir entre el software libre (*open source*). Entre el software más conocido están WEKA (*Waikato Environment for Knowledge Analysis*), KNIME (*Knostanz Information Miner*), Orange, Tanagra, RapidMiner y R (*The R proyect for Statistical Computing*).[10]

Por otra parte, el software con propietario (*close source*). De este último, entre los más conocidos y aplicados, se tienen a SPSS/IBM, SAS Enterprise Miner/SAS, Microstrategy, Oracle DataMining y Teradata.[11]

Las plataformas especializadas en Minería de Datos usualmente incluyen técnicas propias para pre-procesamiento de datos (selección, limpieza, integración y transformación) además de los métodos predictivos y descriptivos que pertenecen a la Minería de Datos incluyendo la visualización de la información.

Como alternativa estable, eficiente y de fácil acceso esta tesis se enfocará al software libre disponible en internet que permite llevar a cabo proyectos que involucran el proceso completo de KDP.

1.3.1 Comparación de software libre

Naturalmente existen diferencias entre las herramientas para Minería de Datos. Se pueden distinguir aspectos como son: el ambiente de la plataforma, lenguaje del código de las rutinas, soporte en bases de datos, compatibilidad, la capacidad estadística, visualización y precisión de las rutinas.

En 2008 los investigadores asociados al desarrollo de la herramienta nombrada *Knowledge Extraction based on Evolutionary Learning* (KEEL) situaron entre las

[10] *International Journal of Advanced Trends in Computer Science and Engineering* (*IJATCSE). A Comparative Study on Data Mining Tools. Volume 4, No. 2,* Marzo – Abril 2015.

[11] *KD Nuggets,* Junio 2015 [En línea] Disponible: http://www.kdnuggets.com/software/suites.html

mejores plataformas *open source* a WEKA[12], KNIME[13], Orange[14], Tangara[15] y RapidMiner[16]. Definiendo criterios claves de funcionalidad con la finalidad de identificar las debilidades y fortalezas que KEEL pretendía armonizar en su construcción. El artículo de la investigación está disponible en su página principal[17].

Para realizar la comparación se definieron marcas clasificatorias en función del nivel de soporte de las plataformas:

- No los soporta la plataforma (N)
- Soporte básico (B)
- Soporte intermedio (I)
- Soporte avanzado (A)

Si las características no se califican en niveles, entonces se indicará su existencia:

- Soporte existente (Y)
- Sin soporte (N)

Los criterios a evaluar fueron el lenguaje usado para desarrollar el software, la interfaz gráfica, Entrada/Salida o Input/Output, variedad de algoritmos de pre-procesamiento y variedad de algoritmos de aprendizaje:

- **Interfaz Gráfica**

Representación gráfica: Interfaz con representación gráfica del flujo del proceso a través de nodos interconectados que facilitan la interpretación.

[12] WEKA The University Waikato, Junio 2015 [En línea] Disponible: http://www.cs.waikato.ac.nz/ml/weka/

[13] *KNIME Open for Innovation,* Junio 2015 [En línea] Disponible: https://www.knime.org/

[14] Orange, Junio 2015 [En línea] Disponible: http://orange.biolab.si/

[15] Tanagra, Junio 2015 [En línea] Disponible: http://eric.univ-lyon2.fr/~ricco/tanagra/en/tanagra.html

[16] Rapid Miner, Junio 2015 [En línea] Disponible: https://rapidminer.com/

[17] *Knowledge Extraction based on Evolutionary Learning (KEEL).* Junio 2015 [En línea]. Disponible: http://www.keel.es/

Visualización de datos: Incluye herramientas de representación de los datos como gráficos, tablas y mecanismos similares.

Manipulación de datos: Contempla las rutinas que permiten hacer operaciones manuales básicas sobre los datos tales como eliminación y modificación de filas y columnas. En la Tabla 1.1 se muestran las diferencias en relación con dichos criterios.

Tabla 1.1 Comparativo de aspectos básicos entre software libre disponible

Software	Lenguaje	Representación gráfica	Visualización de datos	Manipulación de datos
WEKA	Java	Y	A	A
KNIME	Java	Y	A	A
Orange	C++	Y	A	A
Tanagra	C++	A	A	Y
RapidMiner	Java	N	A	A

- **Entrada/Salida**

Esta funcionalidad se enfoca a los diferentes formatos soportados tales como ARFF (formato estándar de Weka). La columna "otros formatos" incluye XLS, CSV, XML. Mientras que en "conexión a bases de datos" se indica si la plataforma soporta esta funcionalidad, si puede cargar o guardar datos en dichos formatos, o bien, que pueda transformarlos a otro formato estándar que el software use. Ver Tabla 1.2.

Tabla 1.2 Comparativo de aspectos básicos entre software libre disponible

Software	Formato ARFF	Otros Formatos	Conexión a bases de datos
WEKA	Y	Y	Y
KNIME	Y	Y	Y
Orange	N	Y	N
Tanagra	Y	N	B
RapidMiner	Y	Y	Y

- **Algoritmos de Pre-procesamiento**

La Tabla 1.3 contempla la variedad de rutinas de pre-procesamiento tales como discretización, selección de características, selección de instancias y trato de valores vacíos.

Tabla 1.3 Comparativo de aspectos básicos entre software libre disponible

Software	Discretización	Selección de Características	Selección de instancias	Trato de valores vacíos
WEKA	I	A	B	B
KNIME	I	A	B	B
Orange	A	I	B	B
Tanagra	A	B	N	A
RapidMiner	I	A	B	B

- **Algoritmos de aprendizaje**

Aquí se compara la variedad de los métodos asociados la Minería de Datos (Clasificación, Regresión, y Agrupamiento, Reglas de Asociación). Ver Tabla 1.4.

Tabla 1.4 Comparativo de aspectos básicos entre software libre disponible

Software	Clasificación	Regresión	Agrupamiento	Reglas de asociación
WEKA	A	A	A	A
KNIME	A	A	A	A
Orange	I	N	I	I
Tanagra	I	A	A	Y
RapidMiner	A	A	A	A

Se puede apreciar que bajo los criterios de este análisis KNIME, WEKA y RapidMiner tienen ventaja en la variedad de algoritmos de aprendizaje y capacidad de compartir información en diferentes formatos. Por otra parte, Orange y Tanagra tienen más variedad en algoritmos de discretización.

1.3.2 Precisión y ejecución

Otro estudio realizado en Jordania en la universidad de Yarmouk se dio a conocer en el mismo año en la conferencia árabe de tecnología de la información (*ACIT Arab Conference on Information Technology*) a través del artículo titulado *A Comparison Study between Data Mining Tools over some Classification Methods*[18].

[18] *SAI, The Science and Information Organization,* Julio 2015 [En línea] Disponible: http://thesai.org/Publications/ViewPaper?Volume=1&Issue=3&Code=SpecialIssue&SerialNo=4

Este estudio consistió en medir el nivel de precisión y capacidad de ejecución de algunos métodos de clasificación en cada una de las herramientas. Se utilizaron dos algoritmos de evaluación; k-fold CV y Holdout, ambos coincidían en partir los datos en bloques; en una porción de datos se entrenaban las técnicas de clasificación y en la otra se evaluaba su desempeño.

Los algoritmos sujetos a la evaluación considerados para el estudio fueron *Naive Bayes (NB), K Nearest Neighbor (KNN), Support Vector Machine (SVM), Decision Tree Classifier (C4.5), One Rule (OneR), Zero Rule (ZeroR).*

Nueve Conjuntos de Datos con diferentes tipos de atributos, tamaño y dimensionalidad fueron seleccionados para poner a prueba la capacidad de ejecución de cada herramienta; fueron tomados de UCI Machine Learning Repository[19]. Ver

Tabla 1.5 Conjuntos de Datos de UCI Machine Learning Repository

#	Conjunto de Datos	Tipo de Atributo	# Instancias	# Atributos
1	Audiology	Categórico	226	69
2	Breast Cancer Wisconsin	Entero	699	10
3	Car Evaluation	Categórico	1,728	6
4	Flags	Categórico, Entero	194	30
5	Letter Recognition	Entero	20,000	16
6	Nursery	Categórico	12,960	8
7	Soybean	Categórico	638	36
8	Spambase	Entero, Real	4,601	57
9	Zoo	Categórico, Entero	101	17

[19]*UCI Machine Learning Repository,* Junio 2015 [En línea] Disponible: http://archive.ics.uci.edu/ml/

Tabla 1.5

Algunas plataformas tuvieron problemas para explotar Conjuntos de Datos con atributos multi-clase y con valores discretos (Tanagra y KNIME). Por otro lado el método OneR no es soportado por Orange, Tanagra y KNIME. Así mismo, ZeroR no tiene implementación en Tanagra y KNIME, a diferencia de WEKA que soporta los seis métodos compilando con éxito en los nueve Conjuntos de Datos.

Se generaron marcas para determinar el éxito y fracaso del método en relación con el Conjunto de Datos en cuestión:

- OK: Si el algortimo ejecutó con éxito
- NA: Si el algoritmo no está implementado
- D: Valor Discreto (problemas de ejecución)
- MC: Multi-clase (problemas de ejecución)

Los resultados del éxito o fracaso con base en dichas marcas se pueden ver en la Tabla 1.6.

Tabla 1.6 Capacidad de ejecución de KNIME y Tanagra usando k-fold CV

	Audiology	**Breast-W**	**Car**	**Flags**	**Letters**	**Nursery**	**Soybean**	**SpamBase**	**Zoo**
NB	OK	OK	OK	OK	OK	OK	OK	OK	OK
OneR	NA	NA	NA	NA	NA	NA	NA	NA	NA
C4.5	OK	OK	OK	OK	OK	OK	OK	OK	OK
SVM	MC/D	OK	MC/D	MC	MC	MC/D	MC/D	OK	MC
KNN	D	OK	D	OK	OK	D	D	OK	OK
ZeroR	NA	NA	NA	NA	NA	NA	NA	NA	NA

En cuanto al nivel de precisión, resultó variado en función del Conjunto de Datos que se tomó como insumo como se muestra en el Cuadro 1.7.

Los números resaltados en el Cuadro 1.7 señalan los métodos que presentaron dificultades.

Tabla 1.7 Precisión usando k-fold CV

	WEKA	KNIME	Orange	Tanagra
NB	56%-96%	52%-95%	58%-97%	63%-93%
OneR	17%-93%	NA	NA	NA
C4.5	59%-97%	55%-97%	54%-96%	57%-96%
SVM	61%-97%	67%-96%	64%-98%	90%-97%
KNN	57%-98%	33%-98%	58%-96%	25%-97%
ZeroR	4%-70%	NA	4%-70%	NA

KNIME y Tanagra tuvieron problemas para ejecutar los métodos SVM y KNN debido a los atributos multi-clase y discretos, sin embargo, tuvieron rangos de precisión considerablemente elevados de aquellos Conjuntos de Datos que si se lograron procesar.

Se puede observar que la plataforma con mejor desempeño es WEKA en capacidad de ejecución. Orange y Tanagra destacaron en su precisión al correr el algoritmo *Naive Bayes (NB).* Por otro lado KNIME cuenta con precisión similar a la de WEKA al ejecutar algoritmos como *Naive Bayes (NB)* y *Decision Tree Classifier (C4.5)*.

Este trabajo se centrará en la plataforma KNIME persiguiendo la idea de que es un ambiente altamente comprensible y organizado debido al concepto workflow en el

que está basado. Por otro lado, la plataforma KNIME sobresale por ser utilizada activamente por alrededor de 6,000 profesionales en diferentes partes del mundo y adoptar estándares (como *PMML Predictive Modeling Markup Language*) que le ofrecen solidez y competitividad[20]. También, la consultora de tecnologías de la información *Gartner* la ubica en el cuadrante líder de plataformas de analítica avanzada junto con SAS, IBM y RapidMiner[21].

[20] *IJCSA, Comparative Analysis of Data Mining Tools and Techniques for Evaluating Performance of Database System, Vol. 6, No.2, Apr 2013*

[21] *KD Nuggets,* Junio 2015 [En línea] Disponible: http://www.kdnuggets.com/2015/02/gartner-2015-magic-quadrant-advanced-analytics-platforms.html

2. Plataforma KNIME

2.1 Definición y características

KNIME Analytics Platform es un software de código abierto para el análisis exploratorio de datos que se promueve con carácter innovador, abierto a nuevas contribuciones, así como flexible e intuitivo. El proyecto nació en la Universidad de Konstanz, Alemania, alrededor de 2004 y cuatro años más tarde se convirtió en la empresa *KNIME.com GmbH* con sede en Zúrich, Suiza, fundada por los desarrolladores[22] de la plataforma con el objetivo de dar servicios de consultoría.

KNIME es una suite compuesta por los siguientes pilares:

- Productividad
  - *KNIME Personal Productivity*
  - *KNIME Partner Productivity*
- Colaboración
  - *KNIME TeamSpace*
  - *KNIME Server Lite*
  - *KNIME Server*
- Ejecución
  - *KNIME Cluster Execution*
  - *KNIME Big Data Extensions*
- Contribuciones adicionales
  - *KNIME Trusted Community*
  - *Community Contributions*

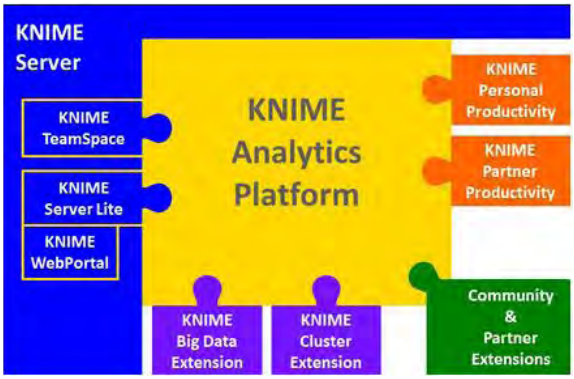


Fig. 2.1 Plataforma KNIME y sus extensiones comerciales

El profesor Michael Berthold, uno de creadores originales y miembro de la empresa explicó que KNIME debería estar disponible para todo aquel que estuviera interesado en análisis exploratorio de datos:

[22] *KNIME Team,* Julio 2015 [En línea] Disponible: https://www.knime.org/team

> *"It is important to stress that there will be no functional difference between the freely available KNIME version and the supported one – there are no plans to restrict the open source version of KNIME in any way, in contrast, we believe KNIME should be freely available to anyone interested in analytical data exploration"*[23]

Es decir, KNIME se encuentra bajo la licencia *GNU General Public License* lo cual hace su descarga gratuita.

KNIME está desarrollado sobre la plataforma Eclipse cuya comunidad está orientada a colaborar con *open source software* (como lo es KNIME) a través de proyectos que contribuyan a la construcción, desarrollo y administración del software a lo largo de su ciclo de vida[24].

Fig. 2.2 Plataforma sobre la cual KNIME está desarrollada

El lenguaje que predomina en el código fuente de KINME es Java, sin embargo, también tiene presencia de XML y SQL.

Además del ambiente para ejecución *KNIME Analytics Plataform*, existe otro disponible para el desarrollo de nuevas rutinas, es decir, su SDK (*Software Developer Kit*). La descarga de ambos ambientes se puede realizar desde su página web. Ver figura 2.3.[25]

[23] *KNIME Open for Innovation,* Julio 2015 [En línea] Disponible: https://www.knime.org/blog/commercial-support-launched-open-source-knime

[24] *ECLIPSE,* Julio 2015 [En línea] Disponible: http://www.eclipse.org/home/index.php

[25] *KNIME,* Descarga, Julio 2015 [En línea] Disponible: https://www.knime.org/downloads/overview

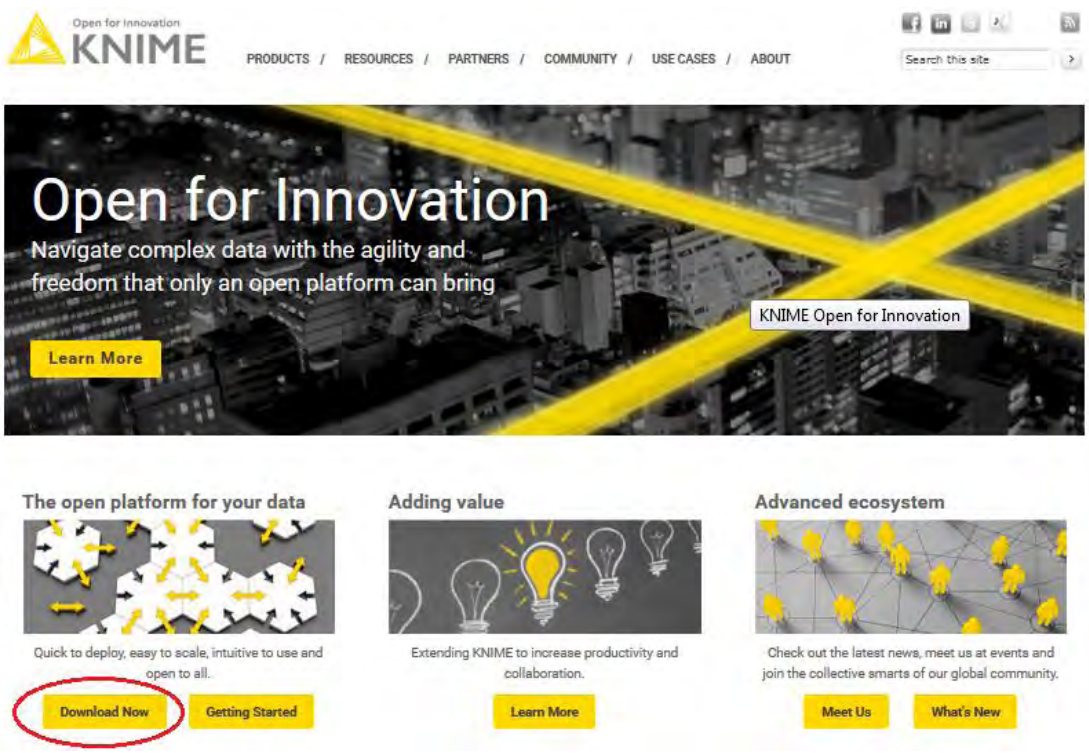


Fig. 2.3 Pagina de descarga KNIME

2.2 Descripción del ambiente

Como se mencionó, una de las características de KNIME es la interfaz intuitiva la cual se rige no por *scripts,* si no por nodos, o unidades de procesamiento, interconectados que generan una secuencia de análisis de datos o *workflow* (Direct Acyclic Graph - DAG")[26].

2.2.1 Workbench

Al ambiente de trabajo de la plataforma se le llama *WorkBench* y se compone de diferentes paneles tales como Menú General, Barra de Tareas, Explorador KNIME, Editor de *Workflow*, Repositorio de Nodos, Descripción del Nodo, Consola, *Outline* y Nodos Favoritos.

[26]KNIME: *The Konstanz Information Miner.* Julio 2015 [En línea] Disponible: http://www.kdd2006.com/docs/KDD06_Demo_13_Knime.pdf

El repositorio de nodos situado parte inferior izquierda del *WorkBench,* alberga los nodos clasificados en relación con modelo codificado según del tipo de solución o contribución al análisis de los datos que represente. Ver Fig 2.4.

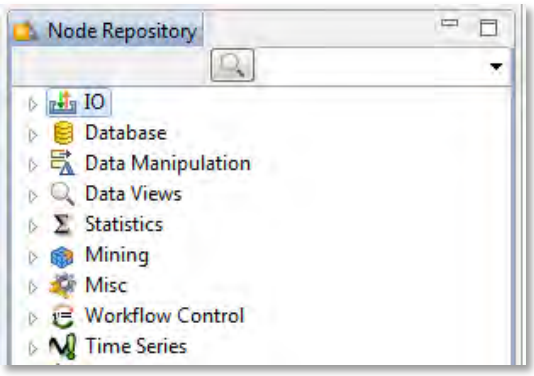


Fig. 2.4 Repositorio de nodos

Cada nodo cuenta con atributos que le permiten integrarse a un *workflow* como unidad de procesamiento del mismo como se puede ver en la Fig. 2.5.

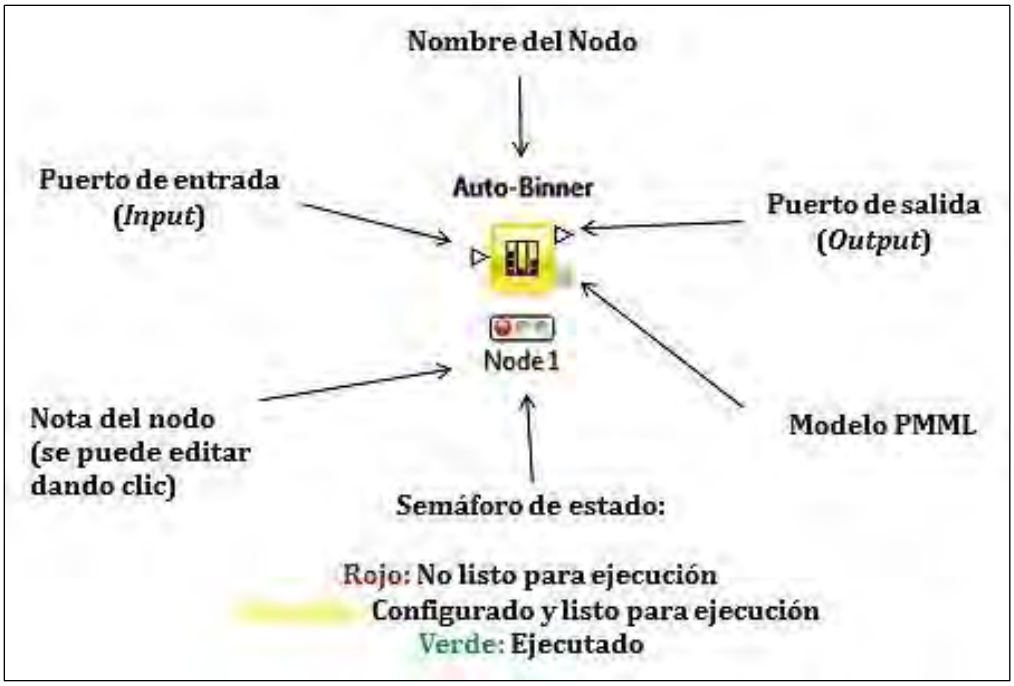


Fig. 2.5 Partes del Nodo KNIME

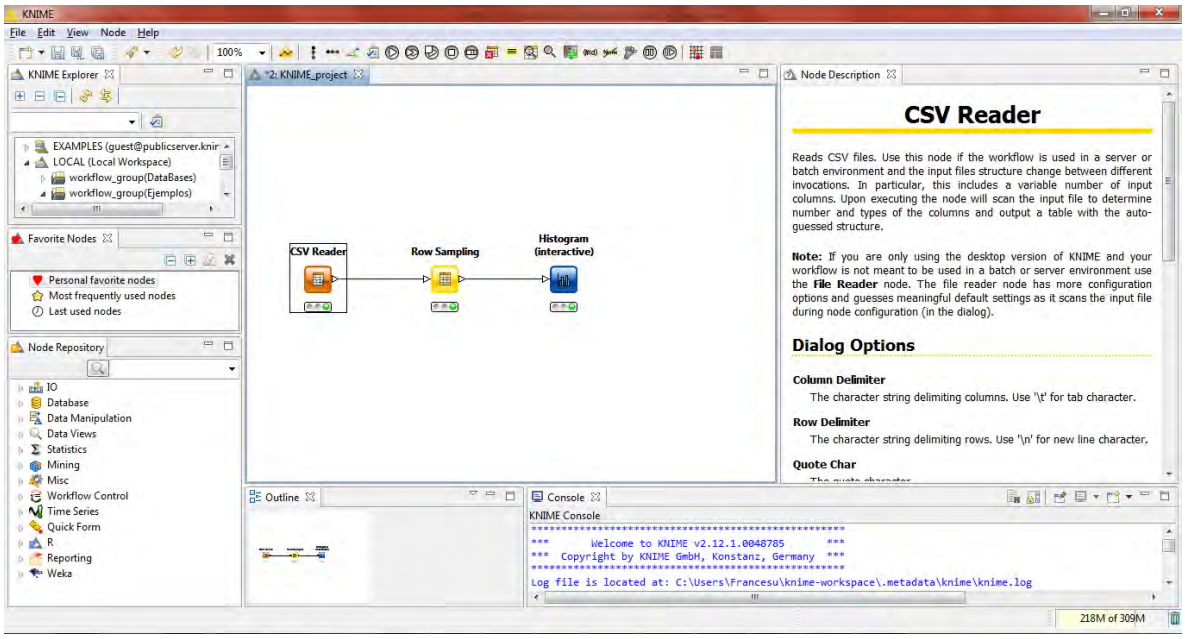


Fig. 2.6 Workflow KNIME_project

Los nodos son arrastrados y depositados en el *Workflow Editor* (parte central del *WorkBench*) para que puedan ser configurados, interconectados y ejecutados como se muestra en la Fig. 2.6.

En la parte superior derecha del *WorkBench* existe un panel llamado *Node Description* con la descripción del modelo que alberga el nodo, opciones de configuración y tipos de parámetros (*input port – output port*). Cada vez que se selecciona el nodo se hace visible la descripción.

El panel *Console* muestra los errores y alertas de la ejecución. KNIME guarda el historial con más detalle en un archivo .log en la carpeta predeterminada llamada *Workspace*.

KNIME Analytics Plataform en su descarga incluye un manual básico (*quickstart*) para la familiarización con el *WorkBench*, mismo puede ser revisado en su sitio web[27].

[27] KNIME, *Workbench*. Julio 2015 [En línea] Disponible: https://tech.knime.org/workbench

2.2.2 Workspace

El *Workspace* es una carpeta que se crea de forma automática en el disco duro de la PC durante la instalación de la plataforma. Misma donde se guardarán los proyectos de las sesiones KNIME.

También es posible cambiar el *Workspace* si se requiere. Esto se hace dentro la sesión KNIME, yendo a *TopMenu->File->Switch Workspace* y después seleccionar la carpeta predefinida por el usuario en la que se desean guardar los proyectos. Ver Fig. 2.7.

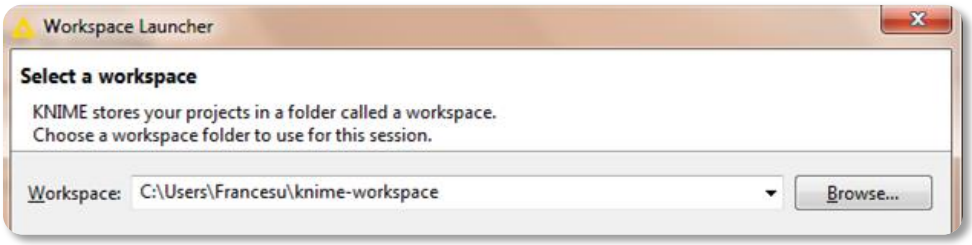


Fig. 2.7 Definición del Workspace

Importar/Exportar *Workflows*

En la Fig. 2.8 se muestra el cuadro de diálogo que lee y copia *workflows* en el *Workspace* (LOCAL) en el que se está trabajando*:*

TopMenu->File-> Import KNIME workflow

Muestra dos opciones que determinan la forma de lectura; referenciado a una carpeta o a un archivo comprimido.

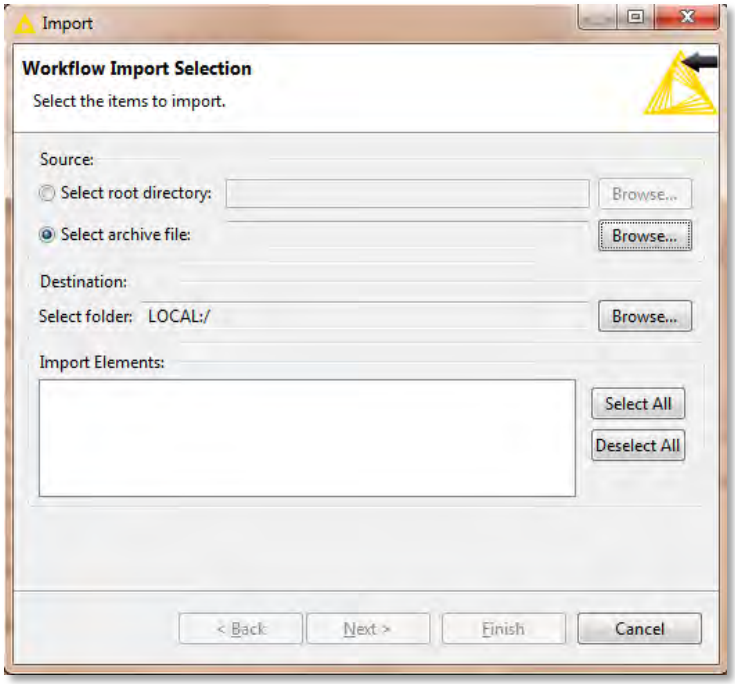


Fig. 2.8 Diálogo Import Workflow

Nota: Los archivos comprimidos son creados por la operación inversa *Import.*

El cuadro de diálogo de la Fig. 2.9 se presenta el que sirve para exportar *workflows* está en el mismo menú:

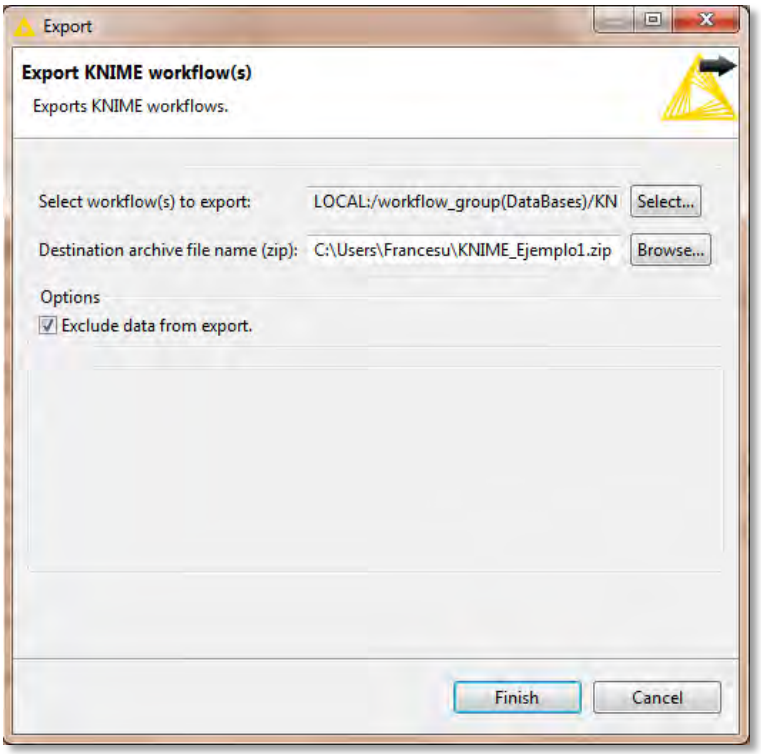


Fig. 2.9 Diálogo Export Workflow

TopMenu->File-> Export KNIME workflow y recibe tres parámetros:

- *Workflow* que se desea exportar del Workspace
- Ruta destino del archivo comprimido
- La instrucción de incluir solo los nodos. Lo que reduce considerablemente el tamaño del archivo.

2.2.3 Configuración KNIME

En el siguiente cuadro (Fig. 2.10) de dialogo se pueden personalizar todas las configuraciones de la plataforma: *TopMenu->File->Preferences*

Una revisión general de las opciones:

- *Databases* especifica los controladores de base de datos más comunes y/o recientes.
- *KNIME Explorer* permite acceder a repositorios compartidos vía *TeamSpace* y *KNIME Server* (extensiones comerciales).
- *KNIME GUI* contiene algunas configuraciones del *workbench* y *layout*.

- *Master key* es usado para cifrar las contraseñas que se ingresan en nodos que conectan con bases de datos, sin embargo, desde la versión KNIME 2.2 será reemplazada por la opción *Workflow Credentials.*
- *Meta info Preferences* desde donde se pueden subir plantillas de *nodos* y *workflows.*
- Después de la opción *Meta info Preferences* la configuración de los paquetes-extensión que han sido instalados.

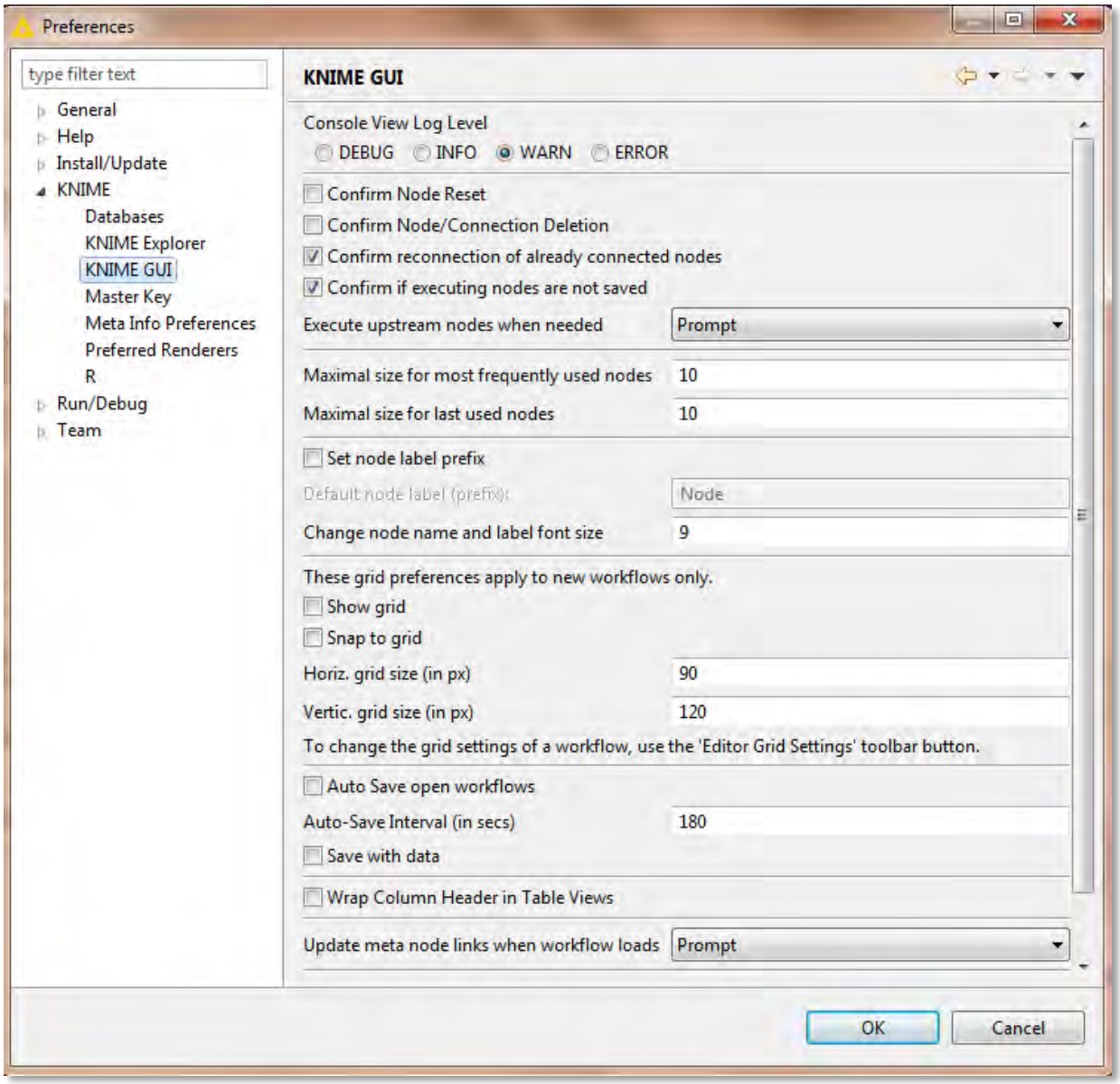


Fig. 2.10 Diálogo Configura Plataforma

2.3 Extensiones y compatibilidad KNIME

Existen extensiones de la funcionalidad de la plataforma, las cuales deben ser instaladas por separado, a través de un cuadro de diálogo llamado *Install* (Fig. 2.11). Para llegar a él hay dos maneras:

1. *TopMenu->File->Install KNIME Extensions*, clic en siguiente y seguir las instrucciones del asistente.
2. *TopMenu->Help->Install New Software* después ingresar la URL del llamado KNIME Update Site (http://update.knime.org/analytics-platform/2.x). Después seleccionar las extensiones, dar clic en siguiente y seguir las instrucciones del asistente.

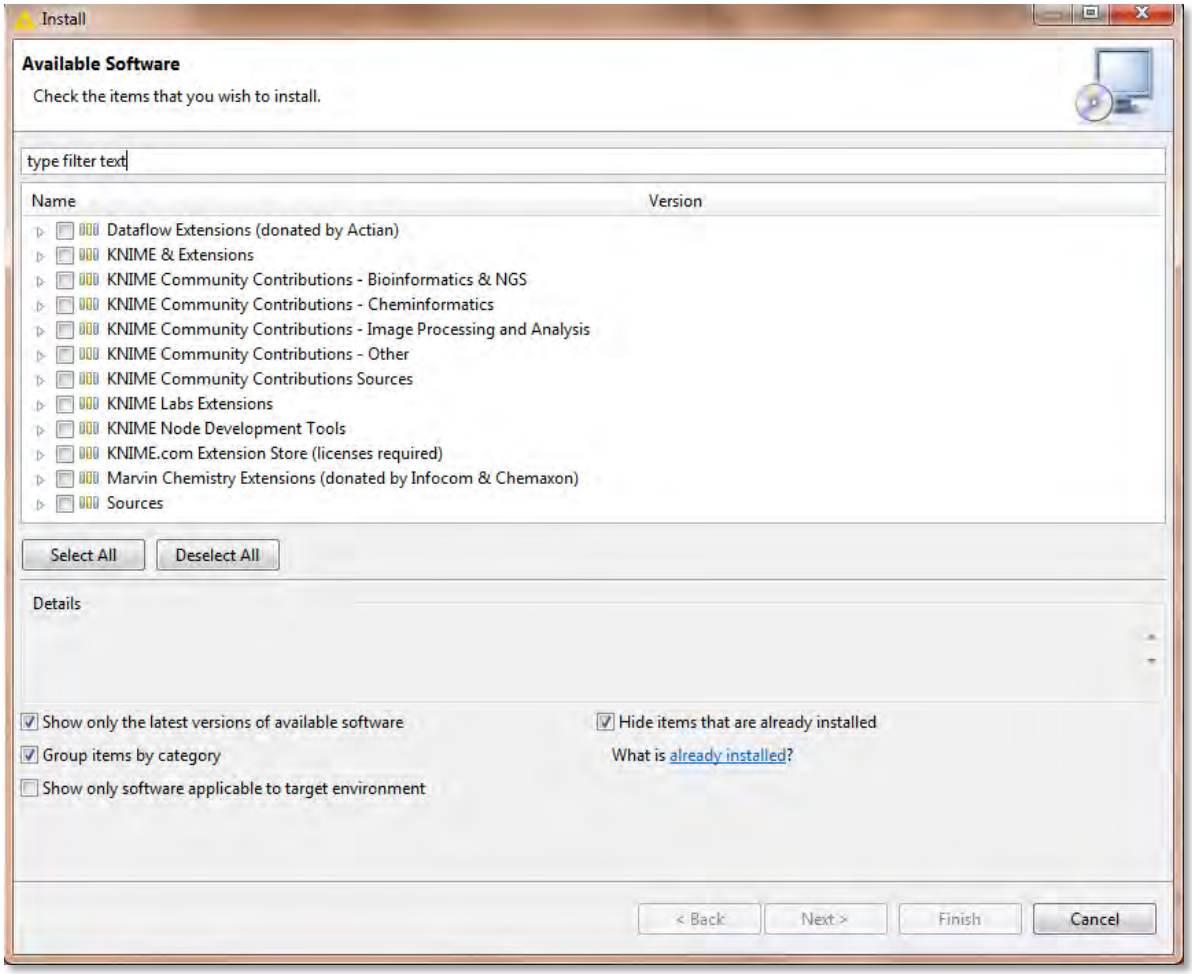


Fig. 2.11 Diálogo Install para Extensiones KNIME

Una vez instaladas deberán aparecer en el repositorio de nodos del *workbench*.

Las extensiones se agrupan principalmente en los siguientes rubros:

- **KNIME & Extensions:** Contiene todas las extensiones revisadas por KNIME para su liberación.
- **KNIME & Lab Extensions:** Contiene paquetes desarrollados por KINME listos para usarse, pero no se garantiza aún la calidad de liberación.
- **KNIME Developer tools:** Contiene algunas herramientas útiles para el desarrollo de nodos.
- **Source:** Contiene el código fuente de KNIME y algunos paquetes donados por terceros y otras comunidades**.**

2.4 Ventajas y Desventajas de KINME

Entre las principales ventajas de ésta plataforma se identificaron las siguientes:

- **La inclusión de *plugins* para proveer de funcionalidad adicional a la plataforma**

Eclipse tiene una rutina instalada capaz de ejecutar *plugins*. Dichos *plugins* requieren cierta estructura para que puedan ser ejecutados y son guardados en un archivo XML (**plugin.xml**) donde se definen las dependencias con otros *plugins* y "**puntos de extensión**" a donde el *plugin* pretende conectarse. Así mismo, otros archivos de la forma **build.properties** que contienen información de exportación y almacenamiento de la clase asociada al *plugin* constituyen la configuración completa del *plugin*.

Las rutinas más conocidas que aportan funcionalidad adicional, cuyos *plugins* están configurados para ser instalados e incluidos a KNIME son los desarrollados por WEKA y R. Una vez instaladas, dichas ruinas se ven integradas en el repositorio de nodos como se observa en la Fig. 2.12.

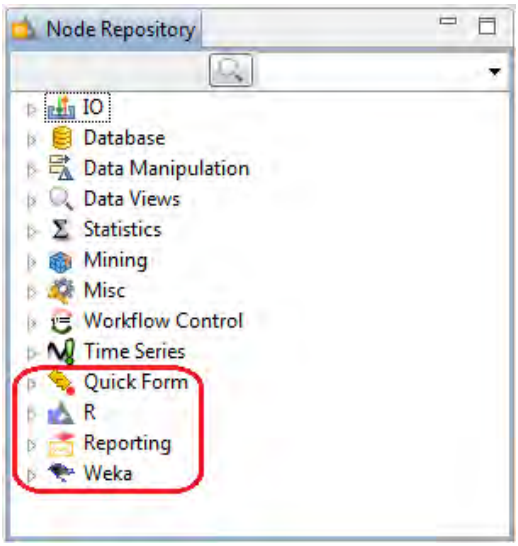


Fig. 2.12 Extensiones en Repositorio de Nodos

- **Capacidad para desarrollar nuevas rutinas**

Además de su versión ejecutable con los módulos pre-existentes, KNIME cuenta con una herramienta descargable llamada **SDK (*Software Developer Kit*)**, a través de la cual es posible desarrollar código Java de nuevas rutinas y realizar la configuración del *plugin* que hará posible la ejecución del mismo. Ver Fig. 2.13.

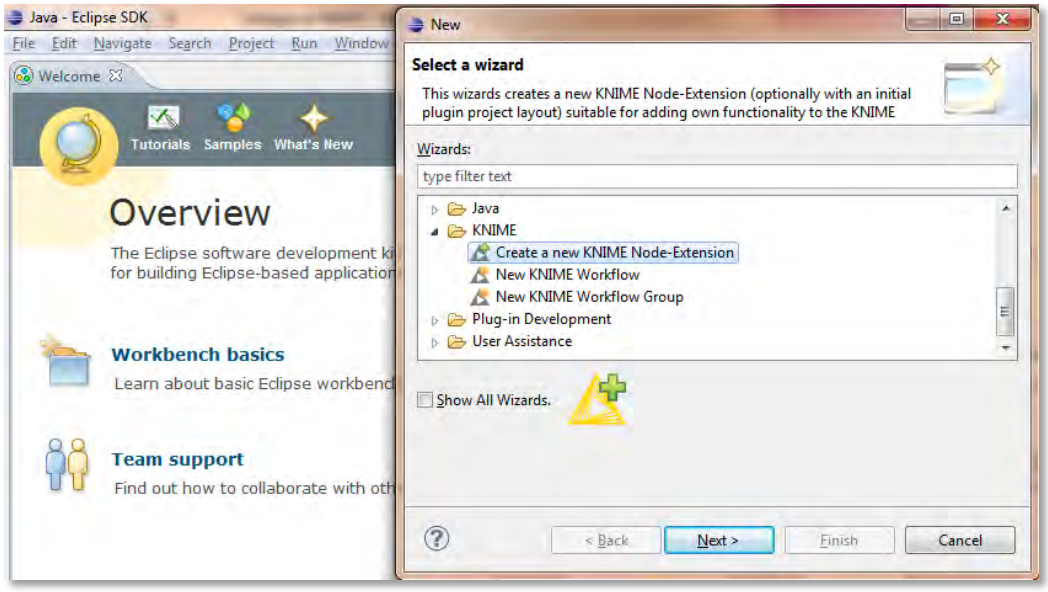


Fig. 2.13 Herramienta SDK

KNIME aprovecha la *arquitectura-plugin* de Eclipse para el desarrollo de nuevos nodos instalando dos "**puntos de extensión"**:

1. "Categories" **org.knime.workbench.repository.categories**
2. "Nodes" **org.knime.workbench.repository.nodes**

La infraestructura del *plugin* es realizada por un asistente para que el usuario invierta poco tiempo en su configuración y pueda centrarse en el desarrollo del nuevo nodo o rutina.

KNIME cuenta con una guía para desarrollo de nuevos nodos donde se muestra el proceso de instalación y algunos ejemplos[28].

- **Foros y comunidades usuarias de KNIME**

Se pueden disipar dudas y discutir diversos temas relacionados con el funcionamiento de KNIME, desde su sitio web donde existe un espacio llamado KNIME Community forum[29] donde practicantes y profesionales hacen sus aportaciones o encuentran soporte de la plataforma.

Para poder participar en los foros de discusión es necesario registrarse creando una cuenta KNIME vinculada a una cuenta de correo electrónico. Ver Fig. 2.14.

[28] KNIME, SDK Guía Rápida. Julio 2015 [En línea]. Disponible: https://tech.knime.org/developer-guide

[29] KNIME, Foro de Usuarios. Julio 2015 [En línea] Disponible: https://tech.knime.org/forum

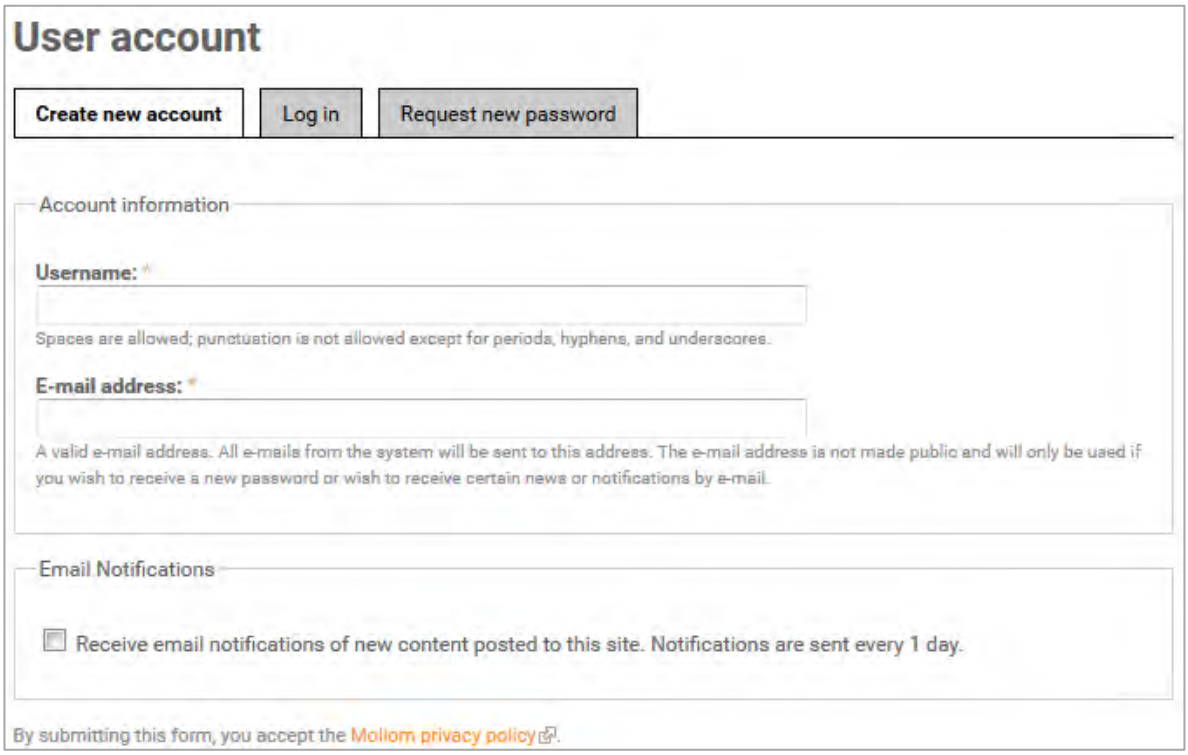


Fig. 2.14 Formulario de registro en KNIME

Como desventajas de KNIME se consideran las siguientes:

- Hay poca difusión de la herramienta en comparación con el software comercial.
- Puede resultar complejo comprender el funcionamiento de la herramienta debido a la variedad de herramientas con las que se integra.

2.5 Conexión a Bases de Datos

La API KNIME es capaz de leer datos provenientes de fuentes estructuradas como lo son las bases de datos. Gracias a un controlador JDBC (*Java Database Connectivity*) que permite la ejecución de operaciones sobre bases de datos desde el lenguaje de programación Java. Por lo que es necesaria la instalación del controlador JDBC que proporciona el Sistema Manejador de Bases de Datos (SMBD).

Existen otros parámetros esenciales para lograr la conexión como son; el nombre de la base de datos, servidor donde se aloja y el puerto de conexión entre el servidor y el sistema operativo, así como un certificado de acceso a la base (usuario y contraseña). Todos ellos son heredados de la instalación del SMBD que definió la estructura de la base de datos fuente. KNIME puede trabajar con cualquiera de los manejadores de bases MySQL, MS SQL, Oracle, PostgreSQL, entre otras.

Los nodos utilizados para llevar a cabo las tareas de conexión, pre-procesamiento y lectura de los datos extraídos de una base, se encuentran contenidos en el repositorio de nodos del ***Workbench*** en el grupo *Database*. Ver Figura 2.15.

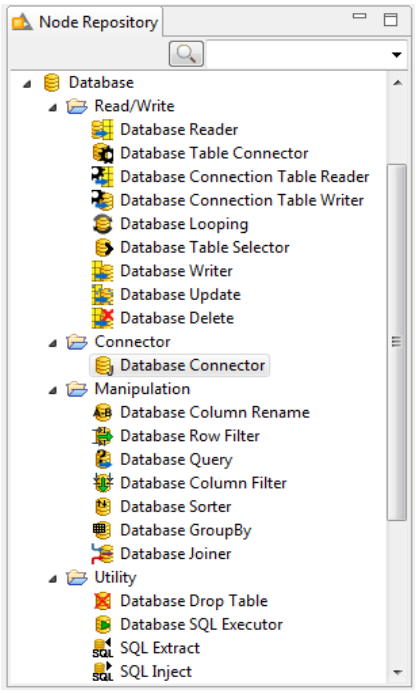


Fig. 2.15 Nodos Base de Datos

Los más representativos son los siguientes:

- **Database Table Connector:** Establece una conexión y genera la primera consulta orientada al propósito del usuario.

- **Database Connector:** Este nodo, exclusivamente establece la conexión a la base de datos en su configuración.

- **Database Reader:** Es posible generar la conexión, realizar una consulta sobre los datos y leer una tabla.

La diferencia entre los nodos de conexión radica en la forma de pre-procesamiento de los datos. El nodo *Database Reader* prepara los datos en una sola configuración, con la característica de que solo puede ejecutar una consulta y la imposibilidad de generar más de una tabla personalizada de la misma conexión. En contraste, los otros dos nodos; *Database Table Connector* y *Database Connector* se usan cuando se quieren llevar a cabo dichas tareas.

2.5.1 Ejemplo 1. *Workflow* Bases de Datos

- ✓ Se elige PostgreSQL para establecer la conexión aprovechando su interfaz gráfica interactiva.
- ✓ Se asume que el usuario tiene conocimiento básico en dicho SMBD para definir una base de datos.
- ✓ Se considera una base existente en PostgreSQL llamada *Dresses,* la cual está disponible en el repositorio público *Machine Learning Repository*[30].

Comenzamos una sesión KNIME y creamos un nuevo workflow llamado **KNIME Ejemplo1.** Enseguida seleccionamos y arrastramos un nodo conector hacia el *Workflow Editor* para que pueda ser configurado como se muestra en la Fig. 2.16.

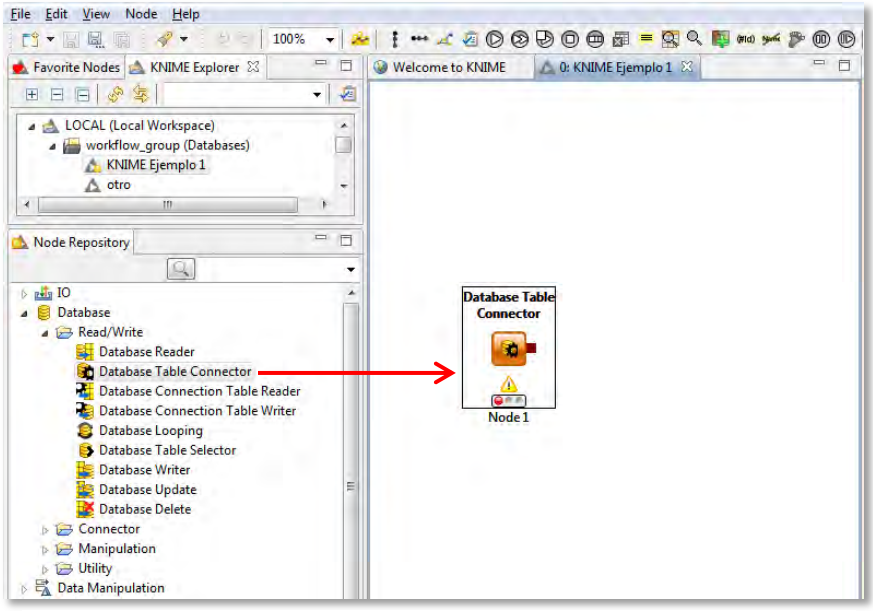


Fig. 2.16 Inicio de KNIME Ejemplo 1

[30] *Dresses_Atribute_Sales,* Repositorio Público UCI. Agosto 2015 [En línea] Disponible: http://archive.ics.uci.edu/ml/datasets/Dresses_Attribute_Sales

Antes de la configuración del nodo, necesitamos asegurarnos que el controlador JDBC instalado sea reconocido por KNIME, lo cual se puede validar en *MainMenu -> File -> Preferences -> KNIME -> Databases* desde donde puede ser cargado.

En este caso se está utilizando el JDBC para PostgreSQL el cual puede ser descargado de la página de PostgreSQL[31], para posteriormente cargarlo a la configuración de KNIME. Ver Fig. 2.17.

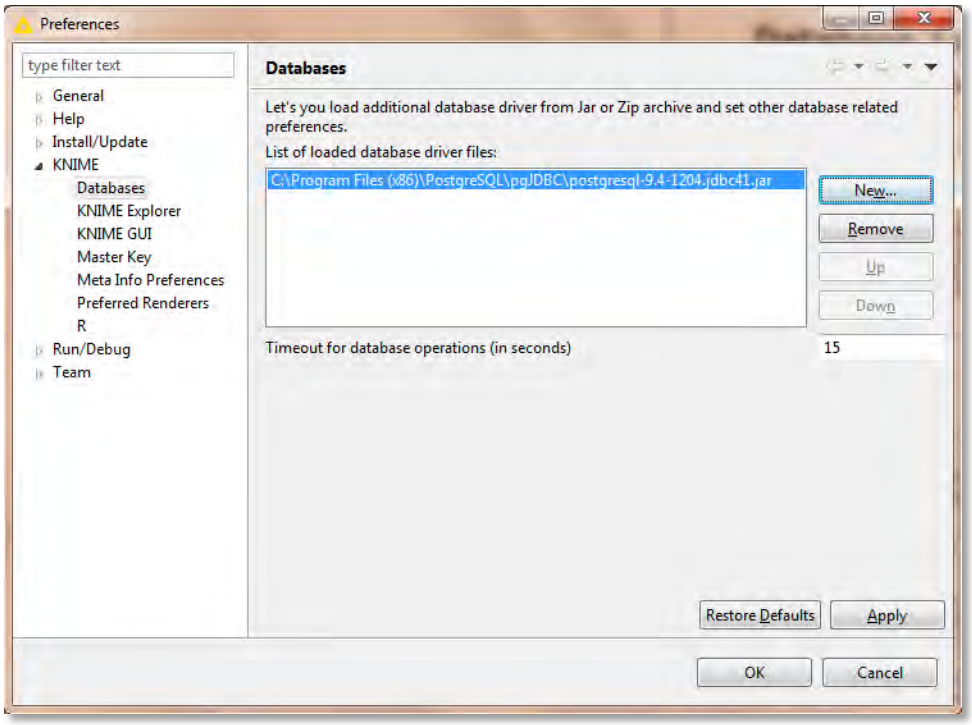


Fig. 2.17 Carga de Controlador JDBC

Fig. 2.18 Edición de Contraseña

Así mismo se debe declarar el certificado de acceso a la base de datos. Esto se logra dando clic derecho en el *workflow* **KNIME Ejemplo 1** y seleccionando la opción *Workflow Credentials*. Inmediatamente se despliega un

[31] PostgreSQL. Agosto 2015. [En línea]. Disponible: https://jdbc.postgresql.org/download.html

cuadro como el que se muestra en la Fig. 2.18.

Cabe mencionar que hay otra alternativa de identificación para acceder a la base de datos directamente en el menú de configuración del nodo, sin embargo, no cifra las contraseñas por lo que se considera obsoleta. Los desarrolladores de la plataforma sugieren usar la opción *Workflow Credentials.*

Reunidos los elementos descritos arriba ya es posible configurar el nodo que permitirá la conexión.

Ingresamos los parámetros a la configuración del nodo *Database Table Connector,* (opción *Configure* del menú contextual del nodo). Ver Fig. 2.19.

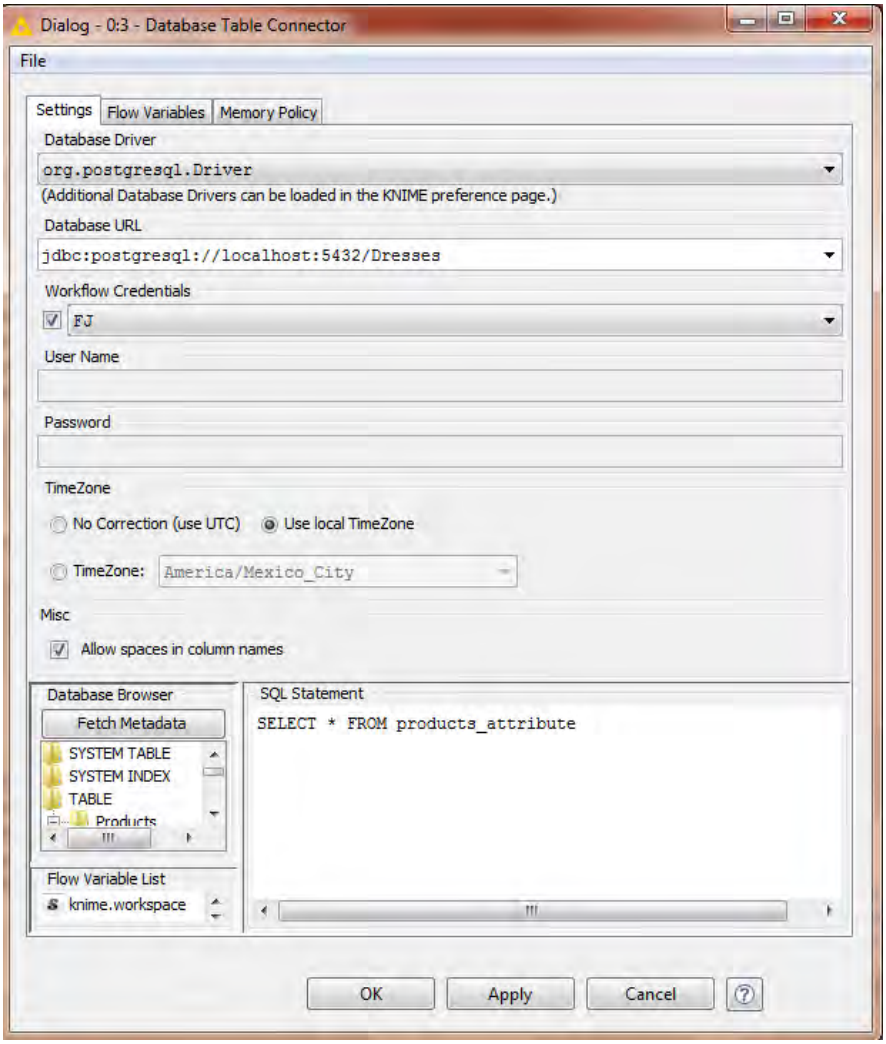


Fig. 2.19 Configuración de Nodo Database Table Connector

1. **Database Driver:** Se selecciona de la lista el controlador previamente cagado
2. **Database URL:** Se construye de la siguiente forma:
   jdbc:<nombre del SMBD>://<nombre del servidor>:<puerto de conexión>/<nombre de la base de datos>
3. **TimeZone:** Usa zona horaria local
4. **Workflow Credentials:** Se selecciona el certificado de acceso previamente cargado
5. **SQL Statement:** donde se ingresa la primera consulta de extracción. Elusuario se puede apoyar en el campo *Fetch Metadata* que permite visualizar las tablas existentes en la base de datos de donde se extraen los datos.

Una vez ingresados los parámetros del nodo se selecciona "ok" y automáticamente cambiará el semáforo a color amarillo, salvo que exista algún error descrito en el panel *Console*.

Luego ejecutamos desde el menú contextual del nodo como se muestra en la Fig. 2.20.

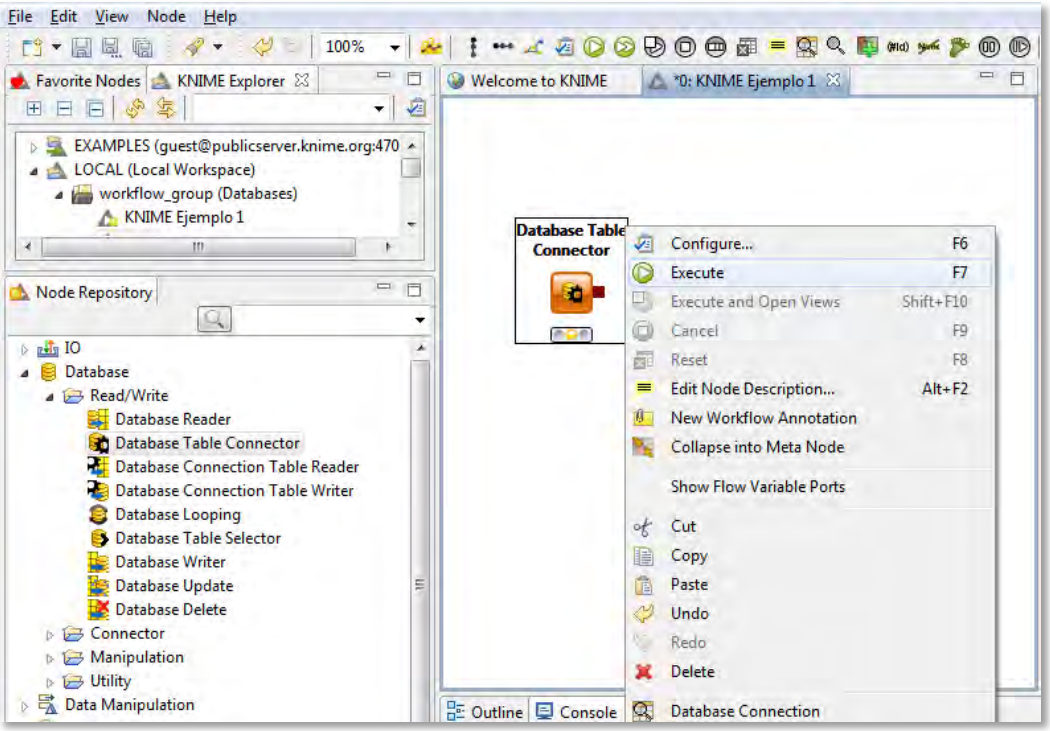
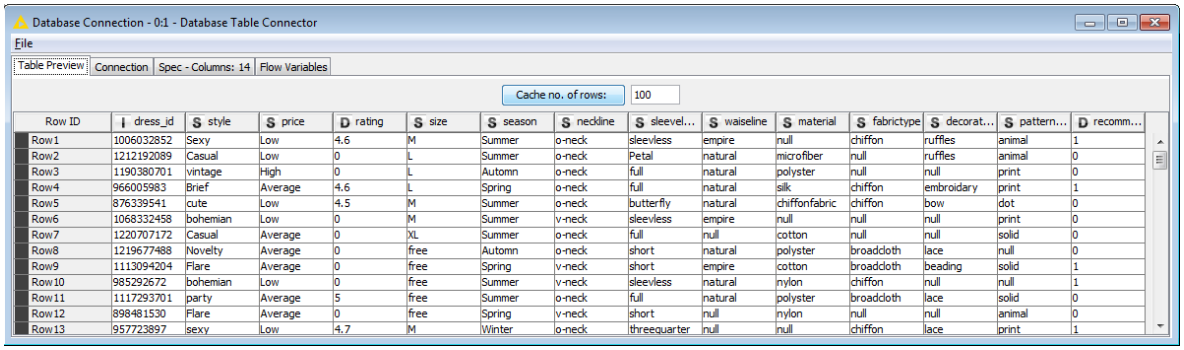


Fig. 2.20 Ejecución de Nodo Database Table Reader

La salida del nodo *Database Table Connector* es una conexión establecida entre el SMBD y la plataforma KNIME con la selección de la primera consulta. Y se puede observar dando clic en *Database Connection* que se encuentra al final del menú contextual del nodo. Ver Figura 2.21.



Row ID	dress_id	style	price	rating	size	season	neckline	sleevel...	waiseline	material	fabrictype	decorat...	pattern...	recomm...
Row1	1006032852	Sexy	Low	4.6	M	Summer	o-neck	sleevless	empire	null	chiffon	ruffles	animal	1
Row2	1212192089	Casual	Low	0	L	Summer	o-neck	Petal	natural	microfiber	null	ruffles	animal	0
Row3	1190380701	vintage	High	0	L	Automn	o-neck	full	natural	polyster	null	null	print	0
Row4	966005983	Brief	Average	4.6	L	Spring	o-neck	full	natural	silk	chiffon	embroidary	print	1
Row5	876339541	cute	Low	4.5	M	Summer	o-neck	butterfly	natural	chiffonfabric	chiffon	bow	dot	0
Row6	1068332458	bohemian	Low	0	M	Summer	v-neck	sleevless	empire	null	null	null	print	0
Row7	1220707172	Casual	Average	0	XL	Summer	o-neck	full	null	cotton	null	null	solid	0
Row8	1219677488	Novelty	Average	0	free	Automn	o-neck	short	natural	polyster	broadcloth	lace	null	0
Row9	1113094204	Flare	Average	0	free	Spring	v-neck	short	empire	cotton	broadcloth	beading	solid	1
Row10	985292672	bohemian	Low	0	free	Summer	v-neck	sleevless	natural	nylon	chiffon	null	null	1
Row11	1117293701	party	Average	5	free	Summer	o-neck	full	natural	polyster	broadcloth	lace	solid	0
Row12	898481530	Flare	Average	0	free	Spring	v-neck	short	null	nylon	null	null	animal	0
Row13	957723897	sexy	Low	4.7	M	Winter	o-neck	threequarter	null	null	chiffon	lace	print	1

Fig. 2.21 Salida de Nodo Database Table Connector

Ahora bien, a continuación se ilustra el uso de algunos nodos compatibles al nodo *Database Table Connector* los cuales apoyan la manipulación de los datos. Dichos nodos se diferencian por tener puertos de entrada y salida de color café como se muestra en la Fig. 2.22.

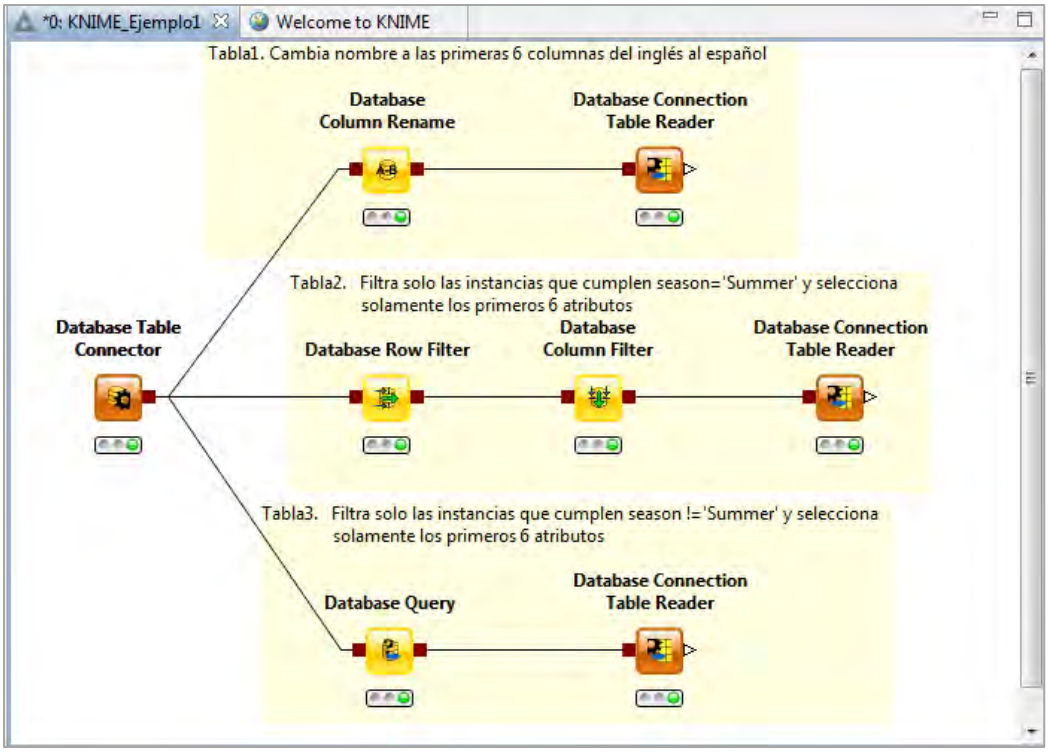
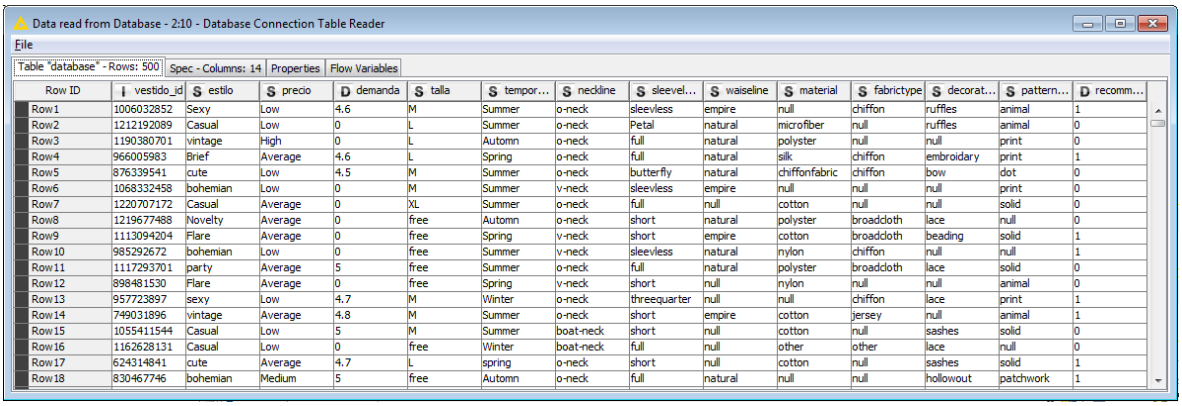


Fig. 2.22 Workflow KNIME Ejemplo 1

Tabla 1: Para generar esta tabla se usa el nodo *Database Column Rename* que permite cambiar nombre de las columnas que el usuario requiera. En este caso se cambia el nombre de las primeras seis columnas.

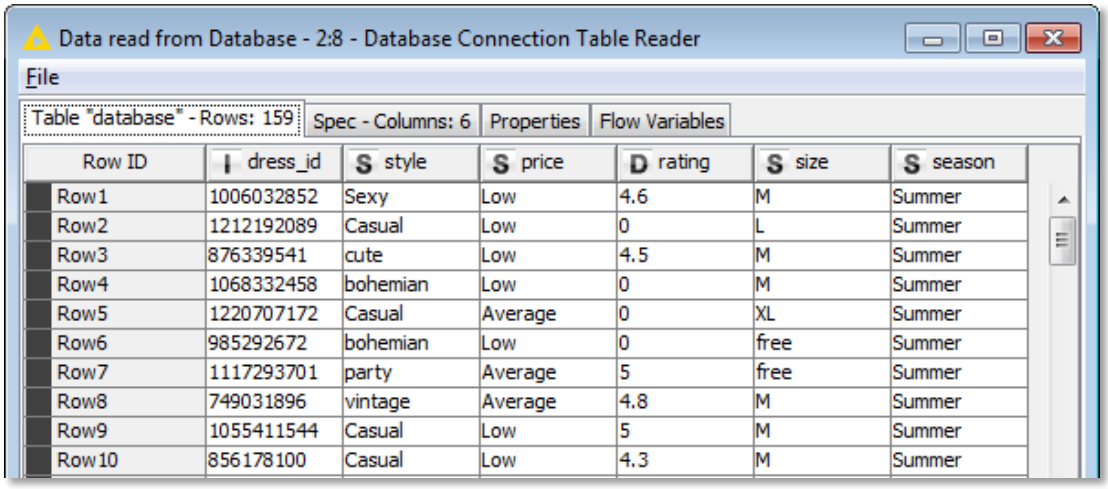
Después el nodo *Database Connection Table Reader* realiza la lectura de los datos seleccionados y modificados. Devuelve una tabla en KNIME como se muestra en la Fig. 2.23.



Row ID	vestido_id	estilo	precio	demanda	talla	tempor...	neckline	sleevel...	waiseline	material	fabrictype	decorat...	pattern...	recomm...
Row1	1006032852	Sexy	Low	4.6	M	Summer	o-neck	sleevless	empire	null	chiffon	ruffles	animal	1
Row2	1212192089	Casual	Low	0	L	Summer	o-neck	Petal	natural	microfiber	null	ruffles	animal	0
Row3	1190380701	vintage	High	0	L	Automn	o-neck	full	natural	polyster	null	null	print	0
Row4	966005983	Brief	Average	4.6	L	Spring	o-neck	full	natural	silk	chiffon	embroidary	print	1
Row5	876339541	cute	Low	4.5	M	Summer	o-neck	butterfly	natural	chiffonfabric	chiffon	bow	dot	0
Row6	1068332458	bohemian	Low	0	M	Summer	v-neck	sleevless	empire	null	null	null	print	0
Row7	1220707172	Casual	Average	0	XL	Summer	o-neck	full	null	cotton	null	null	solid	0
Row8	1219677488	Novelty	Average	0	free	Automn	o-neck	short	natural	polyster	broadcloth	lace	null	0
Row9	1113094204	Flare	Average	0	free	Spring	v-neck	short	empire	cotton	broadcloth	beading	solid	1
Row10	985292672	bohemian	Low	0	free	Summer	v-neck	sleevless	natural	nylon	chiffon	null	null	1
Row11	1117293701	party	Average	5	free	Summer	o-neck	full	natural	polyster	broadcloth	lace	solid	0
Row12	898481530	Flare	Average	0	free	Spring	v-neck	short	null	nylon	null	null	animal	0
Row13	957723897	sexy	Low	4.7	M	Winter	o-neck	threequarter	null	null	chiffon	lace	print	1
Row14	749031896	vintage	Average	4.8	M	Summer	o-neck	short	empire	cotton	jersey	null	animal	1
Row15	1055411544	Casual	Low	5	M	Summer	boat-neck	short	null	cotton	null	sashes	solid	0
Row16	1162628131	Casual	Low	0	free	Winter	boat-neck	full	null	other	other	lace	null	0
Row17	624314841	cute	Average	4.7	L	spring	o-neck	short	null	cotton	null	sashes	solid	1
Row18	830467746	bohemian	Medium	5	free	Automn	o-neck	full	natural	null	null	hollowout	patchwork	1

Fig. 2.23 Tabla 1 – Workflow KNIME Ejemplo 1

Tabla 2: En este caso se usan los nodos *Database Row Filter* y *Database Column Filter* los cuales filtran filas y columnas de la tabla seleccionada respectivamente. De igual manera se ejecuta el nodo *Database Connection Table Reader* para leer los datos. Se muestra su salida en la Fig. 2.24.

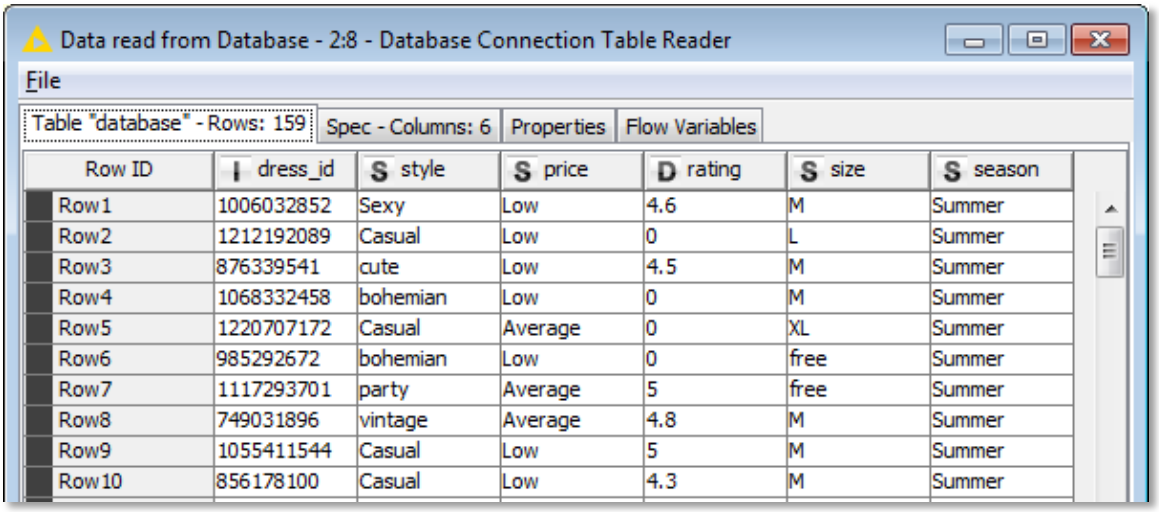


Row ID	dress_id	style	price	rating	size	season
Row1	1006032852	Sexy	Low	4.6	M	Summer
Row2	1212192089	Casual	Low	0	L	Summer
Row3	876339541	cute	Low	4.5	M	Summer
Row4	1068332458	bohemian	Low	0	M	Summer
Row5	1220707172	Casual	Average	0	XL	Summer
Row6	985292672	bohemian	Low	0	free	Summer
Row7	1117293701	party	Average	5	free	Summer
Row8	749031896	vintage	Average	4.8	M	Summer
Row9	1055411544	Casual	Low	5	M	Summer
Row10	856178100	Casual	Low	4.3	M	Summer

Fig. 2.24 Tabla 2 – Workflow KNIME Ejemplo 1

Tabla 3: Esta tabla es generada a razón de filtrar filas y columnas de la selección original, sin embargo, a diferencia de la Tabla 2, los filtros se realizan con un solo nodo; *Database Query* que recibe una consulta SQL con la sintaxis del SMBD con el que se está integrando.

Después *Database Connection Table Reader* ejecuta la consulta y realiza la lectura de los datos. Para visualizar la tabla resultante basta con desplegar el menú contextual del nodo y seleccionar la opción *Data read from Database*. La salida se puede ver en la Fig. 2.25.



Row ID	dress_id	style	price	rating	size	season
Row1	1006032852	Sexy	Low	4.6	M	Summer
Row2	1212192089	Casual	Low	0	L	Summer
Row3	876339541	cute	Low	4.5	M	Summer
Row4	1068332458	bohemian	Low	0	M	Summer
Row5	1220707172	Casual	Average	0	XL	Summer
Row6	985292672	bohemian	Low	0	free	Summer
Row7	1117293701	party	Average	5	free	Summer
Row8	749031896	vintage	Average	4.8	M	Summer
Row9	1055411544	Casual	Low	5	M	Summer
Row10	856178100	Casual	Low	4.3	M	Summer

Fig. 2.25 Tabla 3 – Workflow KNIME Ejemplo 1

Para guardar **KNIME Ejemplo 1** es necesario tenerlo visible en el *Workflow Editor* para después dar clic en el ícono *Save*, o bien, tecleando Ctrl + S.

3. Nodos KNIME

3.1 Proceso ETL

El proceso ETL (por sus siglas en inglés *Extract, Transform, Load*) se deriva del concepto Almacén de Datos o *Data Warehouse* y refiere a tres fases esenciales[32]:

1. **Extracción:** Consiste en extraer datos orientados a un objetivo desde diversas fuentes (bases de datos operacionales, archivos planos, etc.) para traducirlas en un lenguaje estándar que permitirá la transformación.
2. **Transformación:** En esta fase se engloba todo tipo de manipulación de los datos como reducir su dimensión, discretizar campos, filtrar instancias, transponer, tratar valores vacíos y atípicos, etc.
3. **Carga:** La última fase consiste en guardar los datos pre-procesados en una estructura que hará factible su estudio, predicción y visualización.

Existen distintas herramientas de código abierto dedicadas particularmente al proceso ETL como son *Talend Open Source Data Integrator, GeoKettle, Jaspersoft ETL, KETL, Pentaho's Data Integration*, entre otras.

Así mismo, versiones comerciales que persiguen el mismo objetivo; *Talend Studio, Ab Initio , Adeptia , Apatar , Astera Centerprise Data Integrator, SAP Data Integrator, Oracle Warehouse Builder*[33].

[32] *Data Warehouse*. Enero 2016 [En línea] Disponible: http://datawarehouse4u.info/ETL-process.html

[33] *30+ Free and Commercial ETL Tools*. Enero 2016 [En línea] Disponible: http://www.butleranalytics.com/30-etl-tools/

3.2 ETL en KNIME

3.2.1 Extracción

KNIME no solamente es un software de minería de datos, si no también cubre necesidades de ETL.

Los nodos de lectura de datos como *File Reader, CSV Reader, Table Reader, Read Images* y *Table Creator* representan la primera fase del proceso. Al igual que los nodos asociados a una conexión a base de datos como *Database Connection Table Reader* y *Database Reader*. Ver Fig. 3.1.

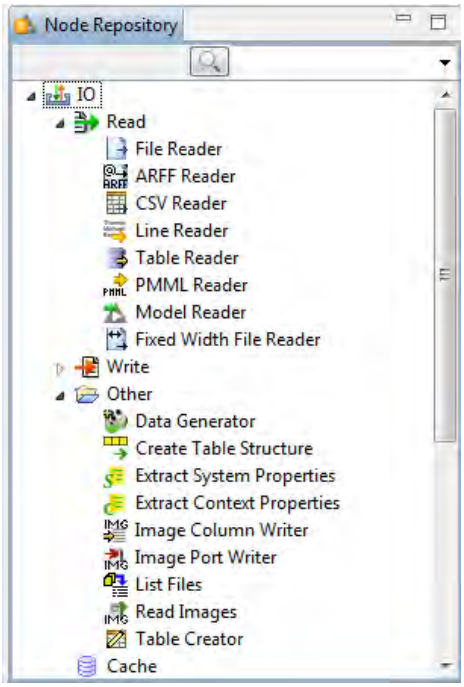
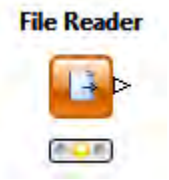


Fig. 3.1 Nodos Lectura

3.2.1.1 IO (Read)

En el agrupado *IO* del repositorio de nodos se pueden encontrar los nodos que leen archivos planos (CVS, TXT, Formato interno) e imágenes.



El nodo más completo utilizado para la lectura de datos provenientes de archivos planos es el ***File Reader***; permite leer archivos en diferentes formatos codificados bajo el estándar ASCII que estén situados en una dirección URL (*Uniform Resource Locator*).

Data Set: Para ilustrar la configuración del nodo tomaremos como insumo el conjunto de datos *Abalone* ubicado en el repositorio público *UCI Machine Learning Repository*[34].

Configuración *File Reader*

Valid URL: Se ingresa la ruta donde se encuentra alojado el archivo (si el archivo está en la PC se tiene la opción del botón *Browse*). Para el ejemplo *Abalone* pegamos la URL HTTP del navegador[35].

[34] Repositorio UCI. Enero 2016 [En línea]. Disponible: http://archive.ics.uci.edu/ml/datasets/Abalone

Preserve user settings for new location: Mantiene las modificaciones hechas en los paneles *Basic Settings* y *Preview*, aunque sea cambiado la URL.

Basic Settings

Read row IDs: Si se habilita, el nodo usa la primera columna como los IDs.

Read column headers: Si se hablita, el nodo usa la primera fila como el nombre de las columnas.

Column delimiter: Es el carácter que separa las columnas del archivo (El nodo procesa internamente la definición del posible separador de las columnas después de ingresar el URL).

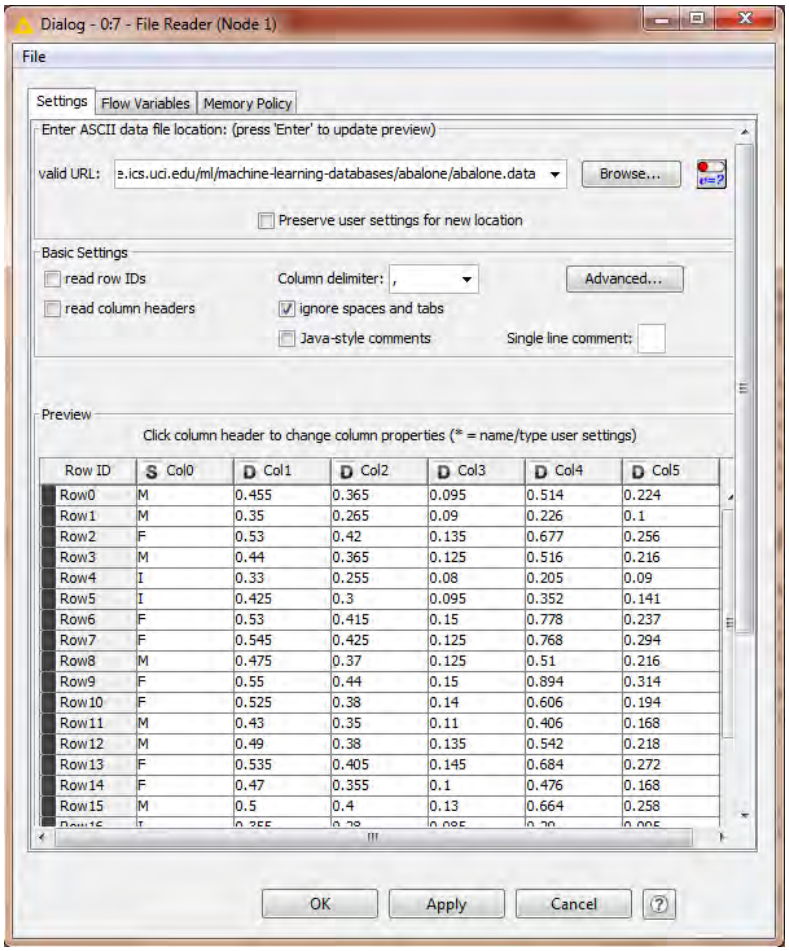


Fig. 3.2 Configuración File Reader

Ignore spaces and tabs: Si es habilitado ignora los espacios y tabulaciones que no están entrecomillados.

Este nodo soporta la opción de agregar comentarios en los archivos sin que sean cargados a la tabla;

- ***Java-style comments:*** usa los caracteres "/*" y "*/", o bien," //"

[35] Repositorio UCI. Enero 2016 [En línea]. Disponible: http://archive.ics.uci.edu/ml/machine-learning-databases/abalone/abalone.data

- ***Single line comment:*** usa el carácter(s) definido por el usuario

Advanced:

- ***Quote Support:*** Permite definir entrecomillados especiales (determinan el principio y el final de una cadena de caracteres especial). De forma predeterminada se tienen \' y \".
- ***Decimal Separator:*** Permite definir caracteres separadores de decimales y millares para los campos de tipo *double*. De forma predeterminada se tiene "." para separar decimales.

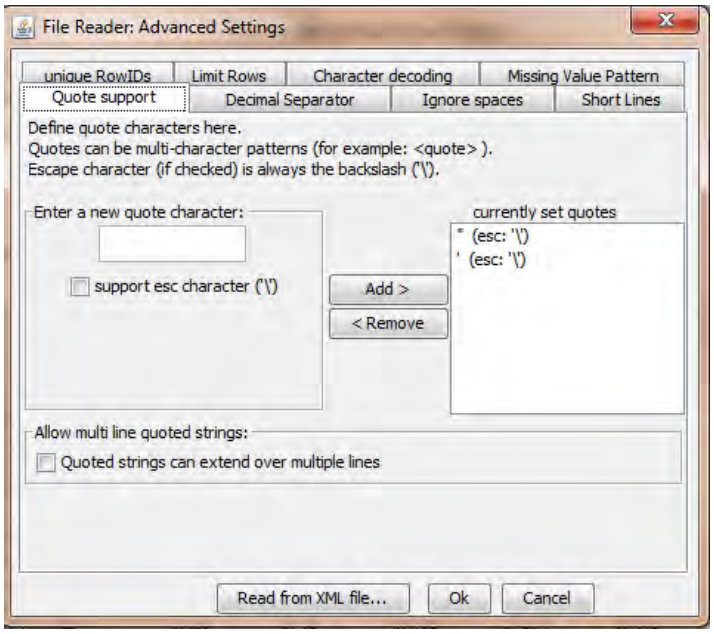


Fig. 3.3 Opciones Avanzadas File Reader

- ***Ignore spaces:*** Bandera que habilita ignorar espacios al final de los registros. De forma predeterminada se tiene deshabilitada para archivos separados por caracteres que no son espacios ni tabulaciones.
- ***Short Lines:*** Bandera que habilita lectura de líneas con muy pocos campos no vacíos.
- ***Unique RowIDs:*** Habilita concatenar un sufijo a los RowIDs repetidos.
- ***Limit Rows:*** Permite leer o excluir un número limitado de registros.
- ***Character decoding:*** En las opciones avanzadas se incluye la posibilidad de leer archivos codificados bajo otros estándares más apropiados para idiomas que no son el inglés, por ejemplo ISO-8859-1 (es una norma de la ISO que define la codificación del alfabeto latino) entre otras normas de codificación. De manera predeterminada se tiene la codificación ASCII.

- ***Missing Value Pattern:*** Permite definir una cadena de caracteres que defina a la vez un valor perdido en cuya celda coincida con la cadena de caracteres (aplica para las variables de tipo caracter).

Para el conjunto de datos *Abalone* dejamos los valores predeterminados de todas las opciones avanzadas.

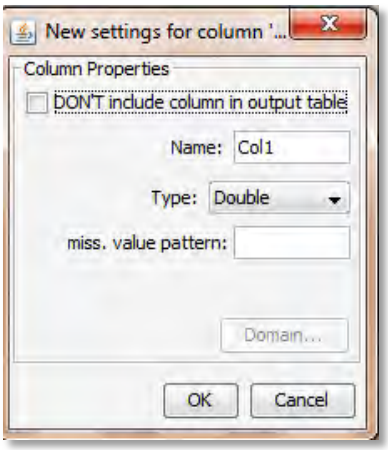


Fig. 3.4 Personalización de Columnas

Preview: Es una vista preliminar de la tabla que es generada después de ejecutar el nodo. Además, se pueden personalizar las columnas (nombre, tipo de dato, valor perdido, si será o no parte salida). El menú que se obtiene dando clic izquierdo en el encabezado de la columna en cuestión.

Aprovechamos este menú para cambiar el nombre de las columnas del conjunto de datos *Abalone*. Ver Fig. 3.4.

Una vez completada la configuración del nodo se da clic en el botón ok en la parte inferior del cuadro de diálogo. Después se ejecuta el nodo dando clic derecho y seleccionando *Execute*, o F7.

Como se mencionó, ***File Reader*** es el nodo más completo que ilustra la lectura de archivos planos. Los demás atienden casos más particulares.

CSV Reader está desarrollado para la lectura de archivos separados por comas (CSV). Funciona bajo los mismos parámetros básicos del nodo ***File Reader*** como son URL, delimitador de columnas, caracter de comentarios, caracter de entrecomillados, banderas para tomar encabezados y RowIDs,

exclusión de líneas con pocos datos y definición de número limitado de filas. Sin embargo, no los contempla todos.

Fixed Width File Reader tiene la característica principal de leer las columnas del archivo definidas por un número de caracteres por columna (no necesariamente el mismo número) en lugar de un caracter delimitador como lo hace ***File Reader***.

Line Reader lee un archivo ASCII en líneas generando una columna con tantas líneas como registros hay en el archivo. Se puede configurar el nombre de la columna, un sufijo para las líneas y limitar el número de líneas.

Table Reader lee una tabla generada en formato interno KNIME y solamente se configura con el URL y habilitar/deshabilitar definición de un número limitado de filas. Las tablas en formato interno se generan con el nodo *Table Writer*.

Los nodos ***List Files*** y ***Read Images*** en conjunto pueden extraer imágenes de tipo SVG y PNG provenientes de una lista de direcciones URL.

- ***List Files:*** Extrae una lista de URLs de los archivos situados en una dirección y los organiza en una tabla (*output*). El nodo tiene la opción de filtrar los archivos de interés (por ejemplo, las extensiones SVG y PNG).

- ***Read Images:*** Recibe en su puerto de entrada una tabla con URLs (donde se alojan las imágenes) y concatena una nueva columna con las imágenes asociadas a la ruta. Su puerto de salida arroja la tabla aumentada.

El nodo ***Table Creator*** permite al usuario ingresar manualmente los valores de las celdas, así como personalizar el nombre de las columnas y filas, y tipo de dato de los campos.

3.2.2 Transformación

KNIME cubre esta segunda fase con las rutinas que se encuentran en el agrupado *Data Manipulation*. Y más modelos pueden ser usados provenientes de otros sistemas a través del lenguaje estándar PMML.

Cabe destacar que cuando los datos fuente están estructurados en una Base de Datos, existe un trato previo mediante las consultas que proveen los nodos *Database* (*Database Query, DataBase Joiner, Database Row Filter, Database Colunm Filter*, etc.). Ver Fig. 3.5.

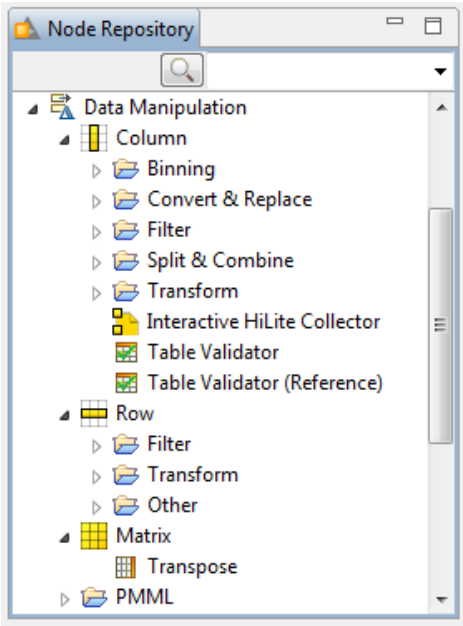


Fig. 3.5 Nodos de pre-procesamiento

Es importante mencionar que todo trato de los datos previo al modelo de minería es considerado como el pre-procesamiento o preparación de los datos y como ya se explicó se encuentra contenido en el KDD.

De manera global la preparación de datos puede ser clasificada en 4 tipos de estrategias [4]:

1. Limpieza de los datos (valores perdidos, valores atípicos y ruido)
2. Reducción de los datos (muestreo y selección de datos relevantes)

3. Integración de los datos (agregaciones, unión de tablas, entre otras)
4. Transformación de los datos (modificaciones sintácticas de los datos)

Dichas estrategias no son mutuamente excluyentes, es decir, hay técnicas como la compactación que es una combinación de reducción e integración [4]. Otro ejemplo es el *Binning*, técnica que sirve para "atenuar ruidos blancos" y al mismo tiempo se considera una reducción a los datos al ser discretizados.

3.2.2.1 Binning

El *binning* es una técnica de discretización de datos que consiste en trasformar o categorizar los valores de un atributo de tipo numérico en un número finito de intervalos llamados *bins*. Ver Fig. 3.6.

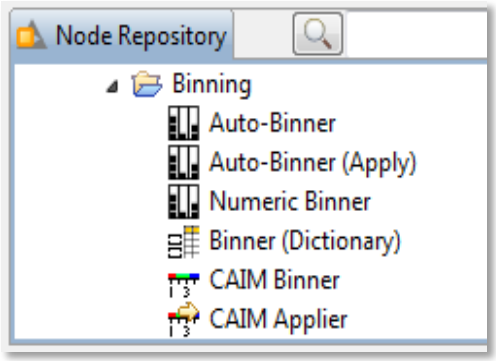
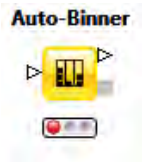


Fig. 3.6 Nodos Binning

Existen dos tipos de algoritmos:

1. **No supervisados**: no se tiene conocimiento a priori sobre los objetivos.
2. **Supervisados:** Se orientan los resultados hacia un objetivo.



Auto-Biner en KNIME es un nodo representativo de *Binning*. Su puerto de entrada recibe una tabla para ser categorizada y devuelve la tabla con los *bins* definidos. También cuenta con un puerto de donde se puede tomar el modelo cuyos parámetros se configuraron en el nodo a través del estándar PMML. Ver Fig. 3.7.

Configuración *Auto-Binner*

Manual Selection: En el cuadro de diálogo es posible definir manualmente las variables que se desean discretizar. Para el ejemplo *Abalone*, solo incluimos la variable *Length*. Ver Fig. 3.7.

Binning Method: El número de *bins* o intervalos son definidos por el usuario de dos formas diferentes; intervalos de la misma longitud, o bien, por cuantiles. Para el ejemplo *Abalone* definimos 4 intervalos de la misma longitud.

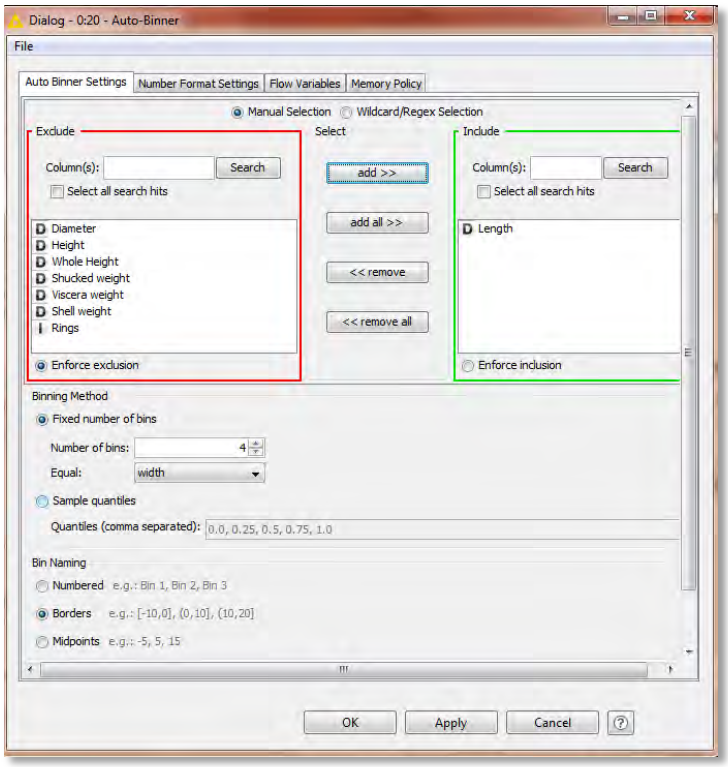


Fig. 3.7 Configuración Auto-Binner

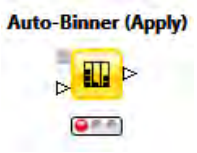
Binnig Name: Es posible editar el nombre de la etiqueta.

Force integer bounds: Bandera que genera intervalos de números enteros.

Replace target column(s): Bandera que reemplaza la columna que está en proceso de categorización en lugar de concatenar una nueva.

Number Format Settings: El nodo también permite editar el formato de los *bins* en proceso. Para el ejemplo mantenemos la configuración predeterminada.

Entre los nodos incluidos en la descarga de la plataforma KNIME existen más relacionados con el *binning*:



Auto-Binner (Apply) recibe en uno de sus puertos de entrada la configuración predefinida en ***Auto-Binner***, así como una tabla que contenga el atributo asociado al modelo. No necesita configuración ya que hereda los parámetros de ***Auto-Binner.***

Numeric Binner contiene un método no supervisado donde el usuario ingresa los intervalos con base en su criterio.

Binner (Dictionary) tiene la característica de recibir dos tablas; la primera es la que será categorizada y la otra (llamada diccionario) la que contiene los rangos con su respectiva etiqueta (nombre del *bin*) que será usada según los casos que caigan en dichos rangos. Ver Fig. 3.8.

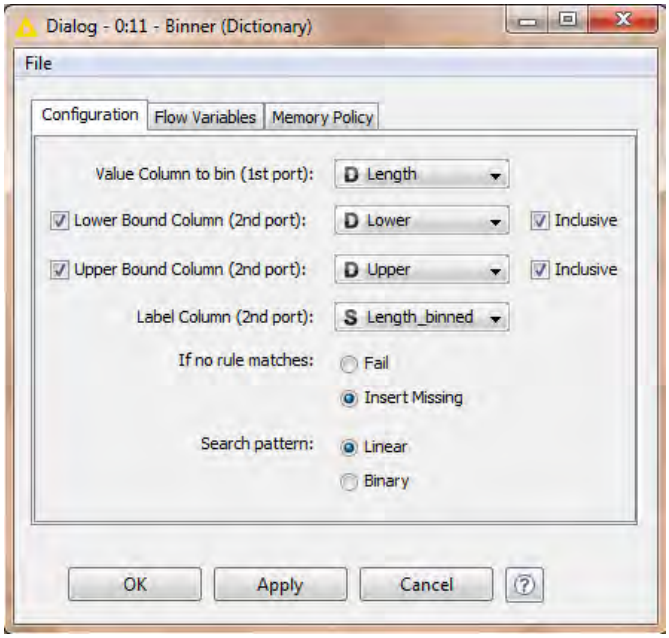


Fig. 3.8 Configuración Binner (Dictionary)

El nodo provee de dos formas de buscar el rango en donde cae cada valor:

- **Lineal:** Busca secuencialmente en el orden de la tabla diccionario y devuelve la etiqueta asociado a la primera coincidencia.
- **Binario:** Ordena los rangos y solo funciona solo devuelve la etiqueta correspondiente si o hay vacíos ni se sobreponen los intervalos.

KNIME cuenta con un algoritmo *binning* **supervisado**, CAIM (por sus siglas en inglés *class-attribute interdependence maximization),* el cual está enfocado en reducir el número de intervalos y maximizar la interdependencia con un atributo llamado clase [5].

La forma en que se configura el nodo es sencilla puesto que el modelo ya está definido. Basta con indicar cuales columnas serán reducidas y la columna-clase asociada.

CAIM Binner tiene la opción de transportar a través de su puerto de salida el modelo con parámetros hacia el nodo ***CAIM Binner Applier***.

La Figs. 3.9 muestra tres diferentes maneras de discretizar el campo *Length* del *Data Set Abalone* con los nodos básicos provee KNIME (cargado previamente con el nodo ***File Reader***):

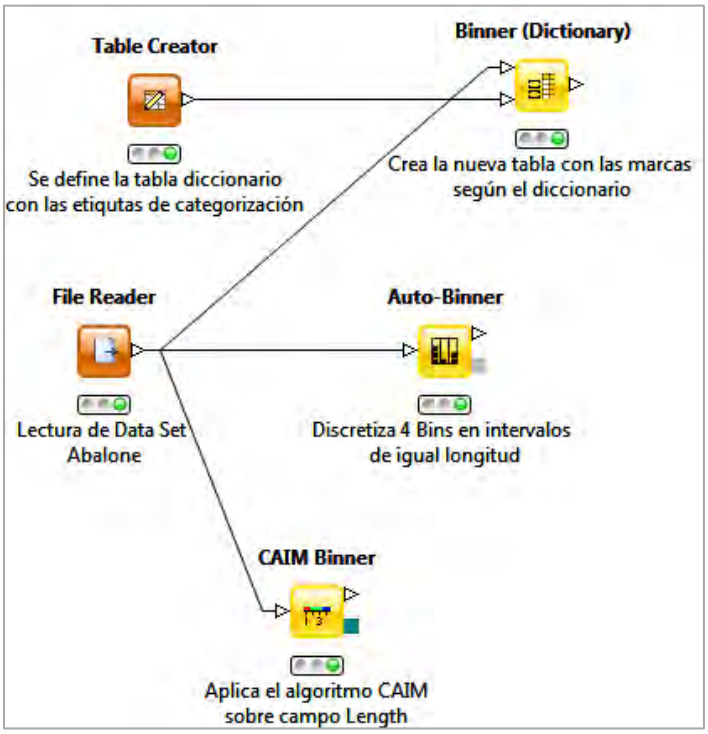


Fig. 3.9 Workflow Binning

A continuación se muestra la salida del nodo *Auto-Binner* donde se puede ver el campo *Length [Binned]* que contiene los intervalos que discretizan al campo *Length.* Ver Fig. 3.10.

Binned Data - 0:20 - Auto-Binner (Discretiza 4 Bins en intervalo...)

File

Table "default" - Rows: 4177 | Spec - Columns: 10 | Properties | Flow Variables

Row ID	Sex	Length	Diameter	Height	Whole ...	Shucke...	Viscera ...	Shell w...	Rings	Length [Binned]
Row0	M	0.455	0.365	0.095	0.514	0.224	0.101	0.15	15	(0.445,0.630]
Row1	M	0.35	0.265	0.09	0.226	0.1	0.048	0.07	7	(0.260,0.445]
Row2	F	0.53	0.42	0.135	0.677	0.256	0.142	0.21	9	(0.445,0.630]
Row3	M	0.44	0.365	0.125	0.516	0.216	0.114	0.155	10	(0.260,0.445]
Row4	I	0.33	0.255	0.08	0.205	0.09	0.04	0.055	7	(0.260,0.445]
Row5	I	0.425	0.3	0.095	0.352	0.141	0.078	0.12	8	(0.260,0.445]
Row6	F	0.53	0.415	0.15	0.778	0.237	0.142	0.33	20	(0.445,0.630]
Row7	F	0.545	0.425	0.125	0.768	0.294	0.15	0.26	16	(0.445,0.630]
Row8	M	0.475	0.37	0.125	0.51	0.216	0.112	0.165	9	(0.445,0.630]
Row9	F	0.55	0.44	0.15	0.894	0.314	0.151	0.32	19	(0.445,0.630]
Row10	F	0.525	0.38	0.14	0.606	0.194	0.148	0.21	14	(0.445,0.630]

Fig. 3.10 Salida de Nodo Auto-Binner

Cabe mencionar que en la literatura existen más métodos supervisados para discretización como: X^2 Discretization, Maximum Entropy Discretization, CAIR Discretization, K-means clustering, One-level Decision Tree, Dynamic Attribute, Paterson and Niblett. [36]

3.2.2.2 Convertir y Reemplazar

Parte de la preparación de datos consiste realizar adecuaciones a los **metadatos** y **transformaciones** a los datos que "no supongan un cambio en el significado de los datos". En KNIME existen nodos con tareas de ese tipo; renombrar atributos, convertir el tipo de dato (carácter a numérico y viceversa, decimal a entero, personalización de redondeo decimal), editar el dominio de atributos, entre otras. KNIME tiene nodos que cumplen dichas tareas en la subcategoría ***Convert & Replace***. Ver Figura 3.11.

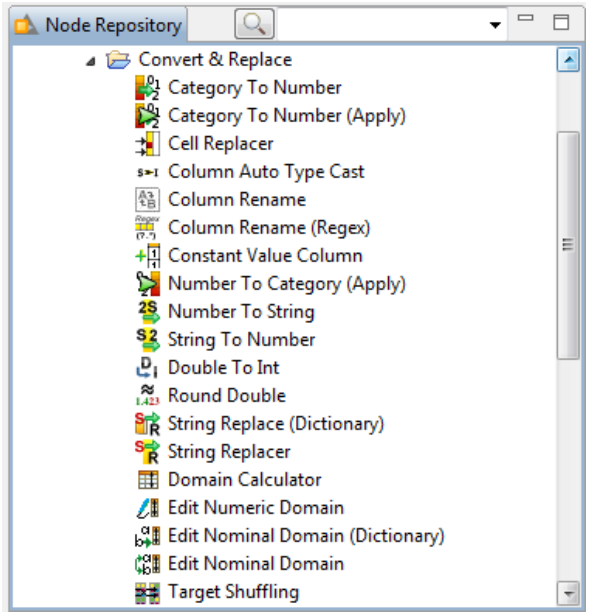


Fig. 3.11 Nodos Convertir y Reemplazar

La manera de configurar estos nodos es similar a los anteriormente descritos; primero leyendo y enlazando los insumos, después desde un cuadro de diálogo e ingresando los parámetros relacionados con la tarea.

A continuación, se describen algunos de los nodos con un workflow (Fig. 3.12). Se toma el ***Data Set*** *Adult* ubicado en el repositorio público *UCI Machine Learning Repository*[37]

[36] *SlidePlayer*, *Discretization,* Enero 2016 [En línea] Disponible: http://slideplayer.com/slide/6718633/
[37] *Adult*. Repositorio Público UCI. Enero 2016 [En línea] Disponible: http://archive.ics.uci.edu/ml/datasets/Adult

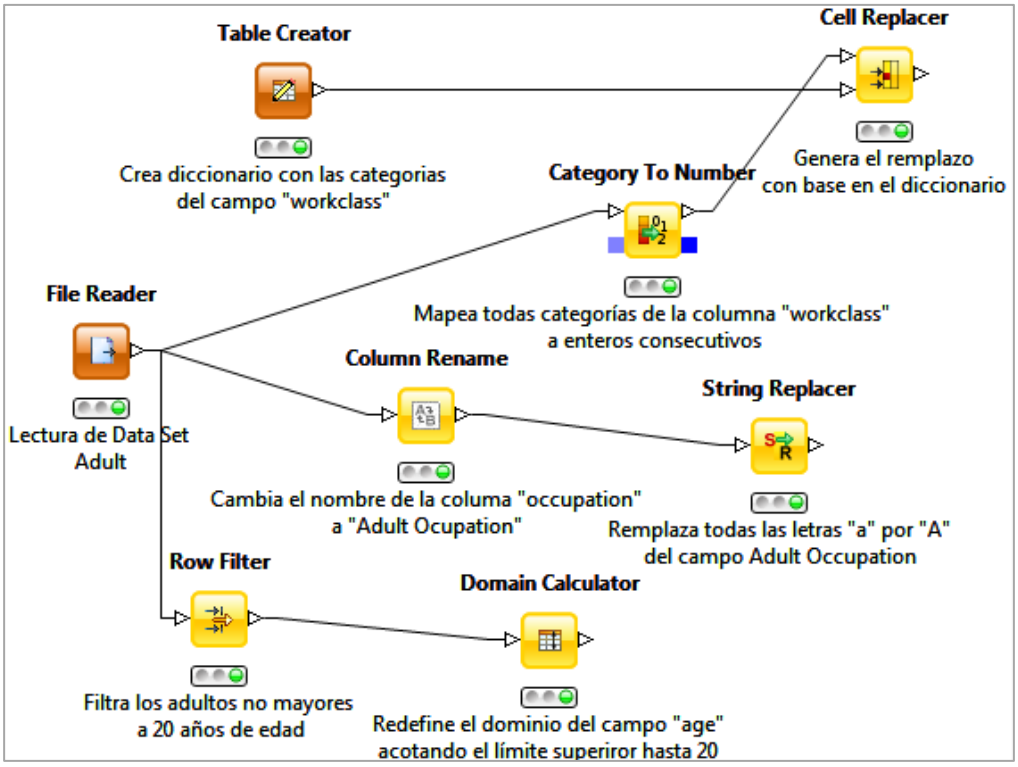
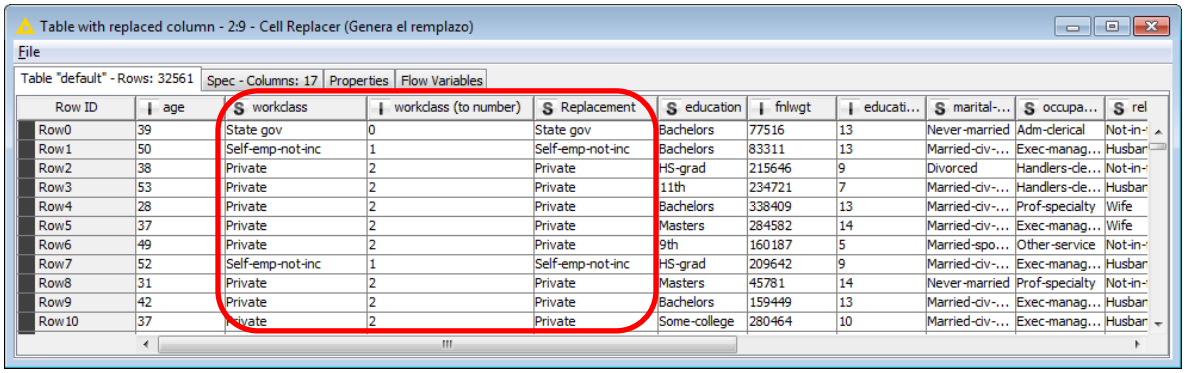


Fig. 3.12 Workflow Convertir y Reemplazar

Category To Number esencialmente genera un índice para cada valor posible de uno o varios campos de una tabla. Adicionalmente, puede pasar al lenguaje estándar PMML dicho modelo de indexación.

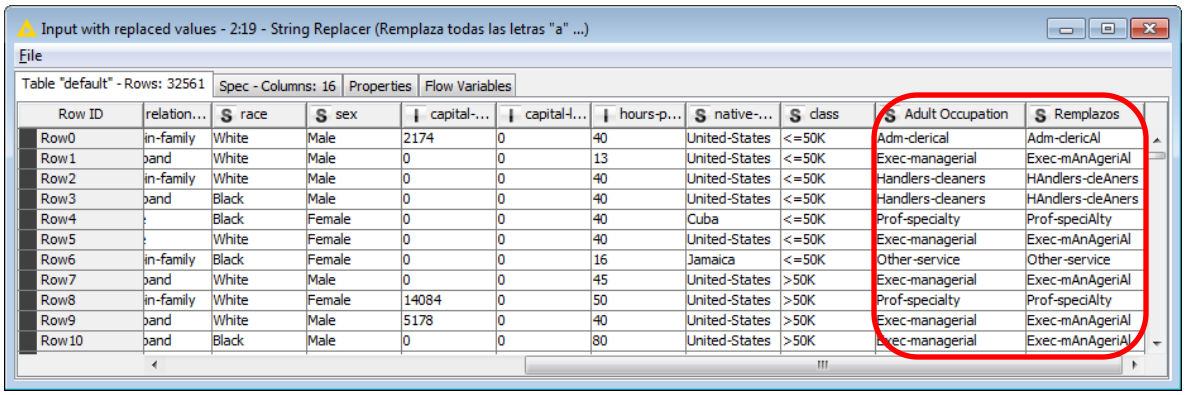
Cell Replacer se configura muy similar al nodo ***Binner Diccionary***, recibiendo dos tablas para su ejecución. Solo que no se limita a atributos de tipo numérico. Ver Fig. 3.13.



Row ID	age	workclass	workclass (to number)	Replacement	education	fnlwgt	educati...	marital-...	occupa...	rel
Row0	39	State gov	0	State gov	Bachelors	77516	13	Never-married	Adm-clerical	Not-in-
Row1	50	Self-emp-not-inc	1	Self-emp-not-inc	Bachelors	83311	13	Married-civ-...	Exec-manag...	Husbar
Row2	38	Private	2	Private	HS-grad	215646	9	Divorced	Handlers-cle...	Not-in-
Row3	53	Private	2	Private	11th	234721	7	Married-civ-...	Handlers-cle...	Husbar
Row4	28	Private	2	Private	Bachelors	338409	13	Married-civ-...	Prof-specialty	Wife
Row5	37	Private	2	Private	Masters	284582	14	Married-civ-...	Exec-manag...	Wife
Row6	49	Private	2	Private	9th	160187	5	Married-spo...	Other-service	Not-in-
Row7	52	Self-emp-not-inc	1	Self-emp-not-inc	HS-grad	209642	9	Married-civ-...	Exec-manag...	Husbar
Row8	31	Private	2	Private	Masters	45781	14	Never-married	Prof-specialty	Not-in-
Row9	42	Private	2	Private	Bachelors	159449	13	Married-civ-...	Exec-manag...	Husbar
Row10	37	Private	2	Private	Some-college	280464	10	Married-civ-...	Exec-manag...	Husbar

Fig. 3.13 Salida de Nodo Cell Replacer

String Replacer puede reemplazar total o parcialmente los valores de las celdas de un campo de tipo carácter. Ver Fig. 3.14.



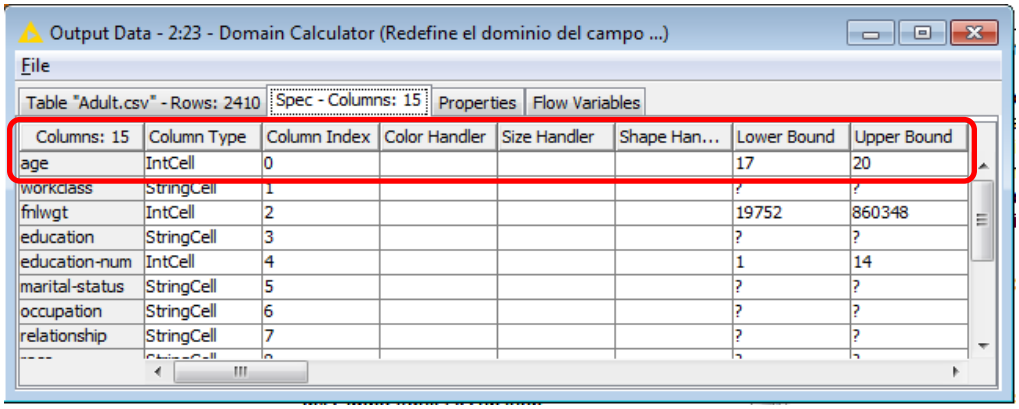
Row ID	relation...	race	sex	capital-...	capital-l...	hours-p...	native-...	class	Adult Occupation	Remplazos
Row0	in-family	White	Male	2174	0	40	United-States	<=50K	Adm-clerical	Adm-clericAl
Row1	band	White	Male	0	0	13	United-States	<=50K	Exec-managerial	Exec-mAnAgeriAl
Row2	in-family	White	Male	0	0	40	United-States	<=50K	Handlers-cleaners	HAndlers-cleAners
Row3	band	Black	Male	0	0	40	United-States	<=50K	Handlers-cleaners	HAndlers-cleAners
Row4		Black	Female	0	0	40	Cuba	<=50K	Prof-specialty	Prof-speciAlty
Row5		White	Female	0	0	40	United-States	<=50K	Exec-managerial	Exec-mAnAgeriAl
Row6	in-family	Black	Female	0	0	16	Jamaica	<=50K	Other-service	Other-service
Row7	band	White	Male	0	0	45	United-States	>50K	Exec-managerial	Exec-mAnAgeriAl
Row8	in-family	White	Female	14084	0	50	United-States	>50K	Prof-specialty	Prof-speciAlty
Row9	band	White	Male	5178	0	40	United-States	>50K	Exec-managerial	Exec-mAnAgeriAl
Row10	band	Black	Male	0	0	80	United-States	>50K	Exec-managerial	Exec-mAnAgeriAl

Fig. 3.14 Salida de Nodo String Replacer

Nota: Los tres nodos descritos incluyen en su cuadro de diálogo la opción para concatenar la columna con los cambios realizados, lo que ofrece integridad a los datos originales.

El nodo ***Dominan Calculator*** edita los metadatos de la tabla. En KNIME todas las tablas generadas, son también indexadas en otras tablas con sus especificaciones como número de columnas, tipo de dato, color y tamaño (en algunos casos), posibles valores y cotas inferiores y superiores. ***Dominan Calculator*** redefine los posibles valores y las cotas para los campos que el usuario desee. Ver Fig. 3.15.

En la práctica, puede ser de interés obtener estadísticos y/o definir nuevos histogramas de un subconjunto del universo original como conjunto de datos independiente por lo que es necesario redefinir el dominio.

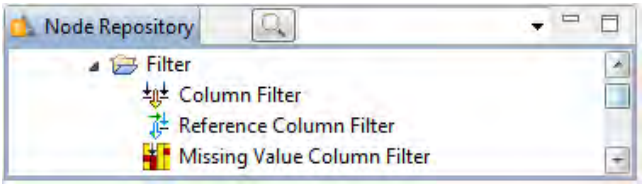


Columns: 15	Column Type	Column Index	Color Handler	Size Handler	Shape Han...	Lower Bound	Upper Bound
age	IntCell	0				17	20
workclass	StringCell	1				?	?
fnlwgt	IntCell	2				19752	860348
education	StringCell	3				?	?
education-num	IntCell	4				1	14
marital-status	StringCell	5				?	?
occupation	StringCell	6				?	?
relationship	StringCell	7				?	?

Fig. 3.15 Salida de Nodo Dominan Calculator

Otros nodos variantes para la edición del dominio son ***Edit Numeric Dominan, Edit Nominal Dominan (Dictionary), Edit Nominal Dominan.***

3.2.2.3 Filtro Columnas



Parte de las tareas de **reducción** de datos se encuentran en los grupos Filter (Column). Ver Fig. 3.16.

Fig. 3.16 Nodos Filtro Columna

Se describen los tres nodos que KNIME incluye en el grupo en la Figura 3.17.

Data Set: *Adult* ubicado en el repositorio público *UCI Machine Learning Repository*[38]

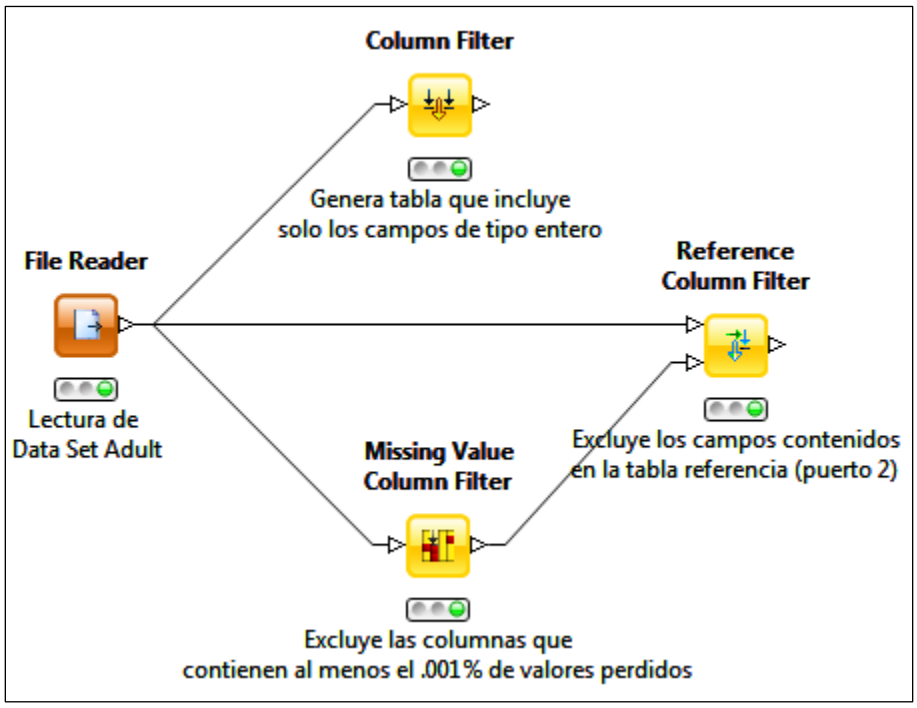
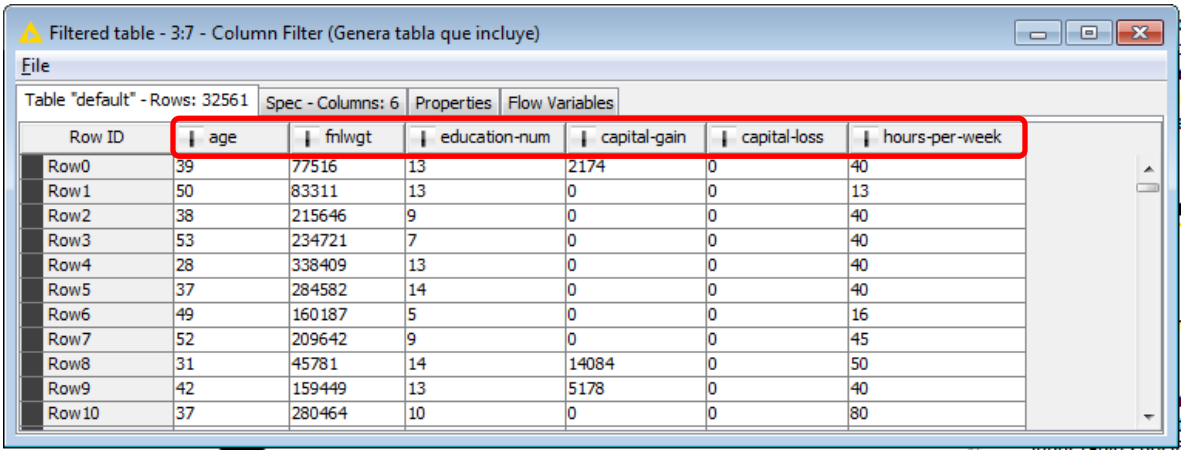


Fig. 3.17 Workflow Filtro Columna

[38] *Adult*. Repositorio Público UCI. Enero 2016 [En línea] Disponible: http://archive.ics.uci.edu/ml/datasets/Adult

Column Filter recibe una tabla en su puerto de entrada y genera otra reducida con las columnas especificadas en el cuadro de diálogo. Ver Fig. 3.18.


Filtered table - 3:7 - Column Filter (Genera tabla que incluye)

File

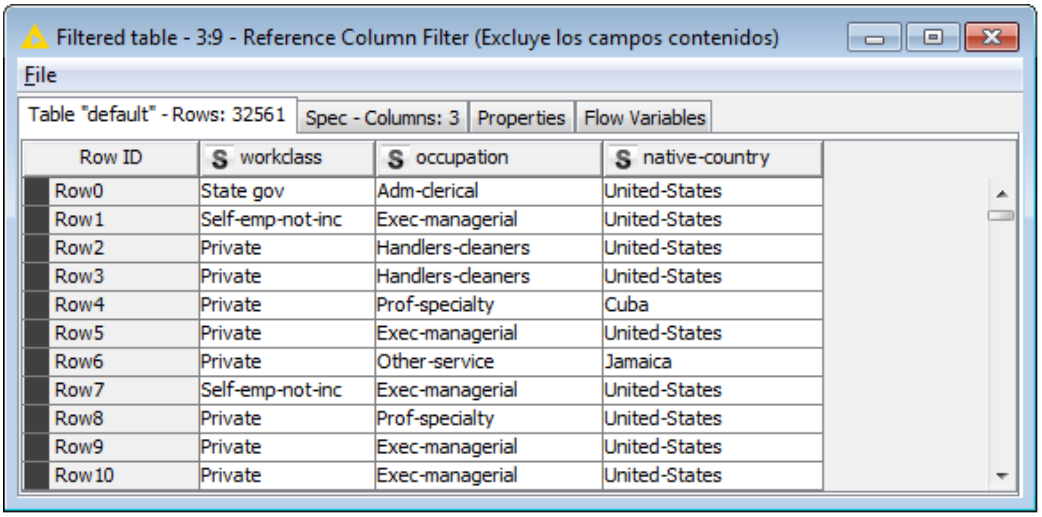
Table "default" - Rows: 32561 | Spec - Columns: 6 | Properties | Flow Variables

Row ID	age	fnlwgt	education-num	capital-gain	capital-loss	hours-per-week
Row0	39	77516	13	2174	0	40
Row1	50	83311	13	0	0	13
Row2	38	215646	9	0	0	40
Row3	53	234721	7	0	0	40
Row4	28	338409	13	0	0	40
Row5	37	284582	14	0	0	40
Row6	49	160187	5	0	0	16
Row7	52	209642	9	0	0	45
Row8	31	45781	14	14084	0	50
Row9	42	159449	13	5178	0	40
Row10	37	280464	10	0	0	80

Fig. 3.18 Salida de Nodo Column Filter

Missing Value Column Filter es una variante de *Column Filter* que incluye en su cuadro de diálogo el parámetro *Missing Value threshold*; cota inferior (% de valores perdidos) que determina los campos que serán excluidos.

Reference Column Filter recibe en el primer puerto la tabla que será filtrada y en el segundo la tabla que contiene los campos referidos. Los campos referidos pueden ser objeto de exclusión o inclusión. En este caso, contiene todas las columnas que tienen al menos un valor vacío del ***Data Set Adult***. Ver Fig. 3.19.


Filtered table - 3:9 - Reference Column Filter (Excluye los campos contenidos)

File

Table "default" - Rows: 32561 | Spec - Columns: 3 | Properties | Flow Variables

Row ID	S workclass	S occupation	S native-country
Row0	State gov	Adm-clerical	United-States
Row1	Self-emp-not-inc	Exec-managerial	United-States
Row2	Private	Handlers-cleaners	United-States
Row3	Private	Handlers-cleaners	United-States
Row4	Private	Prof-specialty	Cuba
Row5	Private	Exec-managerial	United-States
Row6	Private	Other-service	Jamaica
Row7	Self-emp-not-inc	Exec-managerial	United-States
Row8	Private	Prof-specialty	United-States
Row9	Private	Exec-managerial	United-States
Row10	Private	Exec-managerial	United-States

Fig. 3.19 Salida de Nodo Reference Column Filter

3.2.2.4 Dividir y Combinar

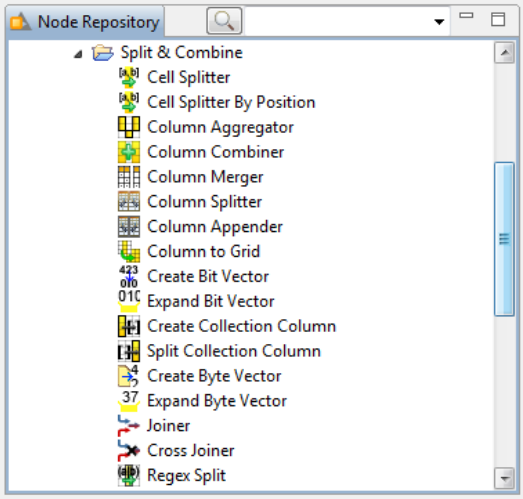


Fig. 3.20 Nodos Dividir y Combinar

Algunas tareas de **integración** de datos están en el grupo **Split & Combine** en el repositorio de nodos. Ver Fig. 3.20.

Quizá estas sean las tareas más comunes soportadas por casi cualquier software de pre procesamiento de datos.

Nota: Los nodos están diseñados para brindar integridad a los datos iniciales al incluir en sus cuadros de configuración la opción de concatenar las columnas con nuevos valores.

A continuación, se muestra un ejemplo de workflow de grupo de nodos Dividir y Combinar. Ver Figura 3.21.

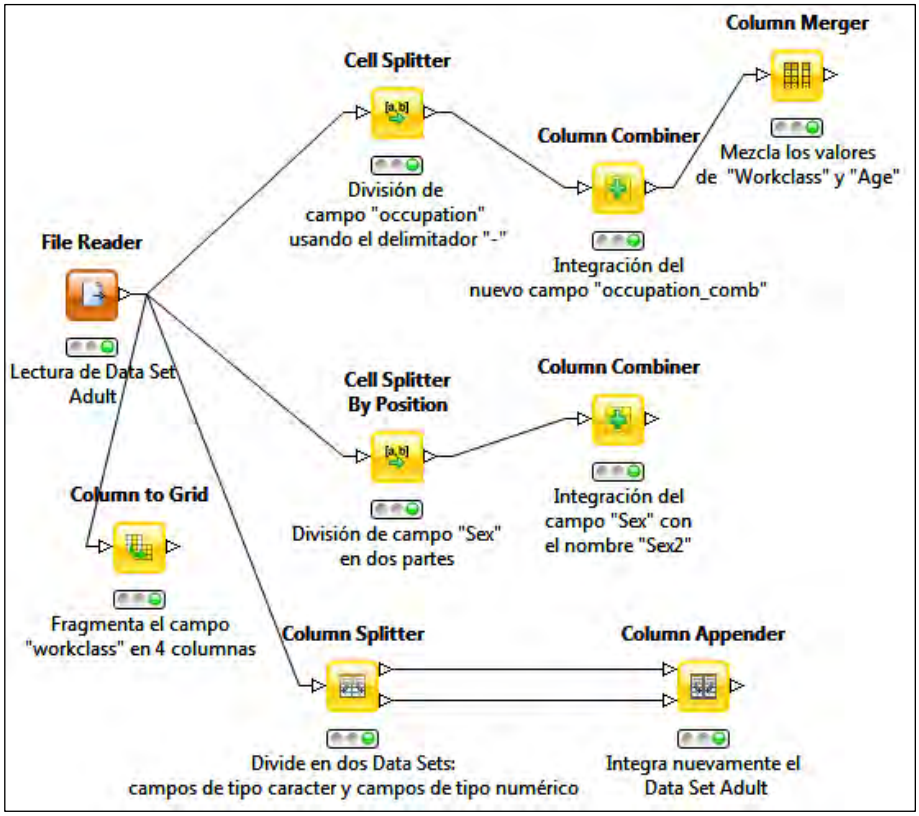
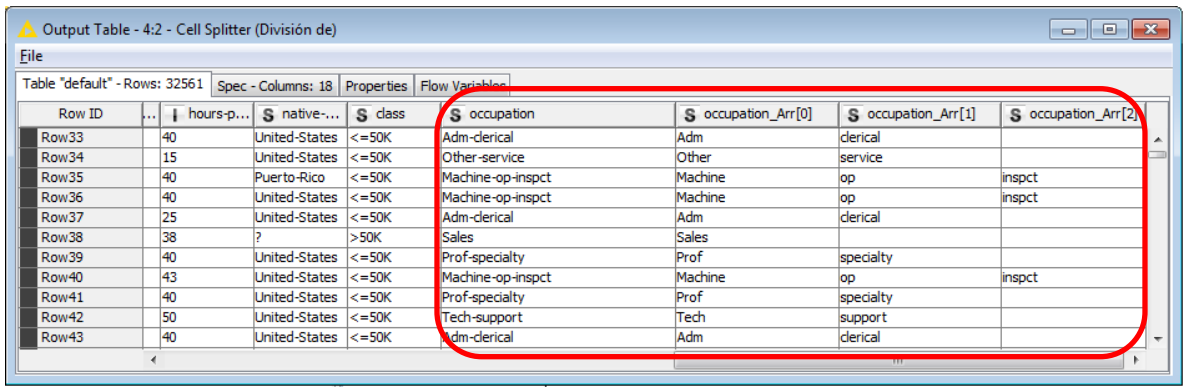


Fig. 3.21 Workflow Divide y Combina

Cell Spliter: es un nodo que divide en partes las celdas de una columna específica de tipo caracter con base en un delimitador y genera otra con las divisiones realizadas (columnas por cada corte o una lista). Ver Fig. 3.22.

Entrada: la tabla que será dividida **Salida:** la tabla con las columnas adicionales

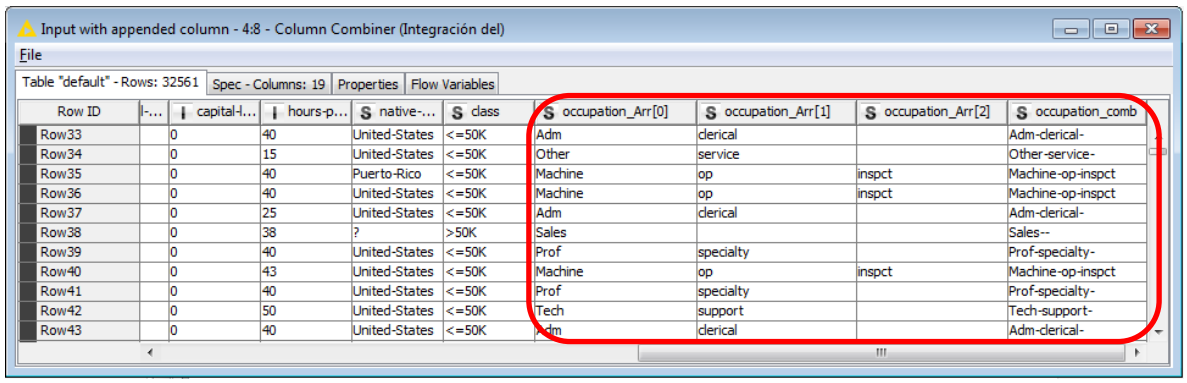


Row ID	hours-p...	native-...	class	occupation	occupation_Arr[0]	occupation_Arr[1]	occupation_Arr[2]
Row33	40	United-States	<=50K	Adm-clerical	Adm	clerical	
Row34	15	United-States	<=50K	Other-service	Other	service	
Row35	40	Puerto-Rico	<=50K	Machine-op-inspct	Machine	op	inspct
Row36	40	United-States	<=50K	Machine-op-inspct	Machine	op	inspct
Row37	25	United-States	<=50K	Adm-clerical	Adm	clerical	
Row38	38	?	>50K	Sales	Sales		
Row39	40	United-States	<=50K	Prof-specialty	Prof	specialty	
Row40	43	United-States	<=50K	Machine-op-inspct	Machine	op	inspct
Row41	40	United-States	<=50K	Prof-specialty	Prof	specialty	
Row42	50	United-States	<=50K	Tech-support	Tech	support	
Row43	40	United-States	<=50K	Adm-clerical	Adm	clerical	

Fig. 3.22 Salida de Nodo Cell Splitter

Column Combiner: el nodo que funciona como la operación inversa del *Cell Splitter*. Es decir, integra los elementos de dos o más columnas de tipo numérico o carácter. Ver Fig. 3.23.

Entrada: Una tabla **Salida:** Tabla original con la columna combinada concatenada.

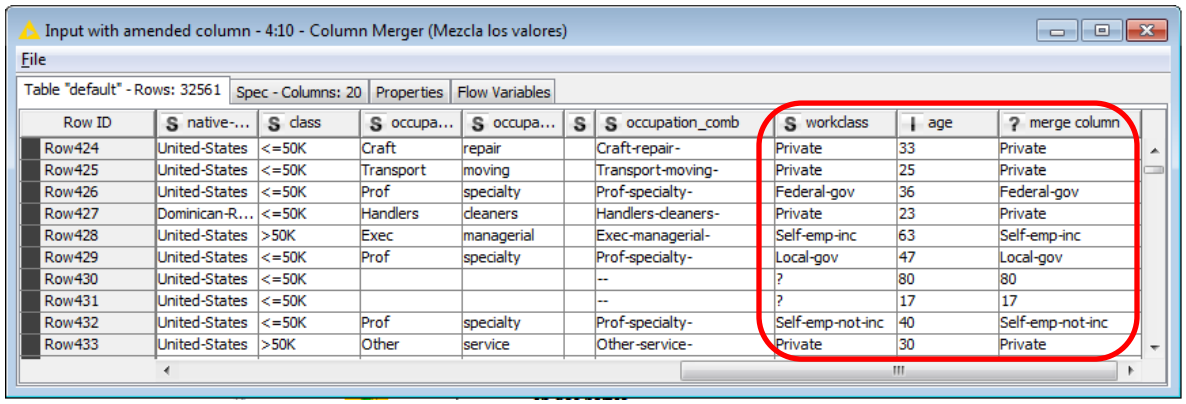


Row ID	capital-l...	hours-p...	native-...	class	occupation_Arr[0]	occupation_Arr[1]	occupation_Arr[2]	occupation_comb
Row33	0	40	United-States	<=50K	Adm	clerical		Adm-clerical-
Row34	0	15	United-States	<=50K	Other	service		Other-service-
Row35	0	40	Puerto-Rico	<=50K	Machine	op	inspct	Machine-op-inspct
Row36	0	40	United-States	<=50K	Machine	op	inspct	Machine-op-inspct
Row37	0	25	United-States	<=50K	Adm	clerical		Adm-clerical-
Row38	0	38	?	>50K	Sales			Sales--
Row39	0	40	United-States	<=50K	Prof	specialty		Prof-specialty-
Row40	0	43	United-States	<=50K	Machine	op	inspct	Machine-op-inspct
Row41	0	40	United-States	<=50K	Prof	specialty		Prof-specialty-
Row42	0	50	United-States	<=50K	Tech	support		Tech-support-
Row43	0	40	United-States	<=50K	Adm	clerical		Adm-clerical-

Fig. 3.23 Salida de Column Combiner

Column Merger reemplaza los valores perdidos de un campo (Primary Column) con los valores de otro (Secondary Column) de la misma tabla dando como resultado un nuevo campo con la mezcla de los dos. Ver Fig. 3.24.

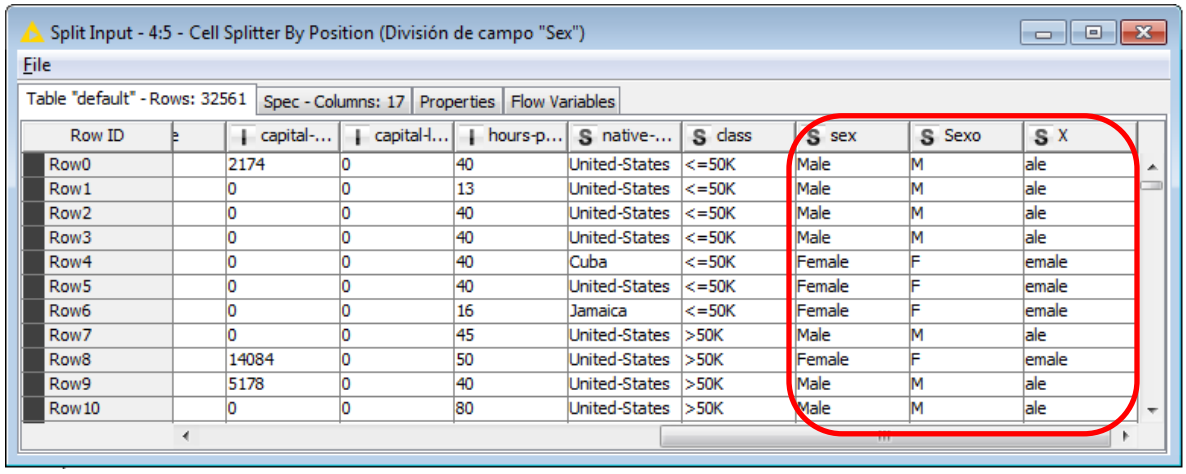
Entrada: Tabla con las columnas que serán mezcladas. **Salida:** Tabla con la nueva columna.



Row ID	native-...	class	occupa...	occupa...	occupation_comb	workclass	age	merge column
Row424	United-States	<=50K	Craft	repair	Craft-repair-	Private	33	Private
Row425	United-States	<=50K	Transport	moving	Transport-moving-	Private	25	Private
Row426	United-States	<=50K	Prof	specialty	Prof-specialty-	Federal-gov	36	Federal-gov
Row427	Dominican-R...	<=50K	Handlers	cleaners	Handlers-cleaners-	Private	23	Private
Row428	United-States	>50K	Exec	managerial	Exec-managerial-	Self-emp-inc	63	Self-emp-inc
Row429	United-States	<=50K	Prof	specialty	Prof-specialty-	Local-gov	47	Local-gov
Row430	United-States	<=50K			--	?	80	80
Row431	United-States	<=50K			--	?	17	17
Row432	United-States	<=50K	Prof	specialty	Prof-specialty-	Self-emp-not-inc	40	Self-emp-not-inc
Row433	United-States	>50K	Other	service	Other-service-	Private	30	Private

Fig. 3.24 Salida de Nodo Column Merger

Cell Splitter tiene una variante en KNIME la cual es ***Cell Splitter By Position***, el objetivo es el mismo con una deferencia en la forma de definir la división, ya que éste último usa el parámetro posición de corte viendo la cadena de caracteres con un orden de izquierda a derecha. Ver Fig. 3.25.



Row ID	e	capital-...	capital-l...	hours-p...	native-...	class	sex	Sexo	X
Row0		2174	0	40	United-States	<=50K	Male	M	ale
Row1		0	0	13	United-States	<=50K	Male	M	ale
Row2		0	0	40	United-States	<=50K	Male	M	ale
Row3		0	0	40	United-States	<=50K	Male	M	ale
Row4		0	0	40	Cuba	<=50K	Female	F	emale
Row5		0	0	40	United-States	<=50K	Female	F	emale
Row6		0	0	16	Jamaica	<=50K	Female	F	emale
Row7		0	0	45	United-States	>50K	Male	M	ale
Row8		14084	0	50	United-States	>50K	Female	F	emale
Row9		5178	0	40	United-States	>50K	Male	M	ale
Row10		0	0	80	United-States	>50K	Male	M	ale

Fig. 3.25 Salida de Cell Splitter By Position

Column Appender concatena dos tablas con el mismo número de registros y con el orden adecuado para la ejecución.

Entrada: Dos tablas ordenadas con el mismo número de registros

Salida: La tabla consolidada

Contexto de uso: Es una alternativa rápida al nodo **Joiner** cuando se tiene certeza del emparejamiento de cada tupla (tablas correctamente ordenadas).

Column to Grid fragmenta uno o más campos de una tabla en tantas partes como se especifique en la configuración, y las acomoda en columnas a modo de cuadricula. Ver Fig. 3.26.

Entrada: Tabla con los campos por fragmentar **Salida:** Tabla en forma de cuadricula con las columnas.

Grid table - 4:14 - Column to Grid (Fragmenta el campo)

File

Table "Adult.csv" - Rows: 8141 | Spec - Columns: 4 | Properties | Flow Variables

Row ID	workclass (0)	workclass (1)	workclass (2)	workclass (3)
Row0	State gov	Self-emp-not-inc	Private	Private
Row1	Private	Private	Private	Self-emp-not-inc
Row2	Private	Private	Private	State-gov
Row3	Private	Private	Private	Private
Row4	Self-emp-not-inc	Private	Private	Self-emp-not-inc
Row5	Private	Private	Federal-gov	Private
Row6	Private	Local-gov	Private	?
Row7	Private	Private	Local-gov	Private
Row8	Private	Federal-gov	State-gov	Private
Row9	Private	Private	Private	Self-emp-not-inc
Row10	Private	Self-emp-not-inc	Private	Private

Fig. 3.26 Salida de Column to Grid

Contexto de uso: Es usado para visualizar mejor los registros de una tabla, sobre todo cuando hay imágenes.

Joiner ejecuta la operación *join* en cualquiera de sus formas (*left join, rigth join, full join, inner join*) con la opción de personalizar los campos que son el criterio de integración.

Entrada: Primera entrada corresponde a la tabla izquierda, segunda entrada corresponde a la tabla derecha **Salida:** Tabla integrada

Contexto de uso: En muchas ocasiones los datos relevantes no están contenidos en una sola tabla, lo que sugiere el uso del **Joiner**.

Cross Joiner integra de manera cruzada, es decir, junta cada registro de la tabla izquierda con cada uno de la tabla derecha. Se recomienda el uso de este nodo solamente cuando el nodo **Joiner** no cumpla con las necesidades de la integración, ya que es una operación costosa en cuanto a tiempo y memoria.

Entrada: Primera entrada corresponde a la tabla izquierda, segunda entrada corresponde a la tabla derecha **Salida:** Tabla integrada

3.2.2.5 Transformar Columnas

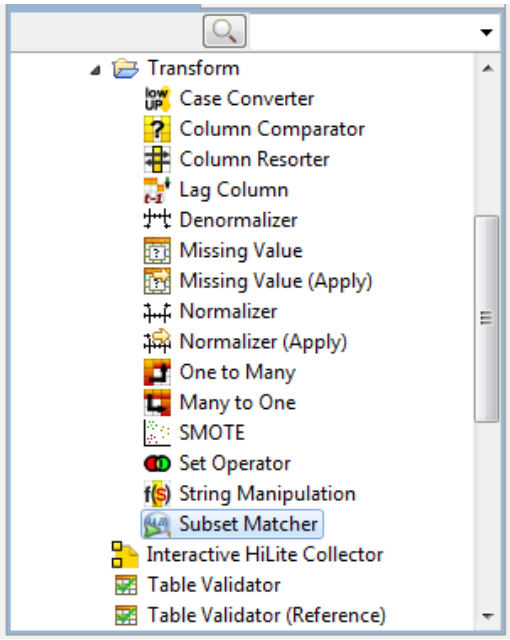


Fig. 3.27 Transforma Columna

En este agrupado es donde se encuentran nodos que se encargan de ejecutar tareas que ayudan al tratamiento de valores perdidos, la cuales que hemos clasificado como **limpieza de datos** y transformación de los mismos.

Transform (Column) también refiere al cálculo de nuevos campos o columnas, basados en las ya existentes, con la finalidad de mejorar el análisis. Ver Figura 3.27.

Case Converter tiene como objetivo convertir cadenas de caracteres a letras mayúsculas y/o a letras minúsculas. Hace la conversión para una o más columnas.

Entrada: Tabla con campos de tipo carácter **Salida:** Tabla con los valores transformados.

Column Comparator compara cada una de las celdas de dos columnas con base en un operador lógico (<,>,=,<=,>=,!=) y concatena la columna con el resultado de dicha comparación. Se puede personalizar la forma de la marca resultante.

Entrada: Tabla arbitraria **Salida:** Tabla más la columna resultante.

Contexto de uso: En ocasiones resulta de interés agregar campos compuestos que contribuyan mejor a los objetivos del procesamiento.

Column Resorter ordena las columnas de una tabla.

Entrada: Tabla por reordenarse **Salida:** Tabla con las columnas reordenadas.

Contexto de uso: puede ser útil ordenar las columnas cuando son muchas.

Lag Column puede copiar una o varias columnas. La nueva columna asigna los valores con n posiciones desfasadas. Es decir, deja vacíos los primeros n valores de la nueva columna y los consecuentes (a partir de la posición n+1) son copiados comenzando desde el primer registro de la columna original (omitiendo los últimos n valores).

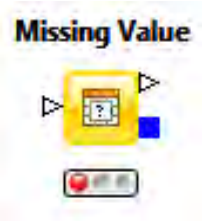
Entrada: Tabla con la columna sujeta a copiarse. **Salida:** Tabla con las columnas extras.

3.2.2.6 Tratamiento de valores perdidos

Un problema relevante con los datos es la presencia de valores perdidos originados por diferentes causas como mal funcionamiento en el sistema, preguntas no contestadas en encuestas, entre otras. Su tratamiento depende de su nivel de aleatoriedad y relevancia ya que está involucrada la interdependencia entre características y la afectación al modelo de minería de datos consecuente [6].

En términos generales, el tratamiento de valores perdidos puede ser clasificado en tres tipos:

- **Ignorar o descartar datos**
  Consiste en descartar las instancias (filas) cuyo valor esté perdido en una o varias características. Así mismo, la eliminación de características (columnas) con una cantidad grande de valores perdidos. Cabe mencionar que antes de descartar características es necesario evaluar el nivel de relevancia en el análisis.
- **Estimación de parámetros**
  Asumir al Data Set como una matriz de n variables aleatorias (columnas) por m valores tomados (filas) y estimar los parámetros de máxima verosimilitud. La propiedad característica de este tipo son los fundamentos estadísticos y los supuestos que se asumen [7].
- **Imputación o sustitución**
  El método de sustitución más común consiste en obtener la media de los valores no vacíos y rellenar con ese valor, o bien, el uso de la moda para las variables nominales. Existen técnicas más sofisticadas con algoritmos clasificadores como son; la media por segmento LDA (Linear Discriminant Analysis), KNN (K Nearest Neighbor), C4.5 (Decision Tree Classifier), regresión lineal, entre otros.



En KNIME existe el nodo ***Missing Value*** declarado como el que ofrece estrategias para el tratamiento de valores perdidos. Sin embargo, se puede complementar con otros nodos como *Missing Value Column Filter* que ya se ha descrito arriba.

Además de que KNIME tiene implementados los algoritmos de clasificación que se utilizan para la imputación sofisticada.

El nodo ***Missing Value*** contiene las siguientes técnicas de tratamiento de valores perdidos:

- Media
- Media por segmentos*
- Máximo
- Mínimo
- Valor asignado por el usuario
- Mediana
- Moda
- Valor anterior no perdido*
- Valor siguiente no perdido*
- Elimina fila*
- Interpolación lineal*

Cada uno de los cuales se puede configurar de manera independiente por campo, o bien, tiene la opción de aplicar una regla predeterminada para todos.

Las técnicas sin la marca * pueden guardarse bajo el estándar PMML. En ese sentido, ***Missing Value (Apply)*** es el nodo encargado de leer y aplicar el modelo PMML generado por *Missing Value* y algunos otros modelos foráneos.

Normalizer: Normalización, estandarización o tipificación es la transformación de una variable numérica a otra del mismo tipo pero con valores que permiten su comparación con otras igualmente transformadas. KNIME contiene tres técnicas de estandarización:

Min-Max Normalization, z-Score Normalization (Gaussian) y Normalization by Decimal Scaling

Entrada: Tabla con las columnas a normalizar **Salida:** Tabla normalizada y Modelo de normalización

Contexto de uso: Resulta de interés comparar valores columnas con escalas diferentes por lo que se estandarizan.

EL modelo de normalización que generado por el nodo *Normalizer* puede ser leído y aplicado por ***Normalizer (Apply).***

En el mismo sentido existe otro nodo muy relacionado con *Normalizer* que es el ***Denormalizer*** cuya tarea es regresar a la escala original aplicando la transformación inversa al modelo de normalización.

Entrada: Tabla normalizada y Modelo de normalización **Salida:** Tabla con escala original

One to Many transforma todos los posibles valores de una columna en tantas columnas como posibles hay valores, marcando con un 1 si la tupla contiene el valor y 0 en caso contrario.

Entrada: Tabla **Salida:** Tabla con las columnas adicionales

Many to One ejecuta la operación inversa del nodo *One to Many*, con la variante que las columnas que serán contenidas en una, pueden tener otros criterios de inclusión como el máximo, mínimo o una expresión regular.

Entrada: Tabla **Salida:** Tabla con la columna adicional

Set Operator realiza operaciones intersección, unión, complemento, complemento de la intersección entre dos columnas de diferentes tablas. La operación se realiza sobre los conjuntos de posibles valores de las columnas. Es decir, viéndolas como variables aleatorias, los conjuntos serían los espacios muestrales de cada una.

Entrada: Dos conjuntos de datos **Salida:** Resultado de la operación

Contexo de uso: Es una alternativa al uso del join cuando no son necesarios de inmediato los demás campos de ambas tablas.

String Manipulation guarda una variedad de transformaciones disponibles para la manipulación de campos de tipo carácter. Integra rutinas como las ejecutadas por ***String Replacer***, ***String To Number*** y ***Case Converter*** además de otras propias como:

- ***Strip****:* Elimina los espacios del principio y fin de una cadena de caracteres.
- ***Substr:*** Extrae una subcadena de caracteres de la cadena original.
- ***Length:*** Devuelve el largo de la cadena de caracteres.
- ***Count:*** Cuenta el número veces que se repite un carácter específico o una expresión regular.
- ***IndexOf:*** Devuelve la posición de en la que se encuentra un carácter específico y donde comienza una expresión regular.
- ***Join:*** Concatena cadenas de caracteres en una sola cadena.

- ***Remove:*** Que elimina de la cadena espacios, subcadenas, signos diacríticos, etc.

Input: Tabla **Output:** Tabla con la transformación en una columna adicional o con remplazo.

Contexto de uso: Muy variado dependiendo de los fines que persiga el usuario.

Table Validator valida los metadatos de una tabla como son el tipo de dato, nombre de la columna, dominio de los atributos.

Entrada: Tabla por validar **Salida:** Primer puerto la tabla correctamente validada si un hubo errores, en caso contrario en el segundo puerto errores de la validación.

3.2.2.7 Filtro Fila

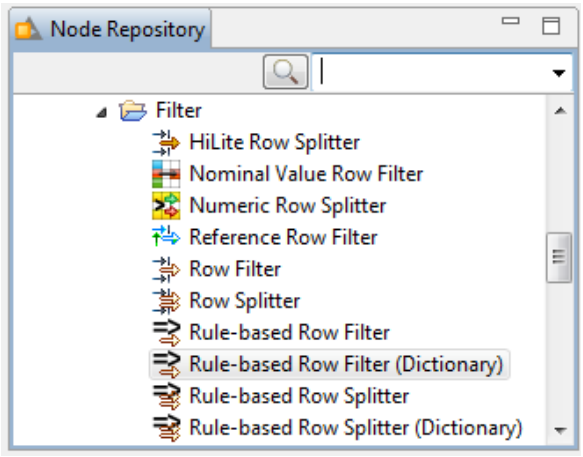


Fig. 3.28 Nodos Filtro Fila

La **reducción de datos** también tiene como objetivo reducir las **instancias**, registros o filas de un conjunto de datos [4]. Es una tarea muy conocida que casi cualquier software de manipulación tiene incluida. Ver Figura 3.28.

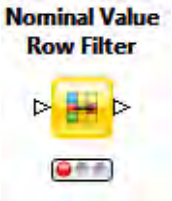
En la ruta **Data Manipulation-> Row-> Filter** se encuentran rutinas básicas de este tipo.

Los nodos que contienen la expresión Spliter generan dos tablas; una con las filas que cumplen la condición de filtro y el resto que no.

HitLite Row Splitter separa en dos tablas un conjunto de datos que esté previamente marcado o seleccionado (***Hilited*** en términos de configuración KNIME) por algunas de sus filas. Separa las filas que están marcadas de las que no lo están.

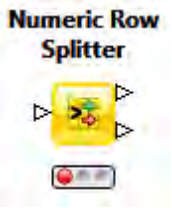
Entrada: Tabla con filas marcadas **Salida:** Primer puerto contiene las filas marcadas y segundo puerto contiene las filas sin marcar.

Contexto de uso: Es de utilidad cuando algunas filas han sido configuradas como **Hilited** y se desea separarlas.



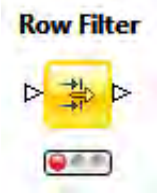
Nominal Value Row Filter filtra una tabla con base en los valores nominales de una columna de tipo carácter. Se configuran los valores cuyas filas se quedarán en la tabla filtrada.

Entrada: Tabla **Salida:** Tabla con filas filtradas



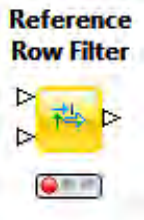
Numeric Row Splitter separa un conjunto de datos en dos tablas. Está basado en los posibles valores numéricos. Se configura un rango para un campo numérico que contendrá el primer conjunto de filas, el resto conforma la segunda tabla.

Entrada: Tabla **Salida:** La tabla en Input dividida en dos



Row Filter filtra una tabla basado en diferentes criterios. Pueden ser definidos por los valores de atributos (numéricos o de tipo carácter), número de las filas y ID´s de las filas de la tabla.

Entrada: Tabla **Salida:** Tabla con filas filtradas



Reference Row Filter filtra los registros de una tabla basada en los valores pertenecientes a otra tabla. Se configura definiendo la columna referencia y la columna criterio para el filtro. Se incluyen o excluyen las filas que cumplen con la coincidencia.

Entrada: El primer puerto es la tabla por filtrar y el segundo es la tabla referencia
Salida: Tabla con filas filtradas

Rule-based Row Filter filtra las filas de una tabla con base en el resultado de una operación lógica como <,<=, =,>,>=, AND, IN, LIKE, MATCHES, OR, MISSING.

Entrada: Tabla **Salida:** Tabla con filas filtradas

Contexto de uso: Su contexto de uso depende de los fines que persiga el usuario.

3.2.2.8 Transforma Filas

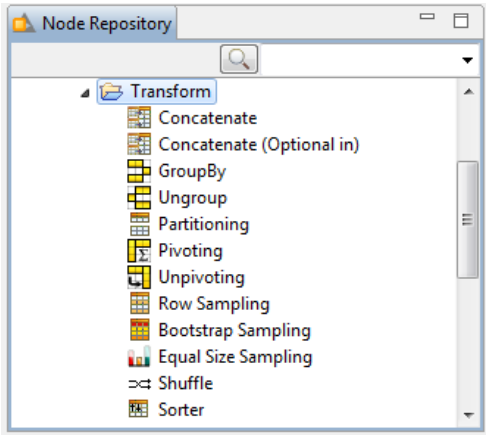


Fig. 3.29 Nodos Transforma Filas

KNIME incluye técnicas de muestreo para reducir las instancias de un conjunto de datos. Así como estrategias de agregación o agrupado, ordenamiento de instancias, concatenación y partición.

Se ubican en el agrupado ***Transform (Rows)*** del repositorio de nodos. Ver Figura 3.29.

Este nodo concatena las filas de dos tablas. En su configuración se pueden excluir o marcar las ID filas duplicadas. Para el trato de las columnas de cada tabla es posible parametrizar la unión o intersección.

Entrada: Dos tablas un número de columnas con el mismo nombre **Salida:** Tabla concatenada

Existe una versión extendida del nodo *Concatenate* llamado ***Concatenate (Optional in)*** que concatena de la misma forma cuatro tablas, es decir, tiene cuatro puertos de entrada y uno de salida.

GroupBy agrupa las filas por los posibles valores que puede tomar una o varias columnas. Parte importante de la configuración de este nodo es seleccionar un **método de agregación** que se aplicará a cada grupo, así como las columnas asociadas al mismo.

En el cuadro de diálogo se incluye una pestaña que contiene una descripción detallada de los diferentes métodos de agregación. Entre los más comunes están:

Count, First, Last, Maximun, Minimun, Missing value count, Mode, Mean, Median, Sum, Kurtosis, Variance, Unique count.

Entrada: Tabla desagregada **Salida:** Tabla agregada

Contexto de uso: Sirven para ver los datos de manera resumida y fácil de comprender.

Pivoting genera una estructura multidimensional que agrega una o más variables. Contiene la misma lista de métodos de agregación que el nodo *GroupBy*. La diferencia radica en la visualización del resultado.

Entrada: Tabla desagregada **Salida:** En el primer puerto de salida se ve la tabla agregada multidimensionalmente. En el segundo la agregación está solo por las **columnas agrupadoras.** En el tercero se ven los totales de las **columnas pivote**.

3.2.2.9 Muestreo

El muestreo es técnica estadística que tiene como objetivo obtener un subconjunto representativo de una población total. Algunos autores lo dividen en dos tipos; probabilístico y no probabilístico [8]. En la manipulación de datos es considerada como técnica de **reducción de instancias** y KNIME incluye nodos que permiten su ejecución.

Row Sampling toma como subconjunto (o muestra) un número absoluto de filas o una proporción del total de observaciones. Dentro de la configuración está la selección de modo de muestreo;

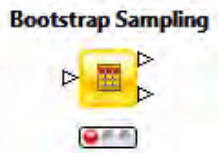
- ***Take from top****:* Toma la muestra de las primeras filas de la tabla.
- ***Linear sampling****:* Selecciona las instancias de manera **uniforme**, siempre incluyendo las filas primera y última.
- ***Draw Randomly:*** Muestreo **aleatorio simple sin reemplazo** donde todas las instancias tienen la misma probabilidad de ser seleccionadas. Opcionalmente se puede definir una semilla de valores aleatorios.
- ***Stratifed Sampling:*** Muestreo **aleatorio estratificado** donde la población está formada grupos homogéneos. Este modo necesita la definición de la variable que segmenta a la población. Opcionalmente se puede definir una semilla de valores aleatorios.

Entrada: Tabla con población total **Salida:** Tabla reducida después del muestreo

Contexto de uso: Manipulando grandes volúmenes de datos es usual caer en la necesidad de reducir los registros alterando lo menos posible su caracterización.

El nodo ***Partitioning*** contiene las mismas técnicas de muestreo que *Row Sampling* con la única diferencia de que el primero llama primera partición a la muestra definida en la configuración del nodo y segunda partición a las filas restantes, obteniendo así dos tablas en el puerto de salida.

Entrada: Tabla con población total **Salida:** Primera partición y segunda partición



Bootstrap Sampling utiliza la técnica de ***Bootstrap*** para llevar a cabo un **muestreo simple con reemplazo** (puede repetir instancias). Con la opción de concatenar una columna con las veces que se repitió cada instancia, así como su ID original.

Entrada: Tabla sujeta a muestreo **Salida:** Primer puerto arroja la tabla con la muestra y segunda tabla arroja las instancias no utilizadas.

Shuffle intercala las filas de una tabla de forma aleatoria. El único parámetro que recibe es una semilla aleatoria.

Entrada: Tabla **Salida:** Tabla reordenada

Sorter tiene la función de ordenar una tabla según el usuario determine en sus parámetros de configuración.

Entrada: Tabla **Salida:** Tabla reordenada

Transpose: transpone una tabla como si fuera una matriz de n columnas por m filas.

Entrada: Tabla **Salida:** Tabla transpuesta

3.2.3 Carga

La carga de los datos pre-procesados se produce al ejecutar exitosamente el nodo de manipulación (semáforo verde), devolviendo a través de su puerto de salida el conjunto de datos transformados (parámetro de entrada del nodo de minería de datos).

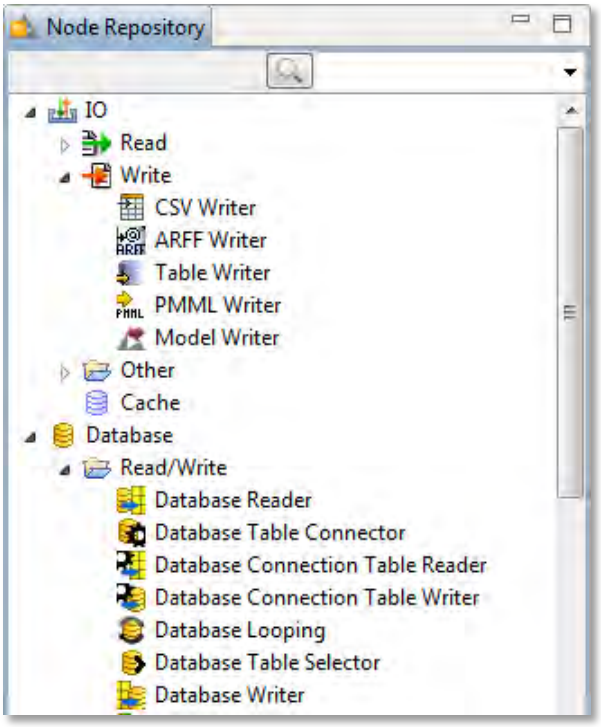


Fig. 3.30 Nodos Carga

3.2.3.1 IO (Write)

Es posible cargar datos trasformados en determinada base de datos, o guardarlos en otro formato.

Los nodos que llevan a cabo esta tarea son distinguidos con el sufijo ***Writer*,** por ejemplo, ***Database Connection Table Writer, Database Writer, CSV Writer, Table Writer***, etc.

Dichos nodos se encuentran en el grupo **IO-> Write** y algunos más en **Database**.

CSV Writer puede cargar un conjunto de datos que esté en formato KNIME a una ruta especificada y cambiarlo al formato según las especificaciones de la configuración. Es posible personalizar parámetros como separador de columnas, si se incluyen las columnas y con qué nombre, etc.

Entrada: Una tabla en formato KNIME que será cargada en una ruta distinta.

Database Writer tiene la característica cargar información a una base de datos creada en un SMBD. En el nodo se configuran los parámetros de conexión a la base donde va a cargar el conjunto de datos u opcionalmente puede recibirlos de otro nodo que ya esté configurado.

Entrada: Tabla en formato KNIME que será cargada a una base de datos.

3.3 Repositorios Públicos

Dada la tendencia Big Data diversas instituciones académicas, gubernamentales, comerciales privadas y de investigación comparten conjuntos de datos (*Data Sets*) a través de sus sitios web de manera pública[39]. Algunos ejemplos de repositorios públicos son los siguientes:

UCI Machine Learning Repository[40] es uno de los repositorios más citados en artículos de ciencias computacionales albergando en la actualidad 335 Data Sets con diversas temáticas. Fue fundado en 1987 por un estudiante en EEUU y actualmente es soportado por *National Science Foundation* y *Rexa.info*.

Datos Abiertos CDMX[41] es un repositorio de datos que el gobierno de la ciudad de México fundó con la finalidad de llevar acabo un *hackathon* dedicado a programadores. Alberga 626 Data Sets aproximadamente, con datos provenientes de diferentes dependencias gubernamentales.

3.3.1 Data Sets

En esta sección se describirán algunos data sets relevantes.

3.3.1.1 Abalone

Descripción: Muestra de conchas de mar con diferentes medidas físicas, las cuales son variables explicativas para la predicción de la edad. Incluye también el sexo de la concha.

[39] *KD Nuggets,* Repositorios Públicos. Enero 2016. [En Línea]. Disponible: http://www.kdnuggets.com/2011/02/free-public-datasets.html

[40] UCI, *Machine Learning Repository*. Repositorios Públicos [En línea] Disponible: http://archive.ics.uci.edu/ml/index.html

[41] Datos Abiertos CDMX. Repositorio Público. Enero 2016. [En línea] Disponible: http://www.datosabiertos.df.gob.mx/sigdata/index.php/Publicacion/index

Atributos: Sexo (nominal), largo (continuo), diámetro (continuo), altura (continuo), peso total (continuo), peso sin concha (continuo), peso viseras (continuo), peso de concha (continuo), anillos (entero).

Instancias: 4177

Fuente: Marine Resources Division, Marine Research Laboratories - Taroona, Department of Primary Industry and Fisheries - Tasmania.

3.3.1.2 Adult

Descripción: Muestra de adultos con datos sociodemográficos y económicos, las cuales son variables explicativas para la predicción de sus ingresos.

Atributos: edad (continuo), clase trabajadora (nominal), fnlwgt (continuo), educación (nominal), educación (continuo), estado civil (nominal), ocupación (nominal), raza (nominal), sexo (nominal), ganancia capital (continuo), perdida capital (continuo), horas por semana (continuo), país de origen (nominal).

Instancias: 48842

Fuente: Ronny Kohavi and Barry Becker - Silicon Graphics

3.3.1.3 BD Transporte de la Ciudad de México

Descripción: Localización de estaciones y paradas, rutas, horarios y frecuencia de servicio del sistema de transporte del Distrito Federal en formato GTFS. Incluye información del Sistema de Transporte Colectivo - Metro, Sistema Metrobús, Servicio de Transportes Eléctricos, Red de Transporte de Pasajeros y Ferrocarriles Suburbanos.

Este conjunto de datos se ofrece a los ciudadanos con la finalidad de que puedan obtener información oficial y confiable, de los distintos sistemas de transporte público del DF, en un formato con estándares internacionales.

Tablas: agency, calendar, calendar_dates, feed_info, frequencies, routes, shapes, stop_times, stops, transfers, trips.

Fuente: Secretaría de Movilidad (Ciudad de México).

Con la finalidad simplificar el análisis de nuestra fuente de información, seleccionaremos como Data Set la **Base de Datos Abiertos de Transporte de la Ciudad de México** ya que su descripción de es intuitiva en su mayoría para el lector. A la vez, puede sembrar el interés de explotar los datos que el gobierno pone a disposición de los ciudadanos.

4. Ejemplo de uso

4.1 Introducción

La Ciudad de México es una de las ciudades más grandes y caóticas del mundo. El aumento de autos privados circulando diariamente en la ciudad provoca embotellamientos continuamente, los cuales afectan la movilidad efectiva de los ciudadanos en la capital. Optar por el transporte público contribuye a la reducción de la población de autos en circulación. Una adecuada planeación y gestión de un sistema de transporte garantiza altos niveles de satisfacción de los usuarios.

Actualmente en la ciudad de México el sistema es administrado por la Secretaria de Movilidad de la Ciudad de México (SEMOVI), organismo creado en 1994 que relevó a la Coordinación General de Transporte y absorbió a otras unidades administrativas[42].

4.2 Objetivo

Estudiar el sistema de transporte de la CDMX siguiendo los principios de la del proceso de minería de datos. Se pretende obtener la mayor información posible de los datos que se recolectan día a día, y no sólo eso, sino que también que dicha información resulte útil para los ciudadanos.

La gente que se desplaza en la capital debería estar interesada en aspectos como las alternativas que existen en el transporte colectivo. Así mismo, ubicar los puntos de parada cercanos a los lugares que frecuenta, trayecto y tiempo de traslado. Las preguntas primordiales de un usuario del transporte público interesado en optimizar su desplazamiento en la ciudad podrían ser las siguientes:

1. ¿Además del metro, que otras formas de transporte son administradas por SEMOVI, así como sus rutas, días de servicio y horarios?
2. ¿Cuáles son las rutas que tardan menos tiempo en hacer su recorrido?

[42] Secretaría de Movilidad (SEMOVI) – CDMX. Febrero 2016 [En línea] Disponible: http://www.semovi.df.gob.mx/

3. ¿Cuáles son las rutas que abarcan mayor territorio de la ciudad?
4. ¿Cuál es la distribución de las frecuencias de arribo de los puntos de parada por ruta por tipo de transporte?
5. Conocer los detalles de las rutas cuyas paradas se encuentran en determinada zona geográfica (punto de partida y destino, distancia recorrida, tiempo del recorrido, horario, días de servicio, agencia).
6. ¿Cuál es la segmentación de zonas geográficas con base en cobertura?

4.3 Comprensión de los datos

El conocimiento o comprensión del problema y enfocar los esfuerzos a darle solución al mismo es uno de los pilares de la preparación de datos, por lo que conocer a detalle las particularidades de la base SEMOVI es en lo primero que hay que prestar atención.

Se infiere el esquema relacional de la base SEMOVI para conocer su estructura; entidades, relaciones y atributos, permitirá definir alcances y limitaciones de la misma. Ver Figura 4.2.

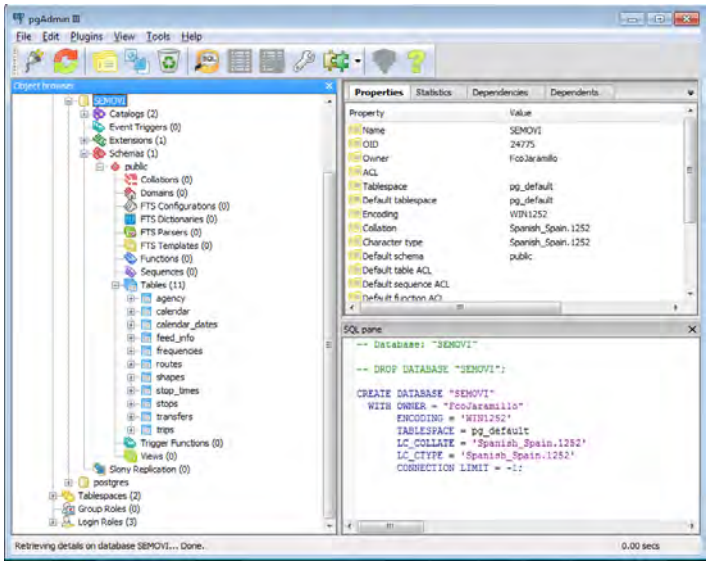


Fig. 4.1 Base SEMOVI en PostgreSQL

Dada la compatibilidad de KNIME con el SMBD **PostgreSQL**, la base SEMOVI será creada con este último con sus respectivos roles y contraseñas. Ver Figura 4.1.

Nota: Cambiar el tipo de codificación de la base de datos de UTF8 a WIN1252 para que las letras "ñ" de la base puedan ser leídas por PostgreSQL.

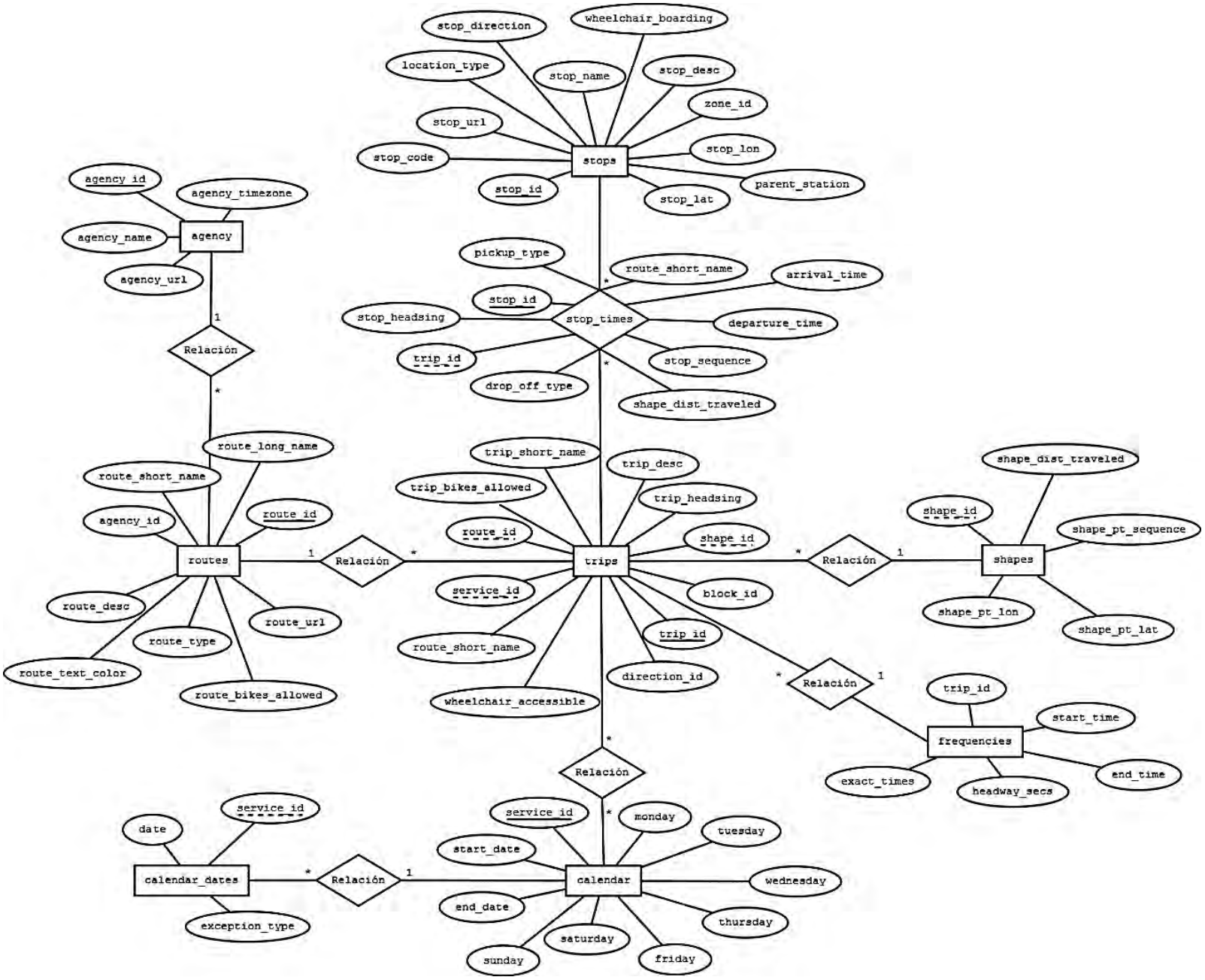


Fig. 4.2 Diagrama ER de Base SEMOVI

Nota: Las relaciones con la leyenda "relación" carecen de tablas en el diseño que inferimos de la base de datos. Sin embargo, el diagrama ilustra su cardinalidad.

A continuación se muestran las tablas creadas y sus atributos (Tablas 4.1 – 4.9):

Tabla 4.1 agency

Tabla	Descripción	Atributos	N° de Instancias
Agency	Guarda la lista de las agencias reguladas por SEMOVI	agency_id; Llave primaria que identifica de forma única a la agencia. agency_name; Nombre de la agencia. agency_url; Página web de la agencia. agency_timezone; Zona horaria. agency_lang; Campo vacío. agency_phone; Teléfono de contacto.	7

Tabla 4.2 routes

Tabla	Descripción	Atributos	N° de Instancias
routes	Almacena el total de rutas con la descripción del trayecto y la agencia a la que pertenecen	agency_id; Llave foránea route_short_name; Código alfanumérico corto que identifica a la ruta, sin embargo existen duplicados. route_long_name; Nombre de la ruta que contiene el lugar de partida y destino. route_desc; Descripciones particulares del trayecto de la ruta (contiene vacíos y el valor RUTA TRAZADA repetidas veces). route_type; Campo categórico de tipo numérico (su dominio solo contiene 5 valores). route_url; Contiene la dirección web del mapa de la ruta (solo 15 tuplas son distintos del vacío en este campo). route_color; Código alfanumérico que	135

		identifica el color. route_text_color; Campo con más del 90% vacío. route_bikes_allowed; Campo con un solo valor (“0”). route_id; Llave primaria que identifica de manera única a las rutas.	

Tabla 4.3 Trips

Tabla	Descripción	Atributos	N° de Instancias
trips	Contiene los posibles viajes de cada ruta existente y una breve descripción	route_id; Llave foránea service_id; Llave foránea trip_short_name; Campo vacío. trip_desc; Descripción del viaje que contiene el lugar de partida y destino. trip_headsign; Destino del viaje. route_short_name; Mismo código de tabla routes direction_id; Campo vacío. block_id; Campo vacío. shape_id; Llave foránea que identifica trayectos diferentes. wheelchair_accessible; Variable dicotómica. trip_bikes_allowed; Campo con un solo valor (“0”). trip_id; Llave primaria que identifica de manera única a los viajes.	1675

Tabla 4.4 Stops

Tabla	Descripción	Atributos	N° de Instancias
stops	Alberga todas las posibles paradas de las rutas existentes	stop_id; Llave primaria que identifica de manera única a las paradas. stop_code; Campo vacío. stop_name; Nombre de la parada. stop_desc; Detalles de la ubicación de la parada. stop_lat; Latitud geográfica de la parada. stop_lon; Longitud geográfica de la parada. zone_id; Campo vacío. stop_url; Campo vacío. location_type; Variable dicotómica. parent_station; Variable con más el 90% vacío (copia valores de stop_name). wheelchair_boarding; Variable discreta. stop_direction; Campo vacío.	5751

Tabla 4.5 stop_times

Tabla	Descripción	Atributos	N° de Instancias
stop_times	Guarda el tiempo entre paradas de los viajes existentes*	trip_id; Llave foránea que identifica a los viajes. stop_sequence; Variable ordinal que determina la secuencia de las paradas. stop_id; Llave foránea que identifica las paradas. arrival_time; tiempo transcurrido desde de primera parada hasta la parada en cuestión. departure_time; tiempo transcurrido desde de primera parada hasta la parada en cuestión. stop_headsign; Campo vacío. route_short_name; Campo vacío. pickup_type; Campo con un solo valor (“0”). drop_off_type; Campo con un solo valor (“0”). shape_dist_traveled; Campo vacío.	68603

Tabla 4.6 shapes

Tabla	Descripción	Atributos	N° de Instancias
shapes	Guarda las coordenadas de los diferentes trayectos	shape_id; Llave primaria que identifica los diferentes trayectos. shape_pt_sequence; variable ordinal que determina la secuencia de las paradas del trayecto. shape_dist_traveled; Campo vacío. shape_pt_lat; Latitud geográfica de la parada del trayecto. shape_pt_lon; Longitud geografica de la parada del trayecto.	27339

Tabla 4.7 calendar_dates

Tabla	Descripción	Atributos	N° de Instancias
calendar_dates	Guarda la relación entre los días feriados y esquema de servicio	service_id; Llave foránea que identifica al esquema de días de servicio. date; Fecha excepción de los días de servicio definidos. exception_type; Variable discreta categórica.	36

Tabla 4.8 calendar

Tabla	Descripción	Atributos	N° de Instancias
calendar	Contiene los esquemas de días de servicio	service_id; Llave primaria que identifica de forma única a la tupla. monday; Determina si hay servicio o no en ese día para el service_id asociado. tuesday; Determina si hay servicio o no en ese día para el service_id asociado. wednesday; Determina si hay servicio o no en ese día para el service_id asociado. thursday; Determina si hay servicio o no en ese día para el service_id asociado. friday; Determina si hay servicio o no en ese día para el service_id asociado. saturday; Determina si hay servicio o no en ese día para el service_id asociado. sunday; Determina si hay servicio o no en ese día para el service_id asociado. start_date; Campo con un solo valor (“20151201”). end_date; Campo con un solo valor (“20160301”).	54

Tabla 4.9 frequencies

Tabla	Descripción	Atributos	N° de Instancias
frequencies	Contiene los horarios de cada viaje	trip_id; Llave foránea que identifica al viaje. start_time; Hora de comienzo del horario. end_time; Hora de término del horario. headway_secs; Variable categórica de tipo numérico. exact_times; Variable con un solo valor ("0").	1675

*** Nota:** quince **paradas** con diferente nomenclatura no se encuentran en *stop_times* y si en *stops*.

Cabe mencionar que se dejaron fuera del diagrama a las tablas:

- **feed_info:** Tabla de un registro que contiene la dirección web de SEMOVI.
- **transfers:** (49 registros y 3 atributos). No es claro su contenido. Aparentemente relaciona las rutas consecutivas. Solo contiene rutas asociadas al metro. Sin embargo, no coincide con tabla stop_times.

4.4 Procesamiento

En la literatura de minería de datos se menciona que el procesamiento de los datos es muy variado de acuerdo a la particularidad del problema. Para el caso de la base de datos SEMOVI, tenemos los datos en un esquema relacional, por lo que las tareas de integración juegan un papel muy importante.

1. **¿Además del metro, que otras formas de transporte son administradas por SEMOVI, así como sus rutas, días de servicio y horarios?**

Como se mencionó arriba, en la base existe una tabla llamada *agency* la cual contiene información general de las agencias que trabajan con SEMOVI, dicha tabla la podemos ver directamente de PostgreSQL. Ver figura 4.3.

	agency_id text	agency_name text	agency_url text	agency_timezone text	agency_lang text	agency_phone text
1	CC	Corredores Concesionados	http://www.semovi.df.gob.mx/	America/Mexico_City		5658-1111
2	MB	Metrobús	http://www.metrobus.df.gob.mx/	America/Mexico_City		5761-6858 Ext. 121 y 136
3	METRO	Sistema de Transporte Colectivo Metro	http://www.metro.df.gob.mx/	America/Mexico_City		5709-1133 Ext. 5051 y 5009
4	NCC	NOCHEBÚS Corredores concesionados	http://www.semovi.df.gob.mx/	America/Mexico_City		5658-1111
5	RTP	Red de Transporte de Pasajeros	http://www.rtp.gob.mx/	America/Mexico_City		5566-5891
6	STE	Servicio de Transportes Eléctricos	http://www.ste.df.gob.mx/	America/Mexico_City		5539-2800
7	SUB	Ferrocarriles Suburbanos	http://www.fsuburbanos.com/	America/Mexico_City		1946-0790

Fig. 4.3 Tabla "agency"

Inmediatamente se puede ver en la Fig. 4.3 cuales son las agencias administradas y reguladas por SEMOVI además del metro.

La siguiente tabla a utilizar es *routes*. Se contará por su atributo llave (*route_id*) para obtener el número de rutas existentes por agencia.

Llevamos a cabo la conexión a la base SEMOVI; Escogemos el nodo **Database Table Connector** con la finalidad de crear varias tablas que completen la respuesta sin tener que realizar más de una conexión a la BD.

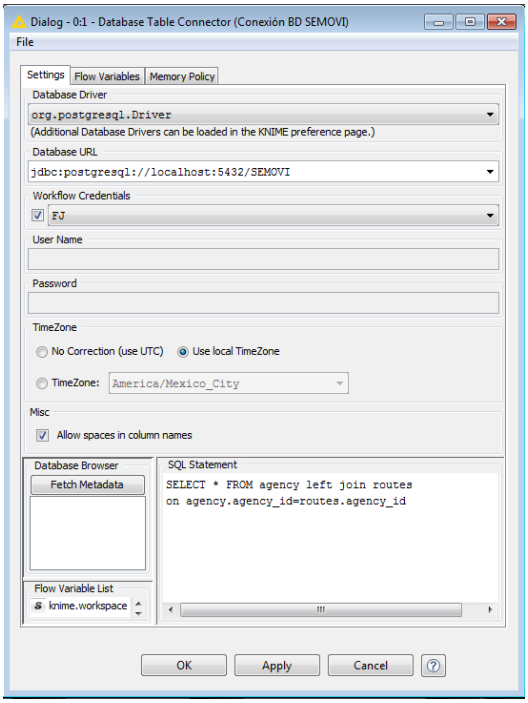


Fig. 4.4 Conexión a SEMOVI

Se ingresan los parámetros correspondientes conforme a los visto en la capítulo 2 como se muestra en la Figura 4.4.

Dado que estamos usando el nodo **Database Table Connector**, debemos escribir una selección inicial con lenguaje SQL:

SELECT * FROM agency left join routes on agency.agency_id=routes.agency_id

Nos permitirá obtener una distribución de las rutas por agencia.

Una vez lograda la conexión y la selección de la unión de las tablas *agency* y *routes* se cargan los datos al formato KNIME mediante la ejecución del nodo **Database Connection Table Reader**.

Ya teniendo los datos en formato KNIME (**Database Connection Table Reader**) se enlazan y configuran los nodos que forman la primera parte del *workflow*; filtra la agencia METRO, cuentan las rutas por agencia, eliminan columnas no relevantes, cambia el nombre del campo aumentado y se carga la tabla resultante en un reporte, como se aprecia en la Fig. 4.5.

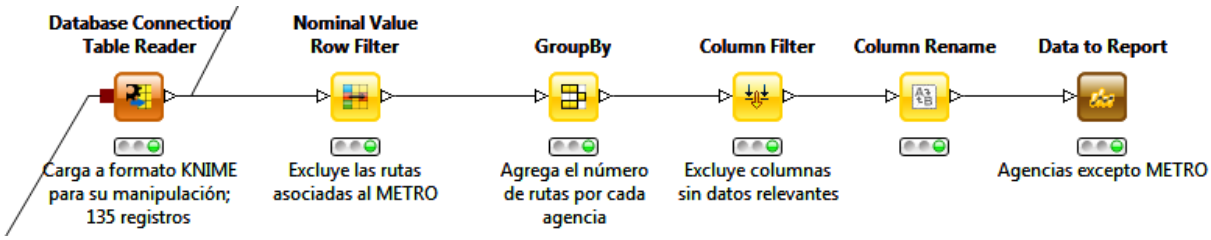


Fig. 4.5 Rutas por Agencia

Cabe mencionar que el reporte final se puede generar de distintas formas según los intereses del usuario. Por ejemplo el nodo ***CSV Writer*** permite guardarlo en la PC en formato CSV a diferencia del nodo ***Data To Report*** que además lo puede convertir a formato imagen PNG y SVG.

La tabla resultante que contiene los datos generales de las agencias distintas al metro y el número de rutas operando se puede ver en la Figura 4.6.

Table View - 0:7 - Data to Report (Agencias excepto METRO)

File Hilite Navigation View Output

Row ID	agency_id	agency_name	agency_url	agency_phone	# Rutas en Operación
Row0	CC	Corredores Concesionados	http://www.semovi.df.gob.mx/	5658-1111	3
Row1	MB	Metrobús	http://www.metrobus.df.gob.mx/	5761-6858 Ext. 121 y 136	5
Row2	NCC	NOCHEBÚS Corredores concesionados	http://www.semovi.df.gob.mx/	5658-1111	2
Row3	RTP	Red de Transporte de Pasajeros	http://www.rtp.gob.mx/	5566-5891	103
Row4	STE	Servicio de Transportes Eléctricos	http://www.ste.df.gob.mx/	5539-2800	9
Row5	SUB	Ferrocarriles Suburbanos	http://www.fsuburbanos.com/	1946-0790	1

Fig. 4.6 Salida Rutas por Agencia

Una vez teniendo los datos leídos en formato KNIME se pueden aplicar técnicas de visualización para apoyar el análisis. En el histograma de la Fig. 4.7 se muestra la distribución de las rutas por cada agencia incluyendo el metro:

De todas las formas de trasporte administradas por SEMOVI, la que tiene más rutas operando es la agencia RTP (Red de Transporte de Pasajeros).

Los nodos usados para esta tarea fueron ***Color Manager*** e ***Histogram***.

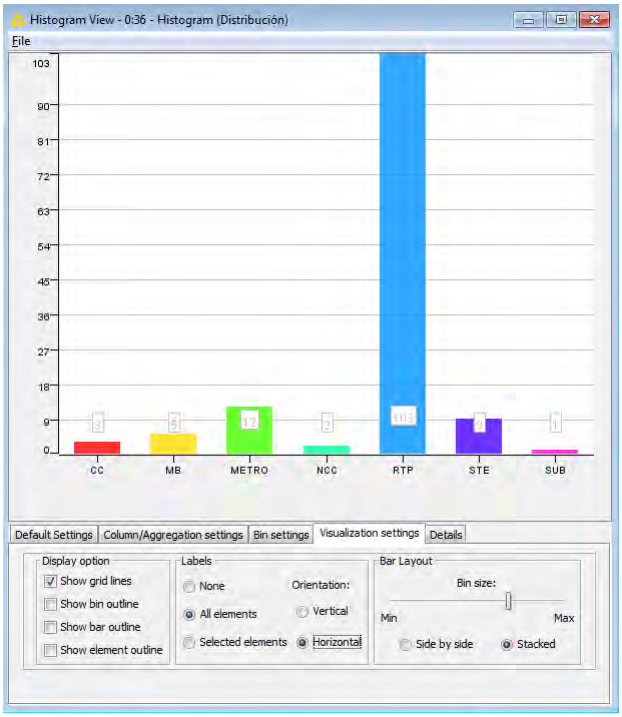


Fig. 4.7 Distribución de Rutas

Lo que sigue es mostrar el detalle de las **rutas incluyendo horarios y días de servicio**. El objetivo es consolidarlo en una sola tabla que muestre los datos de manera clara. Para llegar a ello organizadamente, el resto del workflow se dividirá en **tres flujos** iniciales que procesarán los datos de manera independiente hasta que sea el momento de la consolidación.

Es importante notar que los **horarios y días de servicio están relacionados a los viajes** y no a las rutas directamente. Entonces comenzamos ejecutando la unión de tablas *routes* y *trips* en el primer nodo **Database Query**. Ver Figura 4.8.

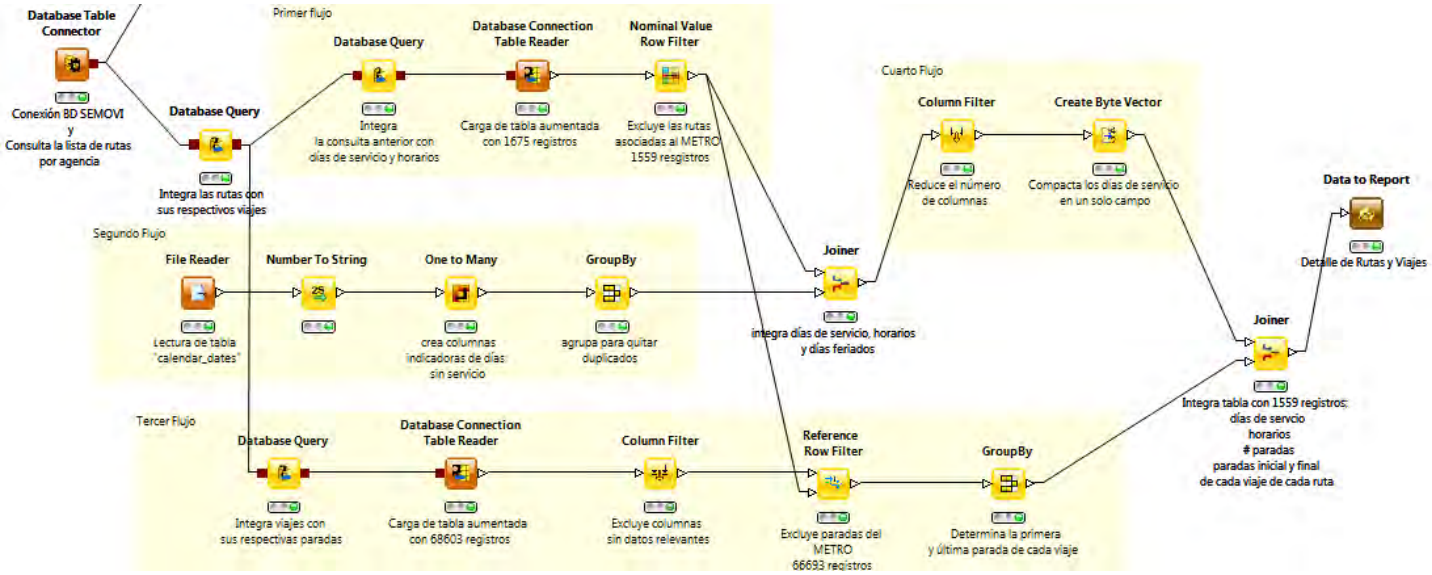


Fig. 4.8 Rutas y Viajes

Primer flujo; se encarga de obtener la lista de viajes con sus días de servicio correspondientes, excluyendo al metro como se muestra en la Figura 4.9.

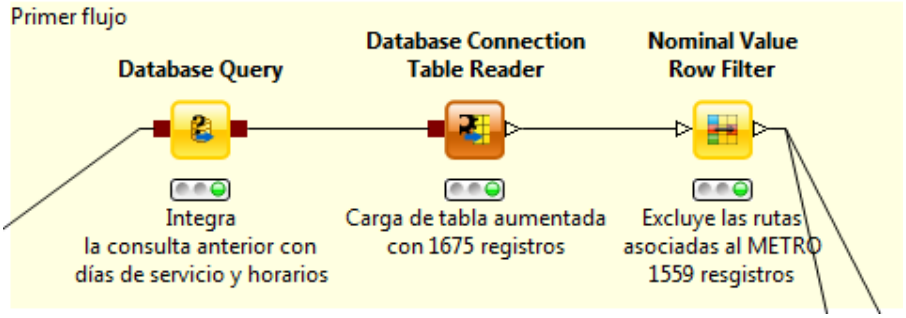


Fig. 4.9 Primer Flujo Rutas y Viajes

Segundo flujo; procesa la tabla *calendar_dates* de tal manera que pueda ceder los días feriados al primer flujo. Ver Figura 4.10.

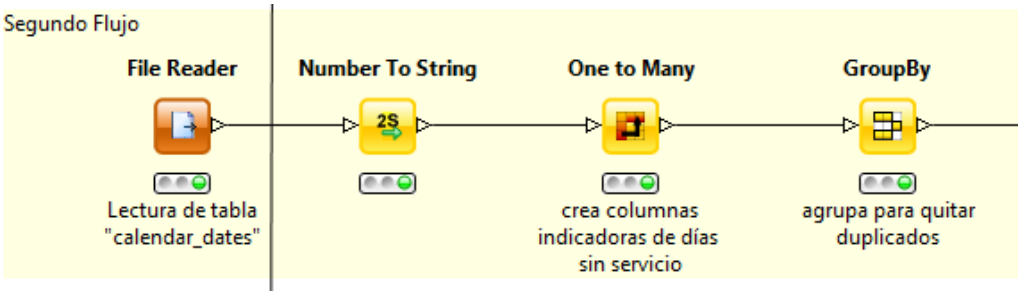


Fig. 4.10 Segundo Flujo Rutas y Viajes

Tercer flujo; obtiene datos agregados por viaje como número de paradas y coordenadas geográficas del punto de partida y destino. También excluye al metro para reducir tiempo de ejecución en la siguiente unión. Ver Figura 4.11.

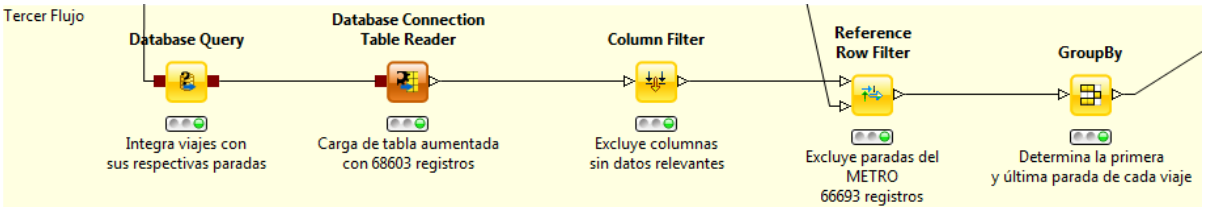


Fig. 4.11 Tercer Flujo Rutas y Viajes

Cuarto flujo; compacta los campos de la tabla proveniente de los flujos 1 y 2 y reduce el número de columnas. Ver Figura 4.12.

Fig. 4.12 Cuarto Flujo Rutas y Viajes

Tabla resultante; contiene los datos relevantes consolidados a nivel viaje y sin atributos vacíos. Ver Tabla 4.10.

Tabla 4.10 Atributos Salida Rutas y Viajes

Atributo	**Descripción**
agency_id	Identificador de las agencias
route_long_name	Nombre largo de la ruta
route_id	Identificador de las rutas
trip_desc	Descripción del viaje
trip_id	Identificador del viaje
start_time	Hora de inicio del horario de servicio
Atributo	**Descripción**

end_time	Hora de término del horario de servicio
Feriado 25 Dic	Variable dicotómica que define si hay servicio o no el día 12-12-2015
Feriado 1 Ene	Variable dicotómica que define si hay servicio o no el día 01-01-2016
days_service	Vector que determina los días de servicio; 1 si hay servicio y 0 lo contrario. El orden de las entradas corresponde a los días de la semana
First(stop_id)	Llave primaria del punto de partida
First(stop_lat)	Latitud geográfica del punto de partida
First(stop_lon)	Longitud geográfica del punto de partida
Last(stop_id)	Llave primaria del punto del destino
Last(stop_lat)	Latitud geográfica del punto del destino
Last(stop_lon)	Longitud geográfica del punto del destino
Count(stop_id)	Número de paradas del viaje
First(stop_name)	Nombre del punto de partida
Last(stop_name)	Nombre del destino

Con la salida descrita arriba se concluye el detalle de las rutas incluyendo horarios y días de servicio. Por lo tanto ha quedado completada la pregunta 1.

2. ¿Cuáles son las rutas que tardan menos tiempo en hacer su recorrido?

Para responder a la pregunta se tomará el campo *arrival_time* de la tabla *stop_times*. Dicha tabla como ya se mencionó, alberga los lapsos entre el punto de inicio y cada parada de los viajes existentes, **incluyendo la última parada**. Por esta vía obtenemos las rutas cuyos viajes les lleva menos tiempo el recorrido.

El primer paso es generar una conexión a la base de datos y la lectura de la tabla *stop_times*.

Dado que las únicas tuplas que nos interesan de la tabla *stop_times* son aquellas que tienen el último valor secuencial, la tabla es ordenada ascendentemente por el campo *sequence* con ayuda del nodo ***Sorter,*** después se seleccionan dichos registros con el nodo ***GroupBy*** y finalmente calculamos el **tiempo de recorrido** con el nodo ***Time Difference***, ubicado en el agrupado ***Time Series*** del repositorio de nodos. Ver Figura 4.13.

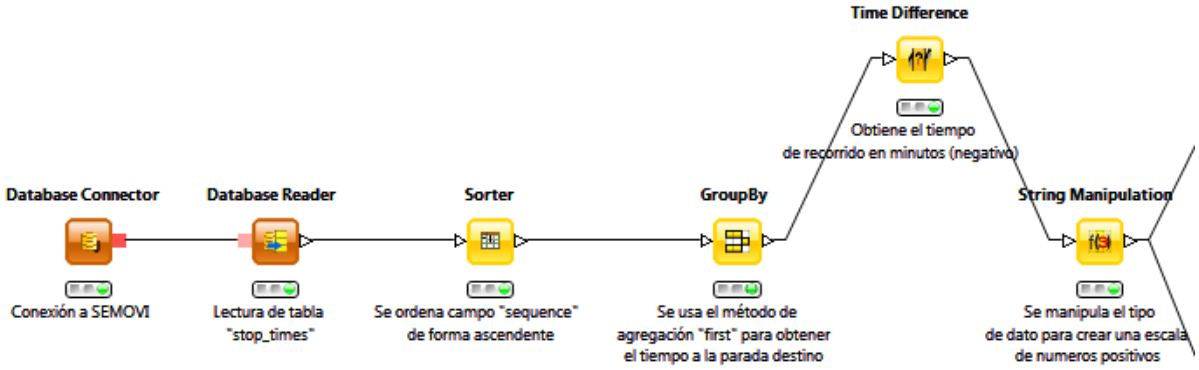


Fig. 4.13 Tiempo Viajes

Vale la pena mostrar la manipulación del tipo de dato realizado con el nodo ***String Manipulations.*** Ver Figura 4.14.

```
toDouble(removeChars(string($time diff$),"-"))
```

Fig. 4.14 Manipulación de Valores

Dentro de la utilería de KNIME también es posible obtener estadísticas básicas de los campos de la base. Para ello se usa el nodo ***Statistics*** contenido en la sección Statistics del repositorio de nodos.

El campo que nos interesa es **Tiempo Recorrido**; Se puede apreciar que el tiempo promedio de los viajes es de 66.65 minutos y que los viajes que menos tardan en hacer su recorrido lo hacen en solo 15.4 minutos según la base SEMOVI en la Figura 4.15.

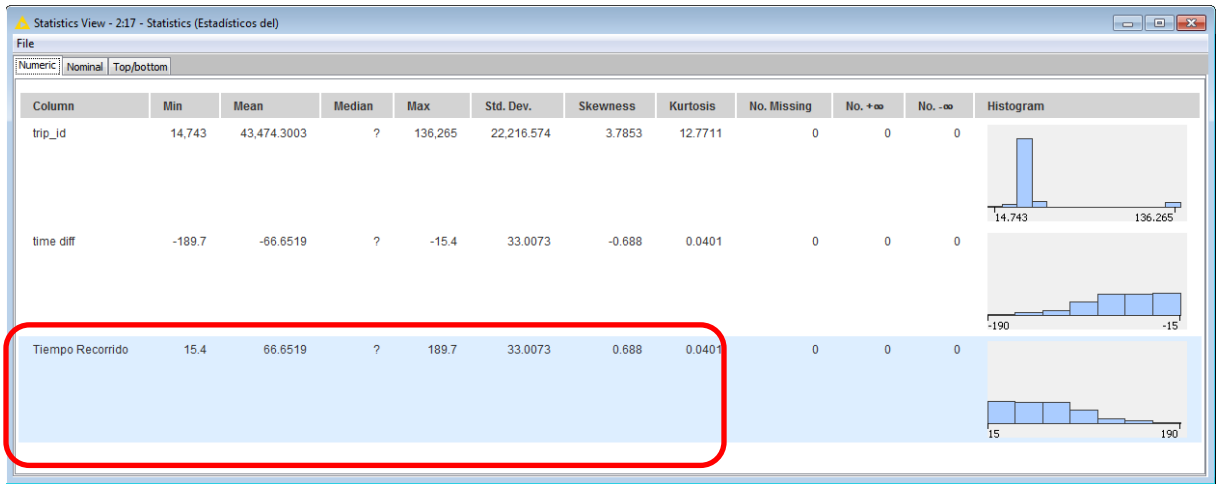


Fig. 4.15 Estadísticos Tabla Frequencies

Para indagar en la distribución de los tiempos de los viajes en relación a las agencias y, además, obtener el detalle de aquellos viajes que tardan menos en realizar su recorrido, aprovechamos la estructura del workflow de la pregunta anterior (Fig. 4.8 Rutas y Viajes):

Modificamos el filtro que excluía al METRO y dejamos todos los posibles valores para el campo agency_id como se muestra en la Figura 4.16.

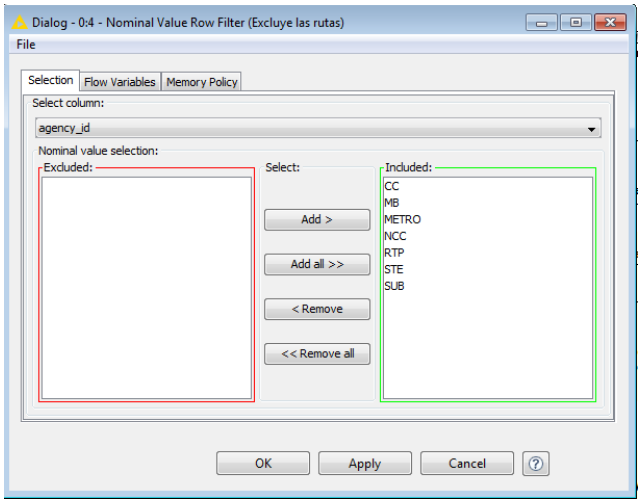


Fig. 4.16 Diálogo Nominal Value Row Filter

Dado que la pregunta 2 no está limitada a algunas agencias, es necesario contar con la información de todos los viajes.

Todo esto con la finalidad de obtener el mismo nivel de detalle de los viajes que hacen menos tiempo (ruta asociada, punto de partida, destino, días feriados, agencia, días de servicio, etc.).

Después de modificar la configuración del nodo ***Nominal Value Row Filter***, se ejecuta nuevamente el *workflow* y la tabla resultante se guarda en formato CSV con la ayuda del nodo ***CSV Writer***.

Hay que mencionar que otra finalidad de usar el *workflow* anterior es ilustrar la forma de trasladar información útil de un proceso a otro.

Una vez guardada la tabla en la PC en formato CSV, utilizamos el nodo ***File Reader*** para su lectura como se puede ver en la Fig 4.17.

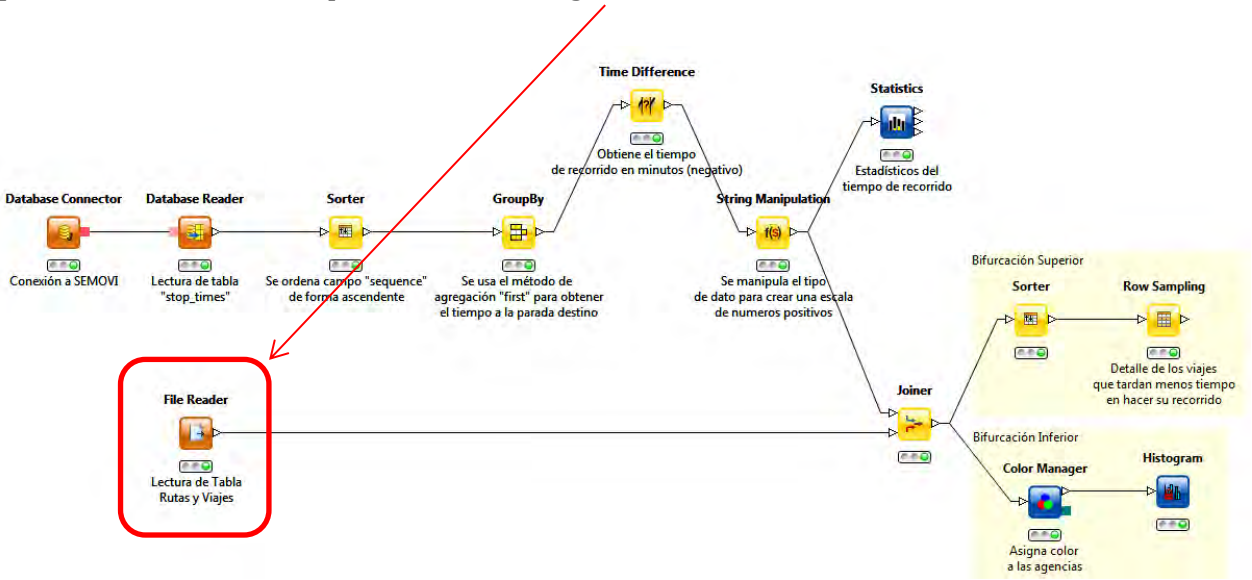


Fig. 4.17 Tiempo Viajes

Joiner se encarga de juntar los datos del tiempo de recorrido de cada viaje con la información anteriormente procesada.

En la **bifurcación inferior** se configuran las agencias con color para poder identificarlas (nodo ***Color Manager***). Ver Figura 4.18.

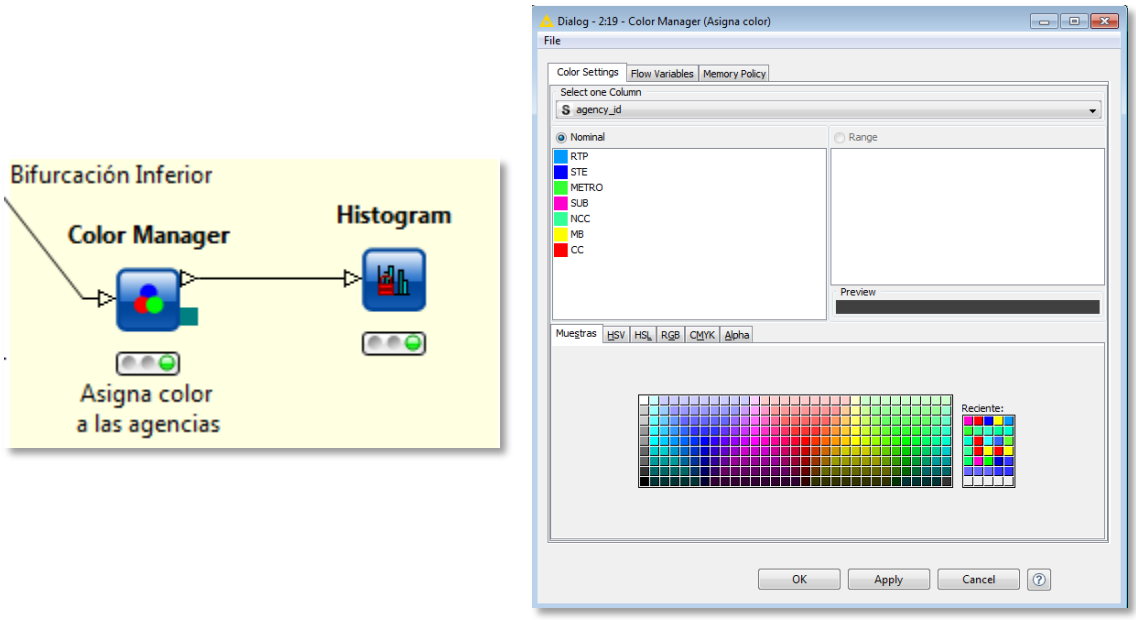


Fig. 4.18 Asigna Color Agencias

Una vez definido el color por agencia, se genera un histograma (nodo ***Histogram***) con la distribución de los tiempos y la ponderación de las agencias en cada *bin.* **Nota**: Los bines del histograma están dados en minutos como se muestra en la Figura 4.19.

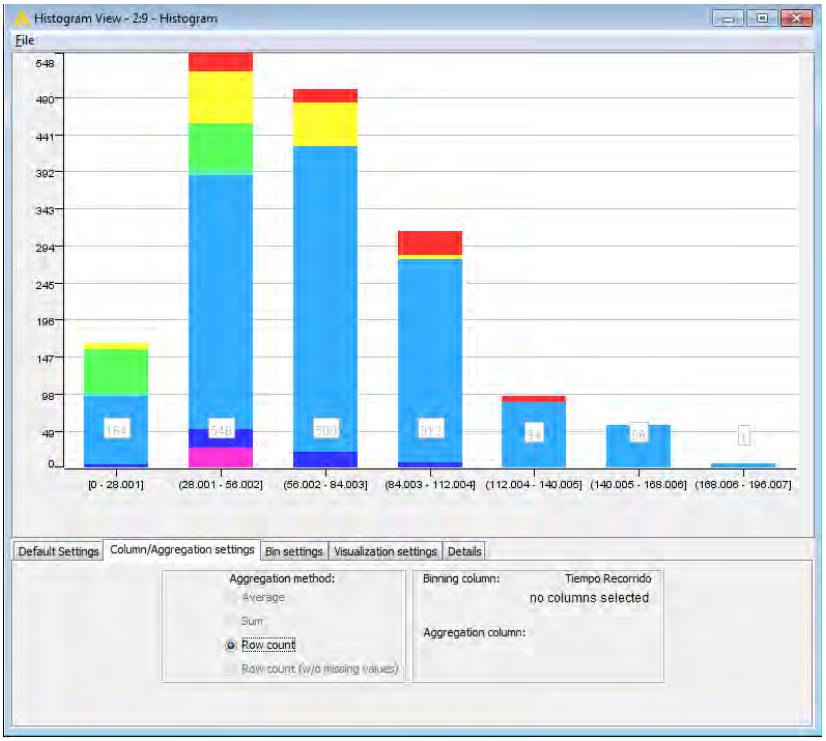


Fig. 4.19 Distribución Tiempo Viajes

En la **bifurcación superior** se ordenan los viajes por el tiempo de recorrido (nodo ***Sorter***) y se extrae la muestra de los primeros 164 viajes más rápidos. Lo anterior con el nodo ***Row Sampling***; previendo así una tabla con del detalle necesario para conocer dichos viajes (ruta asociada, punto de partida, destino, días feriados, agencia, días de servicio). Ver Fig. 4.20.

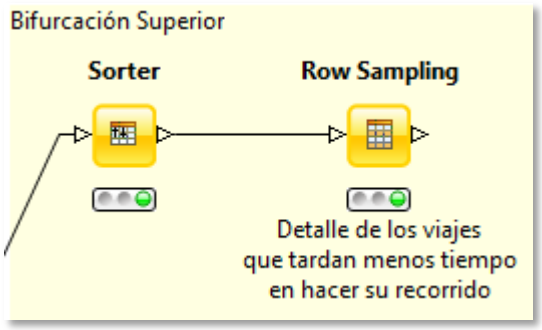


Fig. 4.20 Obtiene Muestra de viajes más rápidos

En Fig. 4.21 se observa que el viaje más rápido es el correspondiente a la ruta del METRO que va de Santa Anita a Martín Carrera cuyo tiempo de recorrido es de 15.4 minutos.

The sampled table - 8:6 - Row Sampling (Detalle de los viajes)

File

Table "default" - Rows: 164 | Spec - Columns: 20 | Properties | Flow Variables

Row ID	trip_id	Tiempo Recorrido	agency_id	route_long_name	route_id	trip_desc
Row6_Row206	14845	15.4	METRO	Santa Anita - Martín Carrera	ROUTE_14246	Martín Carrera - Santa Anita
Row31_Row207	29361	15.4	METRO	Santa Anita - Martín Carrera	ROUTE_14246	Martín Carrera - Santa Anita
Row32_Row208	29362	15.4	METRO	Santa Anita - Martín Carrera	ROUTE_14246	Martín Carrera - Santa Anita
Row5_Row833	14844	15.5	METRO	Santa Anita - Martín Carrera	ROUTE_14246	Santa Anita - Martín Carrera
Row29_Row834	29322	15.5	METRO	Santa Anita - Martín Carrera	ROUTE_14246	Santa Anita - Martín Carrera
Row30_Row835	29337	15.5	METRO	Santa Anita - Martín Carrera	ROUTE_14246	Santa Anita - Martín Carrera
Row1449_Row652	39475	16.6	RTP	Arenal 4ta. Sección - Metro Pantitlán	ROUTE_15334	Arenal 4ta. Sección - Metro Pantitlán
Row1450_Row653	39476	16.6	RTP	Arenal 4ta. Sección - Metro Pantitlán	ROUTE_15334	Arenal 4ta. Sección - Metro Pantitlán
Row1451_Row654	39477	16.6	RTP	Arenal 4ta. Sección - Metro Pantitlán	ROUTE_15334	Arenal 4ta. Sección - Metro Pantitlán
Row1452_Row655	39478	16.6	RTP	Arenal 4ta. Sección - Metro Pantitlán	ROUTE_15334	Arenal 4ta. Sección - Metro Pantitlán
Row1453_Row656	39479	16.6	RTP	Arenal 4ta. Sección - Metro Pantitlán	ROUTE_15334	Arenal 4ta. Sección - Metro Pantitlán
Row1454_Row657	39480	16.6	RTP	Arenal 4ta. Sección - Metro Pantitlán	ROUTE_15334	Arenal 4ta. Sección - Metro Pantitlán
Row1455_Row1202	39481	17.5	RTP	Arenal 4ta. Sección - Metro Pantitlán	ROUTE_15334	Metro Pantitlán - Arenal 4ta. Sección

Fig. 4.21 Viajes Recorrido más Rápido

3. ¿Cuáles son las rutas que abarcan mayor territorio de la ciudad?

Para obtener una distancia estimada de los recorridos de cada viaje se utilizan las tablas *stops* y *stop_times*. Dichas tablas contienen las coordenadas en grados de las paradas y la secuencia de las mismas, así como el viaje al que pertenecen.

Por otro lado, es importante mencionar que SEMOVI rige los servicios de transporte que abarcan principalmente la Ciudad de México, la cual tiene una superficie de aprox. de 1,485 km^2 y la distancia de norte a sur se aproxima a 28.8 km aprox (Largo de la Av. Insurgentes[43] que contiene la ruta completa **Indios Verdes – El caminero** con 46 estaciones). Nos lleva a pensar que las **distancias entre paradas de las rutas** difícilmente excederían los 20 km.

Por lo anterior se utilizará la **distancia euclidiana** para realizar las estimaciones de las distancias. Existen otras formas de aproximación como la **Fórmula de Haversine** que toma en cuenta la curvatura de la tierra, sin embargo, de acuerdo con un artículo publicado en el sitio web *Geographic Information Systems FAQ* [44], es más precisa cuando las distancias exceden los 20 km.

Para poder realizar operaciones como raíz cuadrada y potencias es necesario agregar el nodo ***Math Formula*** al repositorio de nodos KNIME. Se obtiene instalando el plugin ***KNIME Math Expression***. Ver Figura 4.22.

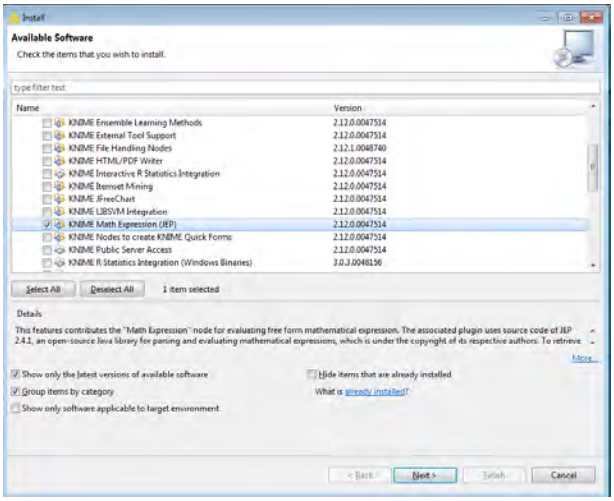


Fig. 4.22 Instalación Nodo Math Formula

Concluida la instalación se almacena en el agrupado ***Misc*** del repositorio de nodos. Ver Fig. 4.23.

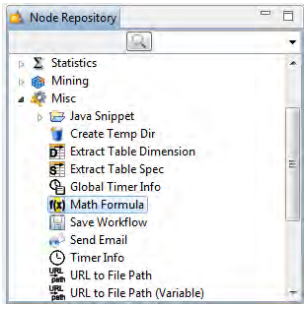


Fig. 4.23 Nodo Math Formula

[43] Secretaría de Transportes y Vialidad (SETRAVI). Mayo 2016 [En línea]. Disponible: http://web.archive.org/web/20140514015723/http://www.setravi.df.gob.mx/wb/stv/avenida_insurgentes

[44] *Great circle distance between 2 points.* Febrero 2016 [En línea]. Disponible: http://www.movable-type.co.uk/scripts/gis-faq-5.1.html

Nuevamente se hace una conexión a nuestra base SEMOVI (*Database Connector*) y se escribe una consulta que permite reunir los atributos de interés para el procesamiento (*Database Reader*). Ver Fig. 4.24.

SELECT *stop_times.trip_id, stop_times.stop_id, stop_times.stop_sequence,*

stops.stop_lat, stops.stop_lon

FROM *stops* ***INNER JOIN*** *stop_times*

ON *stops.stop_id=stop_times.stop_id*

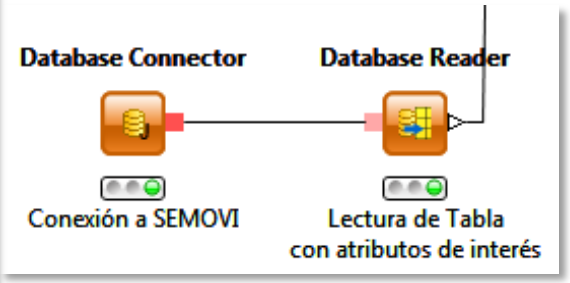


Fig. 4.24 Consulta a Tablas stops y stop_times

El siguiente paso es asegurarnos de que la tabla esté ordenada por viaje y secuencia. Dicho ordenamiento es crítico para el siguiente paso del *workflow*:

La finalidad del ordenamiento es poder concatenar nuevas columnas que contengan las coordenadas de cada parada anterior en el mismo registro. Esto con ayuda del nodo ***Lag Column*** que realizará una copia de los campos *stop_lat* y *stop_long* con un desface de un registro, es decir, pega el primer valor en el segundo registro, el segundo valor en el tercero y así sucesivamente.

De la misma forma el campo *trip_id*. Todo este pre procesamiento sirve para el cálculo de la distancia euclidiana. Ver Fig. 4.25.

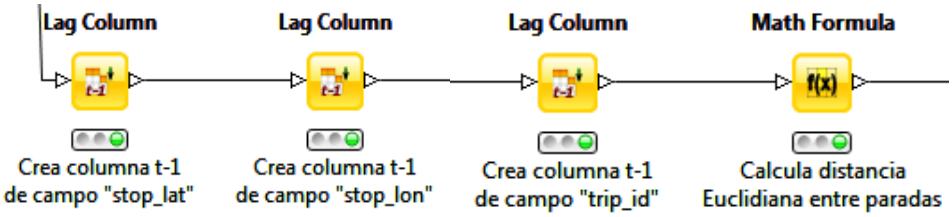


Fig. 4.25 Distancia Viajes Recorrida Parte I

Ahora, el nodo ***Math Formula*** contiene diferentes funciones matemáticas importantes como las trigonométricas, exponenciales, logarítmicas, además de operadores

aritméticos, funciones lógicas y funciones de agregación. Pueden aplicarse a los valores de los atributos de una tabla y generar nuevas columnas.

A continuación, se muestra el cuadro de diálogo de dicho nodo con la configuración que determina la distancia en kilómetros entre las paradas de cada viaje. Ver Figuras 4.26 y 4.27.

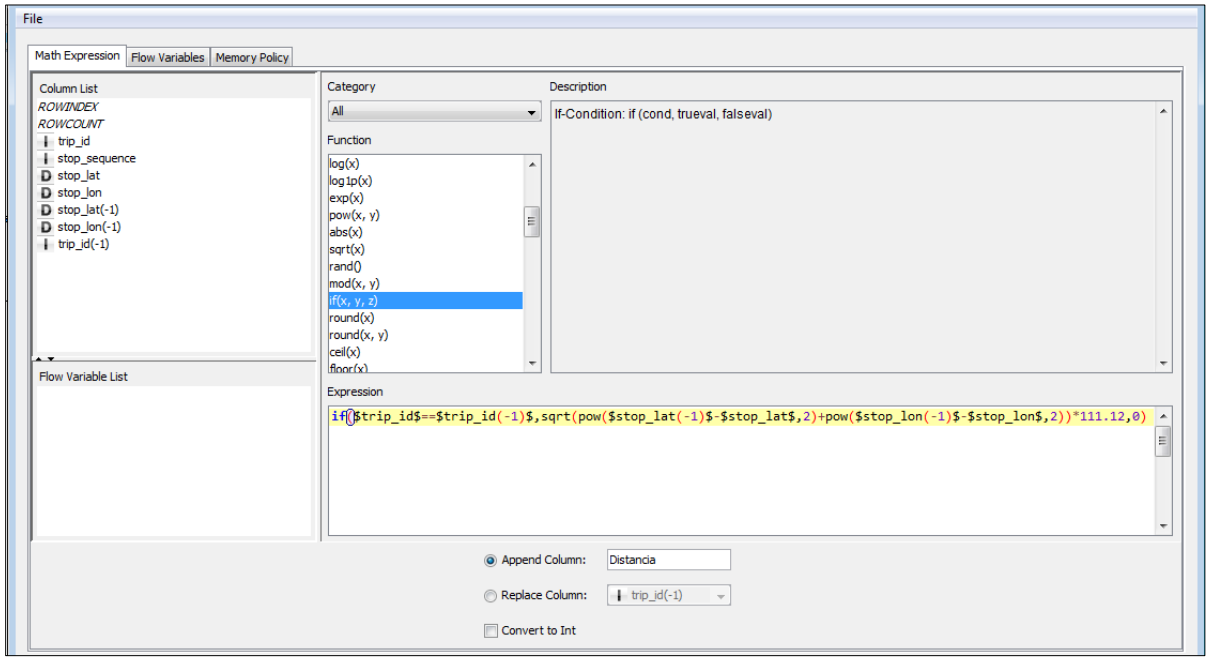


Fig. 4.26 Configuración de Nodo Math Formula

```
if($trip_id$==$trip_id(-1)$,sqrt(pow($stop_lat(-1)$-
$stop_lat$,2)+pow($stop_lon(-1)$ $stop_lon$,2))*111.12,0)
```

Fig. 4.27 Distancia Euclidiana

- La función lógica “if” sirve para saber cuándo terminan las coordenadas asociadas a un viaje y empiezan las del otro para asignar distancia cero.
- Las funciones “sqrt” y “pow” para la raíz cuadrada y la potencia respectivamente, corresponde a la fórmula euclidiana.
- Se multiplica por la constante 111.12 para realizar la conversión aproximada de grados a kilómetros.

Una vez teniendo las distancias entre paradas, se suman las asociadas a cada viaje con el nodo ***GroupBy*** y se calculan estadísticos de la variable sumada Sum(Distancia) con el nodo ***Statistics***:

Tabla 4.11 Estadísticos Campo "Sum(Distancia)"

Media	16.8702 km
Máximo	41.0208 km
Mínimo	4.3087 km
Std. Dev	7.0736 km

Para responder a la pregunta con amplitud concatenamos a los viajes todos los atributos que se generaron en los dos *workflows* anteriores (ruta asociada, punto de partida, punto destino, agencia, días de servicio, **tiempo del recorrido**).

Usamos nuevamente el nodo ***CSV Writer*** para guardar la tabla resultante en la pregunta anterior en la PC. Inmediatamente la volvemos a cargar para usarla en el *workflow* **Distancia Recorrida Viajes** como se ve en la Fig. 4.28:

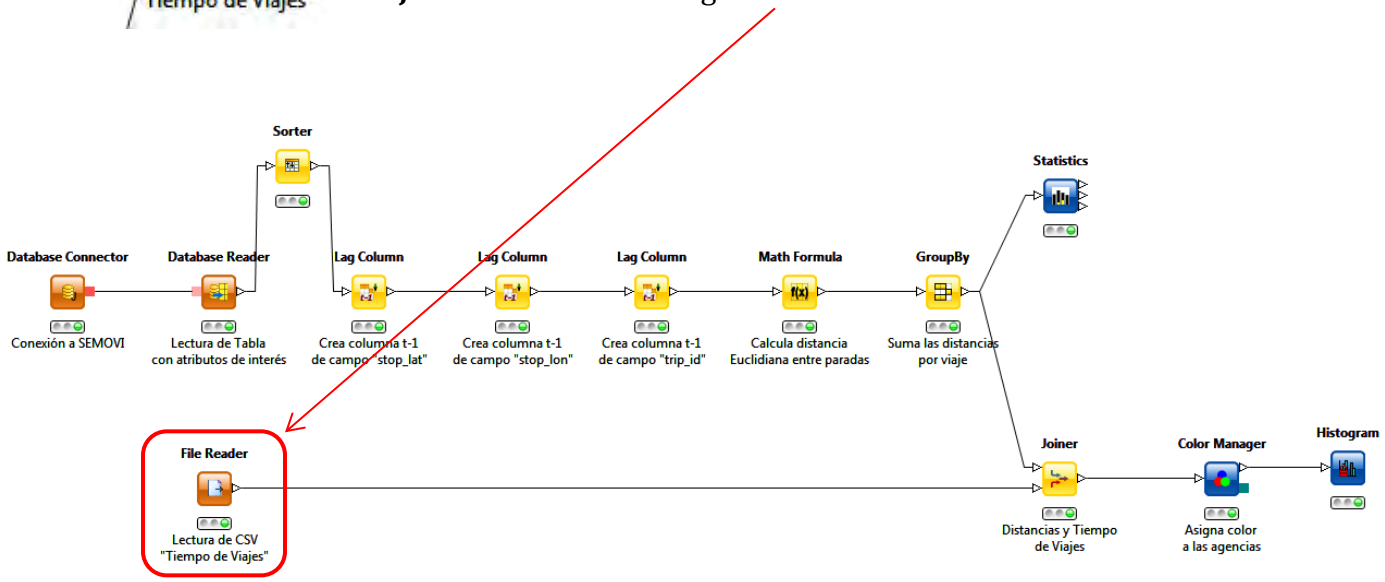
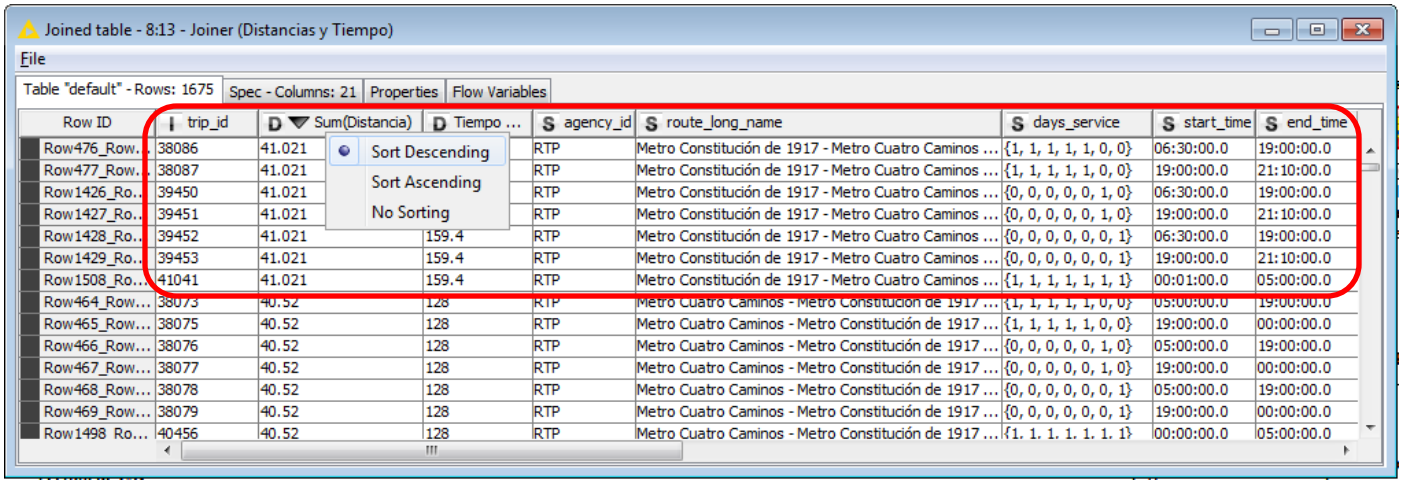


Fig. 4.28 Distancia Recorrida Viajes

Para vizualizar los viajes con el **recorrido más extenso** ordenamos directamente en la tabla resultante del **Joiner** dando clic izquierdo en Sum(Distancia) y seleccionando el modo descendente. Ver Fig. 4.29.

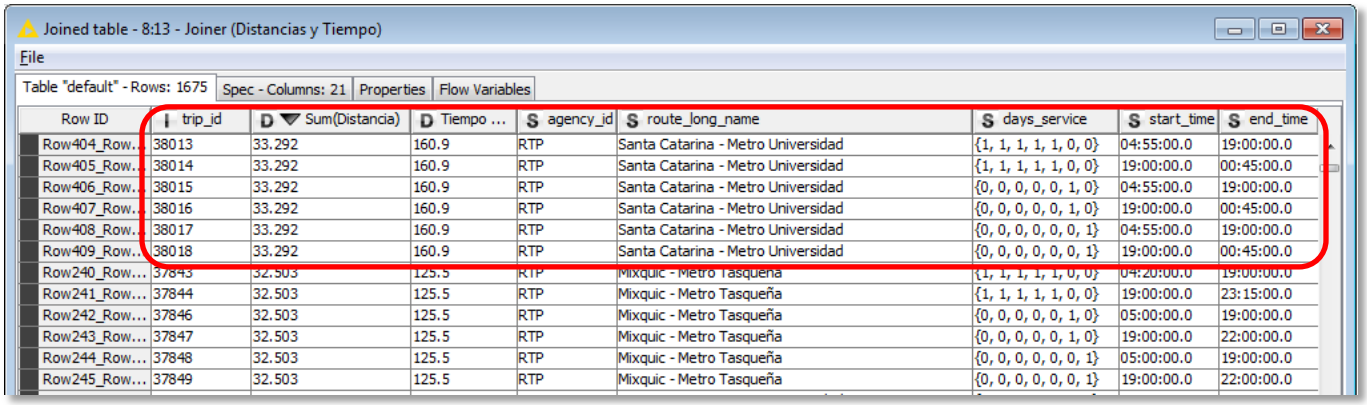
Una vez ordenada la tabla, puede observar que los viajes más extensos corresponden a la agencia RTP cuya ruta va del Metro Cuatro Caminos al Metro Constitución de 1917, la cual tiene viajes de servicio de lunes a domingo en dos horarios diferentes (6:30 am a 7:00 pm y de 7:00 pm a 9:10 pm). Dichos viajes tardan aproximadamente 159.4 minutos en hacer su recorrido. Ver Fig. 4.29.



Row ID	trip_id	Sum(Distancia)	Tiempo ...	agency_id	route_long_name	days_service	start_time	end_time
Row476_Row...	38086	41.021	159.4	RTP	Metro Constitución de 1917 - Metro Cuatro Caminos ...	{1, 1, 1, 1, 1, 0, 0}	06:30:00.0	19:00:00.0
Row477_Row...	38087	41.021	159.4	RTP	Metro Constitución de 1917 - Metro Cuatro Caminos ...	{1, 1, 1, 1, 1, 0, 0}	19:00:00.0	21:10:00.0
Row1426_Ro...	39450	41.021	159.4	RTP	Metro Constitución de 1917 - Metro Cuatro Caminos ...	{0, 0, 0, 0, 0, 1, 0}	06:30:00.0	19:00:00.0
Row1427_Ro...	39451	41.021	159.4	RTP	Metro Constitución de 1917 - Metro Cuatro Caminos ...	{0, 0, 0, 0, 0, 1, 0}	19:00:00.0	21:10:00.0
Row1428_Ro...	39452	41.021	159.4	RTP	Metro Constitución de 1917 - Metro Cuatro Caminos ...	{0, 0, 0, 0, 0, 0, 1}	06:30:00.0	19:00:00.0
Row1429_Ro...	39453	41.021	159.4	RTP	Metro Constitución de 1917 - Metro Cuatro Caminos ...	{0, 0, 0, 0, 0, 0, 1}	19:00:00.0	21:10:00.0
Row1508_Ro...	41041	41.021	159.4	RTP	Metro Constitución de 1917 - Metro Cuatro Caminos ...	{1, 1, 1, 1, 1, 1, 1}	00:01:00.0	05:00:00.0
Row464_Row...	38073	40.52	128	RTP	Metro Cuatro Caminos - Metro Constitución de 1917 ...	{1, 1, 1, 1, 1, 0, 0}	05:00:00.0	19:00:00.0
Row465_Row...	38075	40.52	128	RTP	Metro Cuatro Caminos - Metro Constitución de 1917 ...	{1, 1, 1, 1, 1, 0, 0}	19:00:00.0	00:00:00.0
Row466_Row...	38076	40.52	128	RTP	Metro Cuatro Caminos - Metro Constitución de 1917 ...	{0, 0, 0, 0, 0, 1, 0}	05:00:00.0	19:00:00.0
Row467_Row...	38077	40.52	128	RTP	Metro Cuatro Caminos - Metro Constitución de 1917 ...	{0, 0, 0, 0, 0, 1, 0}	19:00:00.0	00:00:00.0
Row468_Row...	38078	40.52	128	RTP	Metro Cuatro Caminos - Metro Constitución de 1917 ...	{0, 0, 0, 0, 0, 0, 1}	05:00:00.0	19:00:00.0
Row469_Row...	38079	40.52	128	RTP	Metro Cuatro Caminos - Metro Constitución de 1917 ...	{0, 0, 0, 0, 0, 0, 1}	19:00:00.0	00:00:00.0
Row1498_Ro...	40456	40.52	128	RTP	Metro Cuatro Caminos - Metro Constitución de 1917 ...	{1, 1, 1, 1, 1, 1, 1}	00:00:00.0	05:00:00.0

Fig. 4.29 Viajes Más Extensos

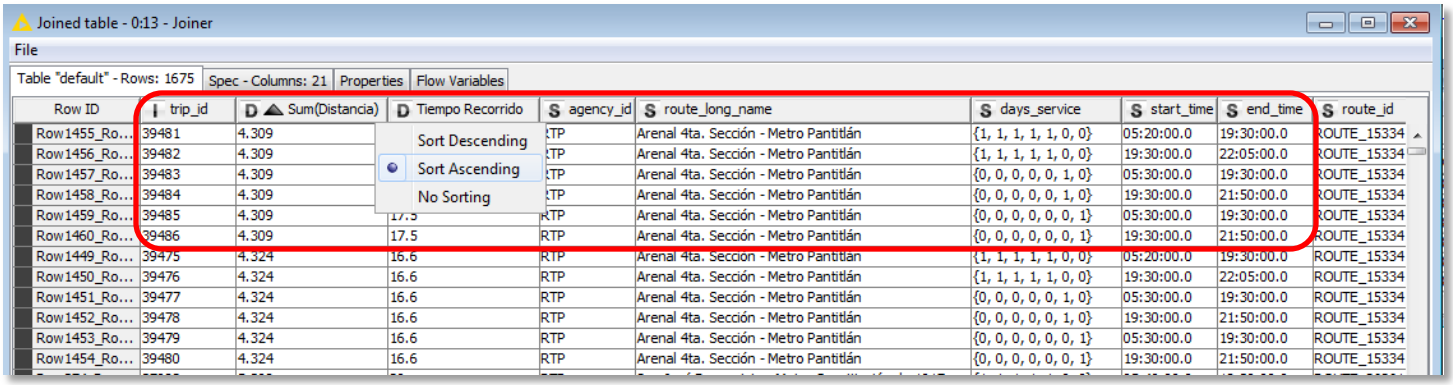
Otra ruta de las más largas es la ruta cuya unidad RTP que va de Santa Catarina a Metro Universidad con 33.292 km de longitud y un tiempo estimado de 160.9 minutos Ver Fig. 4.30.



Row ID	trip_id	Sum(Distancia)	Tiempo ...	agency_id	route_long_name	days_service	start_time	end_time
Row404_Row...	38013	33.292	160.9	RTP	Santa Catarina - Metro Universidad	{1, 1, 1, 1, 1, 0, 0}	04:55:00.0	19:00:00.0
Row405_Row...	38014	33.292	160.9	RTP	Santa Catarina - Metro Universidad	{1, 1, 1, 1, 1, 0, 0}	19:00:00.0	00:45:00.0
Row406_Row...	38015	33.292	160.9	RTP	Santa Catarina - Metro Universidad	{0, 0, 0, 0, 0, 1, 0}	04:55:00.0	19:00:00.0
Row407_Row...	38016	33.292	160.9	RTP	Santa Catarina - Metro Universidad	{0, 0, 0, 0, 0, 1, 0}	19:00:00.0	00:45:00.0
Row408_Row...	38017	33.292	160.9	RTP	Santa Catarina - Metro Universidad	{0, 0, 0, 0, 0, 0, 1}	04:55:00.0	19:00:00.0
Row409_Row...	38018	33.292	160.9	RTP	Santa Catarina - Metro Universidad	{0, 0, 0, 0, 0, 0, 1}	19:00:00.0	00:45:00.0
Row240_Row...	37843	32.503	125.5	RTP	Mixquic - Metro Tasqueña	{1, 1, 1, 1, 1, 0, 0}	04:20:00.0	19:00:00.0
Row241_Row...	37844	32.503	125.5	RTP	Mixquic - Metro Tasqueña	{1, 1, 1, 1, 1, 0, 0}	19:00:00.0	23:15:00.0
Row242_Row...	37846	32.503	125.5	RTP	Mixquic - Metro Tasqueña	{0, 0, 0, 0, 0, 1, 0}	05:00:00.0	19:00:00.0
Row243_Row...	37847	32.503	125.5	RTP	Mixquic - Metro Tasqueña	{0, 0, 0, 0, 0, 1, 0}	19:00:00.0	22:00:00.0
Row244_Row...	37848	32.503	125.5	RTP	Mixquic - Metro Tasqueña	{0, 0, 0, 0, 0, 0, 1}	05:00:00.0	19:00:00.0
Row245_Row...	37849	32.503	125.5	RTP	Mixquic - Metro Tasqueña	{0, 0, 0, 0, 0, 0, 1}	19:00:00.0	22:00:00.0

Fig. 4.30 Viajes Más Extensos 2

Así mismo también están los recorridos más cortos; por ejemplo la ruta que va del Arenal 4ta Sección a Metro Pantitlán recorre 4.309 km aprox., en un tiempo estimado de 17.5 minutos y un servicio de lunes a domingo con dos horarios que varían ligeramente los fines de semana como se puede ver en la Fig. 4.31.



Row ID	trip_id	Sum(Distancia)	Tiempo Recorrido	agency_id	route_long_name	days_service	start_time	end_time	route_id
Row1455_Ro...	39481	4.309		RTP	Arenal 4ta. Sección - Metro Pantitlán	{1, 1, 1, 1, 1, 0, 0}	05:20:00.0	19:30:00.0	ROUTE_15334
Row1456_Ro...	39482	4.309		RTP	Arenal 4ta. Sección - Metro Pantitlán	{1, 1, 1, 1, 1, 0, 0}	19:30:00.0	22:05:00.0	ROUTE_15334
Row1457_Ro...	39483	4.309		RTP	Arenal 4ta. Sección - Metro Pantitlán	{0, 0, 0, 0, 0, 1, 0}	05:30:00.0	19:30:00.0	ROUTE_15334
Row1458_Ro...	39484	4.309		RTP	Arenal 4ta. Sección - Metro Pantitlán	{0, 0, 0, 0, 0, 1, 0}	19:30:00.0	21:50:00.0	ROUTE_15334
Row1459_Ro...	39485	4.309	17.5	RTP	Arenal 4ta. Sección - Metro Pantitlán	{0, 0, 0, 0, 0, 0, 1}	05:30:00.0	19:30:00.0	ROUTE_15334
Row1460_Ro...	39486	4.309	17.5	RTP	Arenal 4ta. Sección - Metro Pantitlán	{0, 0, 0, 0, 0, 0, 1}	19:30:00.0	21:50:00.0	ROUTE_15334
Row1449_Ro...	39475	4.324	16.6	RTP	Arenal 4ta. Sección - Metro Pantitlán	{1, 1, 1, 1, 1, 0, 0}	05:20:00.0	19:30:00.0	ROUTE_15334
Row1450_Ro...	39476	4.324	16.6	RTP	Arenal 4ta. Sección - Metro Pantitlán	{1, 1, 1, 1, 1, 0, 0}	19:30:00.0	22:05:00.0	ROUTE_15334
Row1451_Ro...	39477	4.324	16.6	RTP	Arenal 4ta. Sección - Metro Pantitlán	{0, 0, 0, 0, 0, 1, 0}	05:30:00.0	19:30:00.0	ROUTE_15334
Row1452_Ro...	39478	4.324	16.6	RTP	Arenal 4ta. Sección - Metro Pantitlán	{0, 0, 0, 0, 0, 1, 0}	19:30:00.0	21:50:00.0	ROUTE_15334
Row1453_Ro...	39479	4.324	16.6	RTP	Arenal 4ta. Sección - Metro Pantitlán	{0, 0, 0, 0, 0, 0, 1}	05:30:00.0	19:30:00.0	ROUTE_15334
Row1454_Ro...	39480	4.324	16.6	RTP	Arenal 4ta. Sección - Metro Pantitlán	{0, 0, 0, 0, 0, 0, 1}	19:30:00.0	21:50:00.0	ROUTE_15334

Fig. 4.31 Viajes Más Cortos

4. ¿Cuál es la distribución de las frecuencias de arribo de los puntos de parada por ruta por tipo de transporte?

La tabla **stop_times** guarda los tiempos por viaje de todas las rutas vigentes. Sin embargo, no hay suficientes datos en la base SEMOVI para responder a la pregunta, es necesario contar con el inventario de las unidades.

Con la información que hasta este punto hemos logrado obtener es posible aproximar la frecuencia **por unidad** en circulación de una ruta en específico.

Para ello tomamos arbitrariamente a la ruta **ROUTE_15920** de la base de datos SEMOVI y calculamos el número de veces que la unidad logra hacer su recorrido completo dentro de su horario y días de servicio.

El insumo inicial para esta tarea es la tabla resultante de la pregunta anterior, misma que se deposita en la base de datos SEMOVI con ayuda del nodo ***Database Writer***.

Una vez cargada la tabla ***distancia_de_viajes*** a la base SEMOVI, se configura una conexión a la misma y se selecciona la nueva tabla generada. Luego se filtra solo la ruta 15920 con ***Database Row Filter*** y se ejecuta la lectura de los datos. Ver Fig. 4.32.

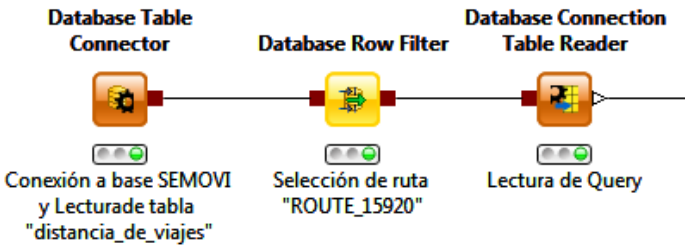


Fig. 4.32 Análisis Ruta 15920

Después, el nodo **Time Difference** calcula el campo ***time diff*** que guarda el **tiempo de servicio en minutos** de los viajes asociados a la ruta **ROUTE_15920**. Ver Fig. 4.33.

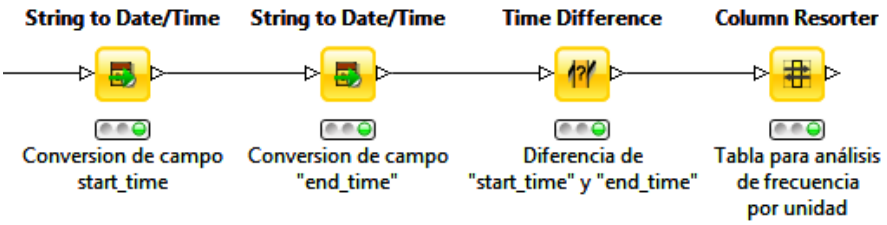
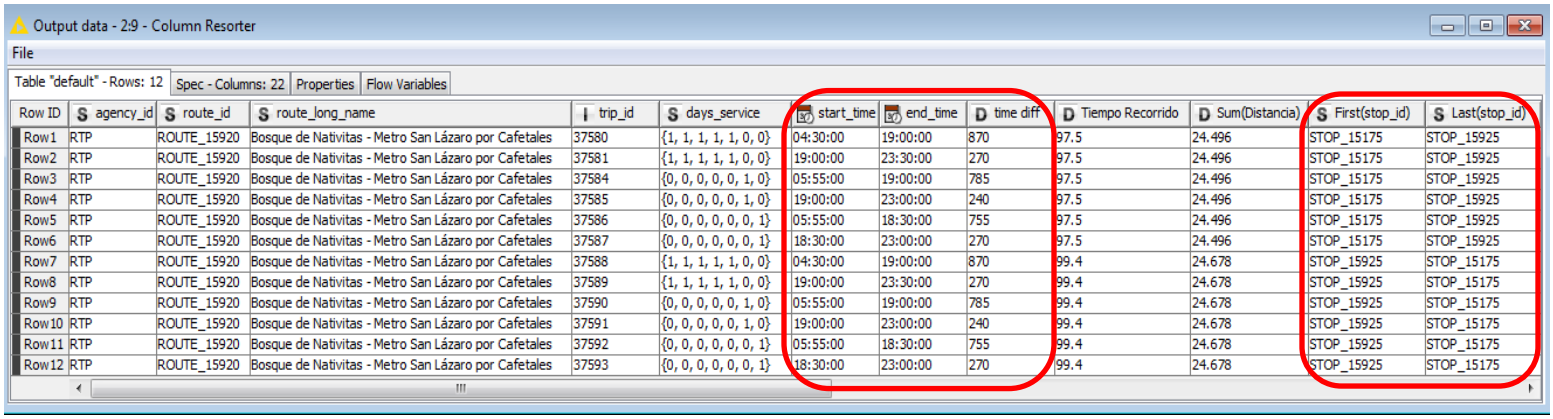


Fig. 4.33 Análisis Ruta 15920 (2)

La tabla resultante contiene los viajes asociados a la ruta 15920 y los campos necesarios para la estimación de la frecuencia por unidad; ***Tiempo Recorrido*** y ***time diff*** (tiempo de servicio).Ver Fig. 4.34.


Output data - 2:9 - Column Resorter

File

Table "default" - Rows: 12 | Spec - Columns: 22 | Properties | Flow Variables

Row ID	agency_id	route_id	route_long_name	trip_id	days_service	start_time	end_time	time diff	Tiempo Recorrido	Sum(Distancia)	First(stop_id)	Last(stop_id)
Row1	RTP	ROUTE_15920	Bosque de Nativitas - Metro San Lázaro por Cafetales	37580	{1, 1, 1, 1, 1, 0, 0}	04:30:00	19:00:00	870	97.5	24.496	STOP_15175	STOP_15925
Row2	RTP	ROUTE_15920	Bosque de Nativitas - Metro San Lázaro por Cafetales	37581	{1, 1, 1, 1, 1, 0, 0}	19:00:00	23:30:00	270	97.5	24.496	STOP_15175	STOP_15925
Row3	RTP	ROUTE_15920	Bosque de Nativitas - Metro San Lázaro por Cafetales	37584	{0, 0, 0, 0, 0, 1, 0}	05:55:00	19:00:00	785	97.5	24.496	STOP_15175	STOP_15925
Row4	RTP	ROUTE_15920	Bosque de Nativitas - Metro San Lázaro por Cafetales	37585	{0, 0, 0, 0, 0, 1, 0}	19:00:00	23:00:00	240	97.5	24.496	STOP_15175	STOP_15925
Row5	RTP	ROUTE_15920	Bosque de Nativitas - Metro San Lázaro por Cafetales	37586	{0, 0, 0, 0, 0, 0, 1}	05:55:00	18:30:00	755	97.5	24.496	STOP_15175	STOP_15925
Row6	RTP	ROUTE_15920	Bosque de Nativitas - Metro San Lázaro por Cafetales	37587	{0, 0, 0, 0, 0, 0, 1}	18:30:00	23:00:00	270	97.5	24.496	STOP_15175	STOP_15925
Row7	RTP	ROUTE_15920	Bosque de Nativitas - Metro San Lázaro por Cafetales	37588	{1, 1, 1, 1, 1, 0, 0}	04:30:00	19:00:00	870	99.4	24.678	STOP_15925	STOP_15175
Row8	RTP	ROUTE_15920	Bosque de Nativitas - Metro San Lázaro por Cafetales	37589	{1, 1, 1, 1, 1, 0, 0}	19:00:00	23:30:00	270	99.4	24.678	STOP_15925	STOP_15175
Row9	RTP	ROUTE_15920	Bosque de Nativitas - Metro San Lázaro por Cafetales	37590	{0, 0, 0, 0, 0, 1, 0}	05:55:00	19:00:00	785	99.4	24.678	STOP_15925	STOP_15175
Row10	RTP	ROUTE_15920	Bosque de Nativitas - Metro San Lázaro por Cafetales	37591	{0, 0, 0, 0, 0, 1, 0}	19:00:00	23:00:00	240	99.4	24.678	STOP_15925	STOP_15175
Row11	RTP	ROUTE_15920	Bosque de Nativitas - Metro San Lázaro por Cafetales	37592	{0, 0, 0, 0, 0, 0, 1}	05:55:00	18:30:00	755	99.4	24.678	STOP_15925	STOP_15175
Row12	RTP	ROUTE_15920	Bosque de Nativitas - Metro San Lázaro por Cafetales	37593	{0, 0, 0, 0, 0, 0, 1}	18:30:00	23:00:00	270	99.4	24.678	STOP_15925	STOP_15175

Fig. 4.34 Atributos Ruta 15920

Arriba en la Fig. 4.34, se aprecia que seis registros corresponden a la ida y otros seis corresponden a la vuelta, ya que el punto destino de los primeros (campo *Last(stop_id)*) son el comienzo del resto (campo *First(stop_id)*).

Por otro lado, cabe destacar que el tiempo de recorrido de los viajes de ida difiere ligeramente contra los viajes de vuelta como se puede ver en la variable *Tiempo Recorrido*.

Otro dato relevante es que el tiempo de recorrido en la base no es variante entre horarios ni días de servicio, lo que nos lleva a pensar que está implícito el supuesto de tránsito habitual en la estimación.

Para responder a la pregunta de la frecuencia **por unidad** en circulación, hay que dividir el **tiempo de servicio** entre el **tiempo de recorrido** (vuelta completa);

Lunes a Viernes

- Tiempo de servicio: Suma de los diferentes horarios al día (*time diff*): 870 + 270 = 1140
- Tiempo de recorrido: Suma de los viajes de ida y vuelta (*Tiempo de recorrido*): 97.5 + 99.4 = 106.9)
- Vueltas al día: 1140 / 106.9 = 10.66, truncado da un total de **10 vueltas al día**.

Sábado

Se lleva a cabo el mismo razonamiento:

- Tiempo de servicio: 785 + 240 = 1025
- Tiempo de recorrido: 97.5 + 99.4 = 106.9
- **Vueltas al día: 9.58 -> 9**

Domingo

Se lleva a cabo el mismo razonamiento:

- Tiempo de servicio: 755 + 220 = 975

- Tiempo de recorrido: 97.5 + 99.4 = 106.9
- **Vueltas al día: 9.2 -> 9**

Cabe mencionar que no todas las rutas de la base se comportan de la misma manera, existen rutas con derivaciones, diferentes horarios y diferentes esquemas de días de servicio.

5. **Los detalles de las rutas cuyas paradas se encuentran en determinada zona geográfica (punto de partida y destino, distancia recorrida, tiempo del recorrido, horario, días de servicio, agencia).**

Existe el atributo **zone_id** en la tabla **stops**, sin embargo, todos sus registros están en vacío. Para responder esta pregunta es necesario segmentar a las paradas pertenecientes a los viajes de las rutas vigentes.

Se utiliza el **algoritmo k-medias** que agrupa a las paradas en segmentos según su cercanía entre ellas tomando como insumo la tabla **stops** de la base SEMOVI. Ver Figuras 4.35 y 4.36.

Fig. 4.35 Workflow k-Means

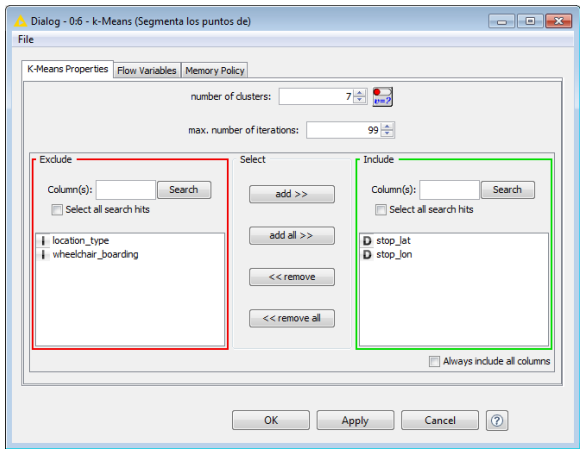


Fig. 4.36 Dialgo Nodo K-Medias

Solamente los atributos stop_lat y stop_lon son incluidos en la ejecución del algoritmo.

El número de clusters se define en 7 persiguiendo una segmentación del tipo:

Norte, Noreste, NorOeste, Este, Oeste, Centro, y Sur.

Adicionalmente se incluyen manualmente las coordenadas de las delegaciones políticas de la CDMX y se integran a la base de datos resultante con los clusters o regiones obtenidos. Los datos de las delegaciones son obtenidos del INEGI. Ver Fig 4.37.

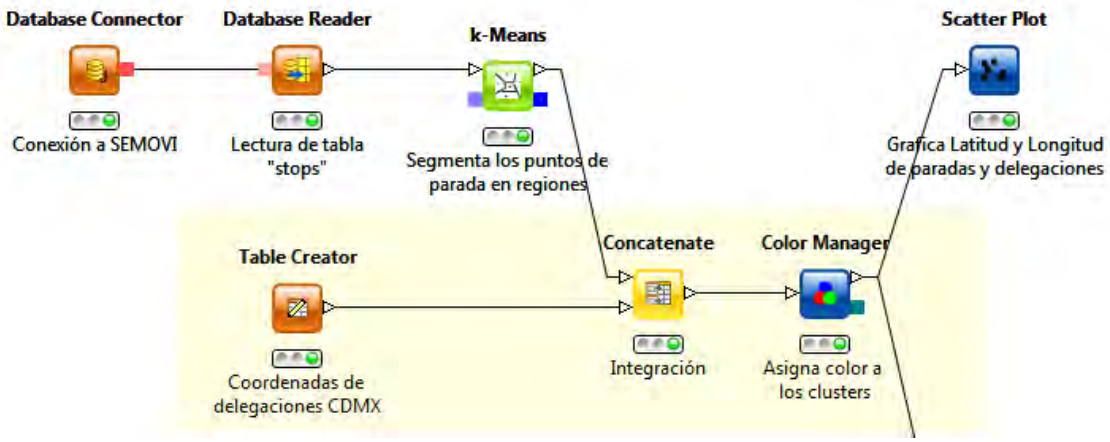


Fig. 4.37 Segmentación Zona Geográfica Parte 1

Scatter plot localiza en un plano cartesiano las coordenadas de las pardas existentes. Se distinguen por color según el cluster al que pertenecen. Así mismo, las delegaciones de CDMX se marcan en color negro como se muestra en la Figura 4.38.

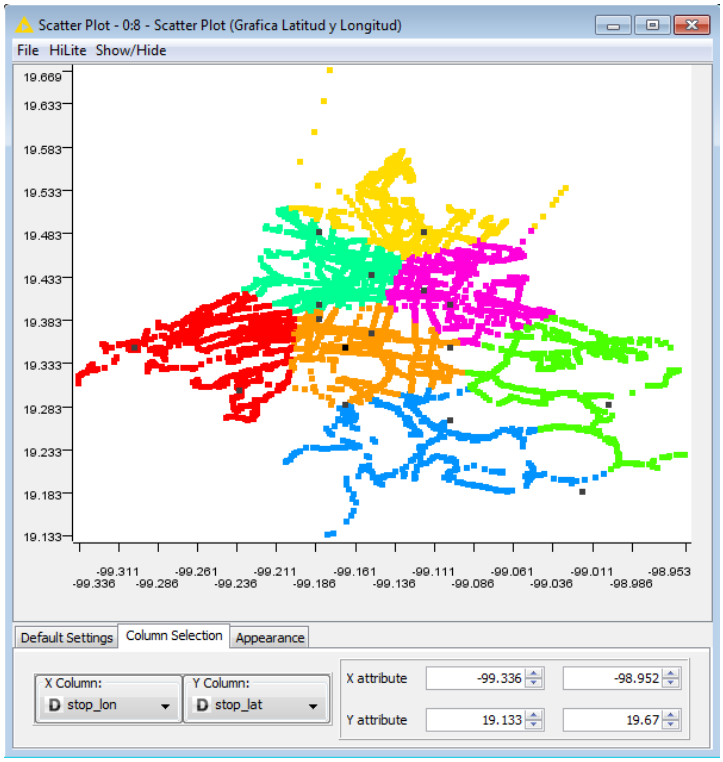


Fig. 4.38 Cobertura SEMOVI

Una vez marcadas las paradas por región se procede a identificar el viaje al que pertenecen integrando la tabla segmentada con una tabla que contiene los atributos de los viajes y rutas. Lo anterior con del nodo **Joiner** por el atributo stop_id.

Ya identificados los clusters según la gráfica Scatter Plot se les asocia un nombre según la región geográfica con el nodo **String Manipulation**.

En la Figura 4.39 se aprecia la agrupación a nivel viaje con el nodo **GroupBy**:

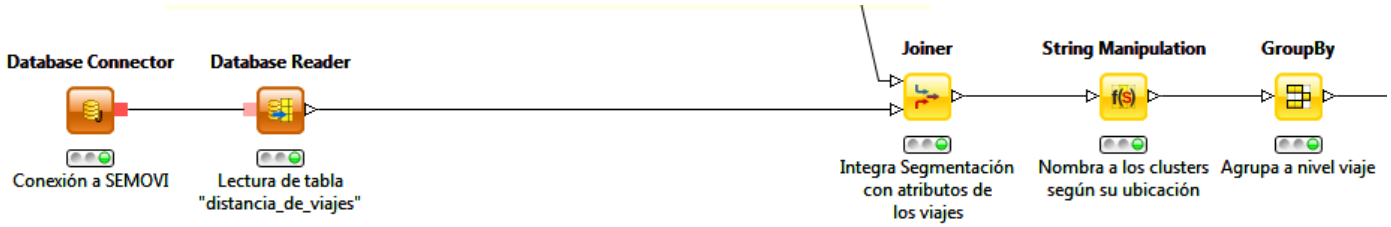


Fig. 4.39 Segmentación Zona Geográfica Parte 2

El resultado es una tabla con cada viaje y las regiones por las que pasa al hacer su recorrido. Cada viaje aparece tantas veces como regiones atraviese. Ver Fig. 4.40.

Group table - 10:22 - GroupBy (Agrupa a nivel viaje)

File

Table "default" - Rows: 3578 | Spec - Columns: 23 | Properties | Flow Variables

Row ID	Cluster	trip_id	Sum(Distancia)	Tiempo ...	agency_id	route_long_name	route_id	days_service	start_time	end_time
Row324	cluster_1	14743	16.266	30.4	METRO	Pantitlán - Observatorio	ROUTE_14243	{1, 1, 1, 1, 1, 0, 0}	05:00:00.0	10:00:00.0
Row1966	cluster_4	14743	16.266	30.4	METRO	Pantitlán - Observatorio	ROUTE_14243	{1, 1, 1, 1, 1, 0, 0}	05:00:00.0	10:00:00.0
Row325	cluster_1	14840	20.065	36.8	METRO	Cuatro Caminos - Tasqueña	ROUTE_14244	{1, 1, 1, 1, 1, 0, 0}	05:00:00.0	10:00:00.0
Row1967	cluster_4	14840	20.065	36.8	METRO	Cuatro Caminos - Tasqueña	ROUTE_14244	{1, 1, 1, 1, 1, 0, 0}	05:00:00.0	10:00:00.0
Row2590	cluster_5	14840	20.065	36.8	METRO	Cuatro Caminos - Tasqueña	ROUTE_14244	{1, 1, 1, 1, 1, 0, 0}	05:00:00.0	10:00:00.0
Row326	cluster_1	14841	20.065	37.1	METRO	Cuatro Caminos - Tasqueña	ROUTE_14244	{1, 1, 1, 1, 1, 0, 0}	05:00:00.0	10:00:00.0
Row1968	cluster_4	14841	20.065	37.1	METRO	Cuatro Caminos - Tasqueña	ROUTE_14244	{1, 1, 1, 1, 1, 0, 0}	05:00:00.0	10:00:00.0
Row2591	cluster_5	14841	20.065	37.1	METRO	Cuatro Caminos - Tasqueña	ROUTE_14244	{1, 1, 1, 1, 1, 0, 0}	05:00:00.0	10:00:00.0
Row327	cluster_1	14842	20.881	38.2	METRO	Indios Verdes - Universidad	ROUTE_14245	{1, 1, 1, 1, 1, 0, 0}	05:00:00.0	10:00:00.0
Row1033	cluster_2	14842	20.881	38.2	METRO	Indios Verdes - Universidad	ROUTE_14245	{1, 1, 1, 1, 1, 0, 0}	05:00:00.0	10:00:00.0

Fig. 4.40 Tabla Segmentada Nivel Viaje

Para contestar a la pregunta en el workflow se filtran los viajes de las rutas que pasan por una región en particular. Ver Fig. 4.41.

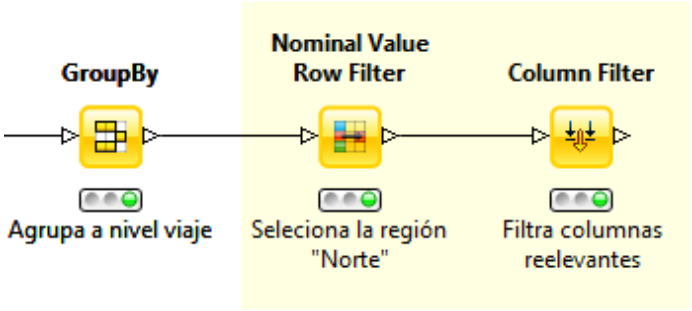
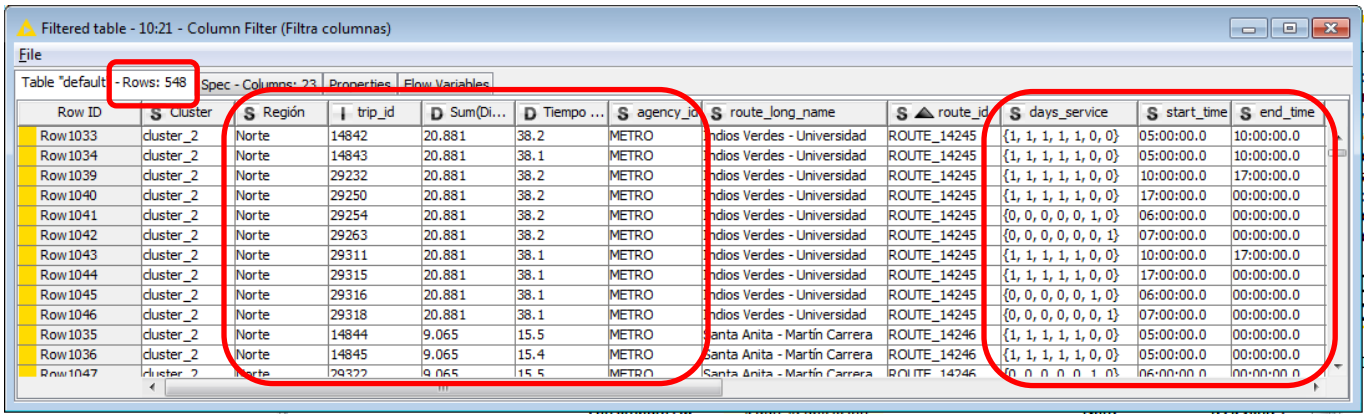


Fig. 4.41 Segmentación Zona Geográfica - Filtro

La tabla resultante contiene los viajes cuyo conjunto del total de paradas se localiza en la región norte. Ver Fig 4.42.



Row ID	Cluster	Región	trip_id	Sum(Di...	Tiempo ...	agency_id	route_long_name	route_id	days_service	start_time	end_time
Row1033	cluster_2	Norte	14842	20.881	38.2	METRO	Indios Verdes - Universidad	ROUTE_14245	{1, 1, 1, 1, 1, 0, 0}	05:00:00.0	10:00:00.0
Row1034	cluster_2	Norte	14843	20.881	38.1	METRO	Indios Verdes - Universidad	ROUTE_14245	{1, 1, 1, 1, 1, 0, 0}	05:00:00.0	10:00:00.0
Row1039	cluster_2	Norte	29232	20.881	38.2	METRO	Indios Verdes - Universidad	ROUTE_14245	{1, 1, 1, 1, 1, 0, 0}	10:00:00.0	17:00:00.0
Row1040	cluster_2	Norte	29250	20.881	38.2	METRO	Indios Verdes - Universidad	ROUTE_14245	{1, 1, 1, 1, 1, 0, 0}	17:00:00.0	00:00:00.0
Row1041	cluster_2	Norte	29254	20.881	38.2	METRO	Indios Verdes - Universidad	ROUTE_14245	{0, 0, 0, 0, 0, 1, 0}	06:00:00.0	00:00:00.0
Row1042	cluster_2	Norte	29263	20.881	38.2	METRO	Indios Verdes - Universidad	ROUTE_14245	{0, 0, 0, 0, 0, 0, 1}	07:00:00.0	00:00:00.0
Row1043	cluster_2	Norte	29311	20.881	38.1	METRO	Indios Verdes - Universidad	ROUTE_14245	{1, 1, 1, 1, 1, 0, 0}	10:00:00.0	17:00:00.0
Row1044	cluster_2	Norte	29315	20.881	38.1	METRO	Indios Verdes - Universidad	ROUTE_14245	{1, 1, 1, 1, 1, 0, 0}	17:00:00.0	00:00:00.0
Row1045	cluster_2	Norte	29316	20.881	38.1	METRO	Indios Verdes - Universidad	ROUTE_14245	{0, 0, 0, 0, 0, 1, 0}	06:00:00.0	00:00:00.0
Row1046	cluster_2	Norte	29318	20.881	38.1	METRO	Indios Verdes - Universidad	ROUTE_14245	{0, 0, 0, 0, 0, 0, 1}	07:00:00.0	00:00:00.0
Row1035	cluster_2	Norte	14844	9.065	15.5	METRO	Santa Anita - Martín Carrera	ROUTE_14246	{1, 1, 1, 1, 1, 0, 0}	05:00:00.0	00:00:00.0
Row1036	cluster_2	Norte	14845	9.065	15.4	METRO	Santa Anita - Martín Carrera	ROUTE_14246	{1, 1, 1, 1, 1, 0, 0}	05:00:00.0	00:00:00.0
Row1047	cluster_2	Norte	29322	9.065	15.5	METRO	Santa Anita - Martín Carrera	ROUTE_14246	{0, 0, 0, 0, 0, 1, 0}	06:00:00.0	00:00:00.0

Fig. 4.42 Viajes Cruce Zona Norte

Arriba en la Fig. 4.42 se puede ver que son **548 viajes** que pasan por la **región norte**. Un caso particular es la ruta que va de Universidad a Indios Verdes de la agencia METRO que recorre 20.88 km aprox. en un tiempo estimado de 38.2 min. Da servicio de lunes a viernes de 5:00am a 12:00pm, sábados de 6:00am a 12:00pm y domingos de 7:00am a 12:00pm.

6. ¿Cuál es la segmentación por zona geográfica basada en la cobertura?

La segmentación **basada en la cobertura** de todas las rutas bajo el régimen de SEMOVI se obtiene de agrupar en *clusters* las coordenadas de cada una de las paradas de la base, esto con ayuda del algoritmo k-medias:

Tabla 4.12 Segmentación Basada en Cobertura

Grupo	Región	Cantidad de paradas	Cantidad de rutas que cruzan la región	Agencias asociadas
Cluster_0	Sur	7,901	27	MB, RTP, STE
Cluster_1	Noreste	11,020	54	CC, MB, METRO, NCC, RTP, STE, SUB
Cluster_2	Norte	10,721	46	MB, METRO, NCC, RTP, STE, SUB
Cluster_3	Oeste	10,013	32	CC, METRO, RTP
Cluster_4	Noroeste	9,552	47	CC, MB, METRO, RTP, STE
Cluster_5	Centro	9,522	53	MB, METRO, NCC, RTP, STE
Cluster_6	Este	9,874	29	CC, RTP

Arriba en la Tabla 4.11 se puede observar que la región Sur y Este tiene menos cobertura a diferencia de las regiones Norte y Noreste. Además que la agencia RTP tiene mayor presencia, encontrándose en todas las regiones. Ver Fig. 4.43.

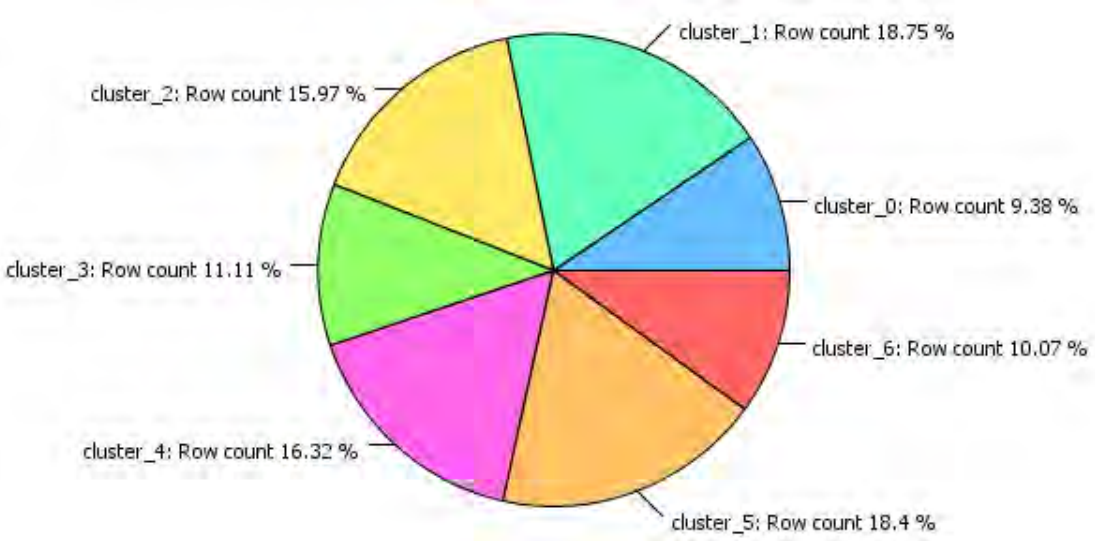


Fig. 4.43 Proporción de Rutas por Región

Con ayuda del nodo ***Pie Chart*** se puede mostrar gráficamente la proporción de las rutas por región. Ver Fig. 4.43.

Por otra parte, las etiquetas que determinan la región se asignaron usando la sentencia *replace* en la configuración del nodo **String Manipulation**. Ver Figura 4.44 y 4.45.

```
replace(replace(replace(replace(replace(replace(replace($Cluster$,"cluster_2",
"Norte"),"cluste_1","Noreste"),"cluster_4","NorOeste"),"cluster_6","Este"),"cluste
r_5","Centro"),"cluster_3","Oeste"),"cluster_0","Sur")
```

Fig. 4.44 Asignación de Regiones

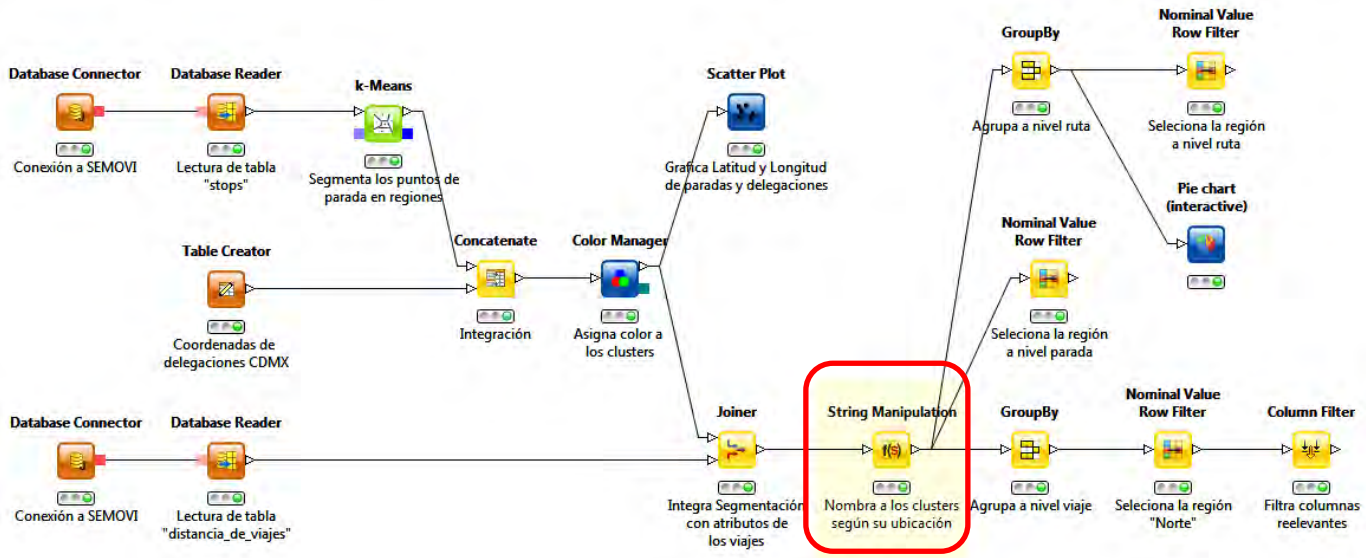


Fig. 4.45 Segmentación Zona Geográfica

Observaciones generales sobre el conjunto de datos de SEMOVI.

1. El esquema relacional de la base de datos podría mejorar definiendo llaves primarias en cada una de las entidades y agregando tablas por cada relación entre entidades con la finalidad hacer más efectiva la comprensión e integración de los datos.

2. Muchos campos podrían ser poblados y aportar funcionalidad a la base.
3. Hay datos relevantes y potencialmente útiles para responder cuestionamientos acerca de las rutas y lo relacionado con su recorrido.
4. Se cumplieron los objetivos de explotación de la base de datos ya que nos proporcionó un mejor panorama de lo que es SEMOVI sus agencias y la cobertura que actualmente hay en CDMX.

5. Conclusiones

La experiencia generada durante este proyecto, puede verse desde dos perspectivas: la técnica, que tiene que ver con el funcionamiento y uso de la plataforma; y la analítica, que se refiriere al conocimiento de la fuente, planteamiento de objetivos, limpieza de datos, manipulación, rutinas de minería, visualización y toma de decisiones. Así mismo, resultó muy valiosa la comparación con otras herramientas que persiguen los mismos fines. Consiguientemente podemos decir que KNIME es una herramienta robusta en cuanto al paquete de algoritmos que contiene implementados y en relación a su interfaz gráfica. El estudio también dejó ver que la precisión de algunos algoritmos de clasificación de la plataforma están limitados cuando se trata de procesar atributos multi-clase y/o discretos, sin embargo, solo ocurre con algunos de ellos, por lo que se puede optar por un algoritmo diferente sin tener que cambiar de herramienta.

En lo personal, los conocimientos de la licenciatura que pude aplicar en esta tesis fueron los derivados de las materias de bases de datos, probabilidad, estadística, geometría y análisis multivariado. Las bases teóricas de dichas materias me facilitaron la comprensión de los nodos. Considero que para que a un usuario le sea provechoso el uso de plataformas de minería de datos, éste debe tener conocimiento matemático y de bases de datos.

Puedo decir que la minería de datos colabora al desarrollo de sistemas y aplicaciones que apoyan la vida diaria de los humanos. Por ejemplo, la aplicación que traza una o varias rutas óptimas que permiten al usuario llegar a su destino por diferentes vías (caminando, bicicleta, transporte público o auto privado) y estima un tiempo de llegada, es una herramienta que se usa diariamente en una ciudad para desplazarnos y que requiere procesar la información almacenada. En parte, la elección de un ejemplo práctico como fue la SEMOVI, estuvo motivada por lo cotidiano que resulta el desplazamiento de un ciudadano por una ciudad a través del transporte público.

Conclusiones

La importancia de usar una herramienta de minería de datos consiste en que proporciona una gama de utilidades destinadas al análisis, a diferencia de los SMBD´s que su principal función es almacenar la información de forma estructurada. En otras palabras, KNIME provee información condensada y asimilable para el usuario, además de brindar apoyo visual que mejora el aprendizaje y facilita su uso.

Entre las ventajas más importantes de la plataforma podemos mencionar las siguientes:

- Es una plataforma ligera y sencilla de instalar.
- Hay actividad constante en los foros de discusión de la página web de la plataforma.
- La conexión a la base datos de manera local es simple.
- Permite la manipulación de datos antes de su carga al formato KNIME.
- Permite la importación y exportación de datos en diferentes formatos.
- La API posee una descripción detallada de la configuración de los nodos (parámetros, entradas, salidas, método, supuestos, y en algunos casos ejemplos).
- Es posible extender rutinas de análisis desarrolladas en otras plataformas como lo son R y Weka.

Entre las desventajas de la plataforma podemos mencionar:

- Puede resultar rígido al momento de hacer cambios estructurales si no se tienen conocimientos suficientes sobre la herramienta.
- El usuario no invierte tiempo en diseñar código, sin embargo, puede que al principio dedique un tiempo considerable en reconocer el nodo o nodos que dan solución a su necesidad.

Como sugerencias para mejorar la plataforma se identificaron a las siguientes:

- Agregar un nodo que elimine duplicados y genere dos *outputs*; tabla sin duplicados y tabla con los registros eliminados.

- Permitir que se guarden los cambios en las vistas interactivas.
- Agregar nodos que simulen datos provenientes de diferentes distribuciones (binomial, poisson, uniforme, normal, lognormal, entre otras).

El uso de la plataforma KNIME como software introductorio a la minería de datos resulta eficiente ya que un *workflow* puede entenderse como un diagrama del *proceso de descubrimiento de información* que le brinda organización y claridad al usuario. De esta manera, es posible alcanzar los objetivos de explotación de manera eficaz y obteniendo resultados profesionales y precisos.

Finalmente, como trabajo futuro se propone lo siguiente:

- Explorar a mayor detalle las extensiones que la plataforma pone a disposición.
- Crear nodos personalizados con la herramienta SDK (Software Developer Kit).
- Diseñar de workflows con parámetros variables.
- Utilizar nodos que generan ciclos en el workflow. Por ejemplo, los nodos *Counting Loop Start* y *Loop End* para ciclos con un número predeterminado de ejecuciones, o bien, *Generic Loop Start* y *Variable Condition Loop End* que definen un ciclo basado en una condición de paro. Ambos tipos, funcionan interconectándolos al inicio y al final del workflow que se desea ejecutar repetidas veces.
- Aplicar más modelos de minería de datos como lo son las redes bayesianas (método de clasificación), regresión y series de tiempo (métodos predictivos), text mining (método de compactación) y reglas de asociación difusa (método se asociación), por mencionar algunos.

Este trabajo en esencia pretende poner en contacto directo al usuario con los conceptos claves propios de la disciplina y las rutinas básicas de procesamiento de datos.

En el gobierno hay muchas bases de datos que pueden estudiarse para obtener los mayores beneficios para la población en aspectos como seguridad, empleo, salud y educación, y a través de herramientas como KNIME podemos lograr dicha meta.

Bibliografía

[1] Cios K. J., Pedrycz W., Swiniarski R. W. and Kurgan L., "A Knowlegde Discovery Approach", Springer, (2007).

[2] Coenen F., "Data Mining: Past, Present and Future," *Cambridge University Press*, no. 1-24, (2004).

[3] Witten, Ian H., "The University of Waikato," (2003). [Online]. Available: http://www.cs.waikato.ac.nz/~ihw/papers/04-IHW-Textmining.pdf.

[4] Herrera F. and Cano J. R., "Técnicas de reducción de datos en KDD. El uso de Algoritmos Evolutivos para la Selección de Instancias", *Seminario Sobre Sistemas Inteligentes*, Madrid, (2006).

[5] Kurgan L. and Cios K., "CAIM Discretization Algorithm", *IEEE Transactions on Knowledge and Data Engineering*, (2004).

[6] Batista G. E. A. P. A. and Monard M. C., "An Analysis of Four Missing Data Treatment Methods for Supervised Learning," University of Sao Paulo - USP.

[7] Allison P. D., "Handling Missing Data by Maximun Likelihood", *SAS Global Forum, Paper 312-2012.*

[8] Manuel V., "Muestreo Estadístico y Aplicaciones", Santiago de Chile: EDITORIAL UNIVERSITARIA S.A., (2005).

[9] ACM, SIGKDD, [Online]. Available: http://www.kdd.org/.

[10] KDnuggets. [Online]. Available: http://www.kdnuggets.com/2013/11/top-conferences-data-mining-data-science.html.

[11] Cios K. J., Pedrycz W., Swiniarski R. W. and Kurgan L., "A Knowlegde Discovery Approach", Springer, (2007).

Bibliografía

[12] Coenen F., "Department of Computer Science, The University of Liverpool," (2004). [Online]. Available: http://cgi.csc.liv.ac.uk/~frans/PostScriptFiles/kerDataMining_2010-8-19.pdf.

[13] Silipo R., and Mazanetz M. P., "The KNIME Cookbook", Zurich, Suiza.

[14] Silipo R., "KNIME Beginner´s Luck", Zurich, Suiza.

[16] Piatestky-Shapiro G. and Frawley W., "Knowledge Discovery in Databases", The AAAI Press, (1991).

[17] Martinez Cruz M. d. R., "Minería de Datos Multiperspectiva", México, D.F., (2007).

[18] Fayyad U., Shapiro Piatetsky G., Smyth P., "The KDD Process for extracting Useful Knowledge from Volumes of Data", vol. 39, no. 11, (Noviembre, 1996).