TariyKDD: Una herramienta genérica para el Descubrimiento de Conocimiento en Bases de Datos, débilmente acoplada con el SGBD PostgreSQL.

Andrés Oswaldo Calderon Romero
Iván Dario Ramirez Freyre
Alvaro Fernando Guevara Unigarro
Juan Carlos Alvarado Perez

Universidad de Nariño
Facultad de Ingeniería
Programa de Ingeniería de Sistemas
San Juan de Pasto
2006

TariyKDD: Una herramienta genérica para el Descubrimiento de Conocimiento en Bases de Datos, débilmente acoplada con el SGBD PostgreSQL.

Andrés Oswaldo Calderon Romero
Iván Dario Ramirez Freyre
Alvaro Fernando Guevara Unigarro
Juan Carlos Alvarado Perez

Trabajo de Grado presentado como requisito para optar la título de Ingenieros de Sistemas

Ricardo Timaran Pereira
Ph.D
Director del Proyecto

Universidad de Nariño
Facultad de Ingeniería
Programa de Ingeniería de Sistemas
San Juan de Pasto
2006

Índice general

Capítulo 1

INTRODUCCIÓN

El proceso de extraer conocimiento a partir de grandes volúmenes de datos ha sido reconocido por muchos investigadores como un tópico de investigación clave en los sistemas de bases de datos, y por muchas compañías industriales como una importante área y una oportunidad para obtener mayores ganancias [47].

El Descubrimiento de Conocimiento en Bases de Datos (DCBD) es básicamente un proceso automático en el que se combinan descubrimiento y análisis. El proceso consiste en extraer patrones en forma de reglas o funciones, a partir de los datos, para que el usuario los analice. Esta tarea implica generalmente preprocesar los datos, hacer minería de datos (data mining) y presentar resultados [3, 6, 9, 26, 37]. El DCBD se puede aplicar en diferentes dominios por ejemplo, para determinar perfiles de clientes fraudulentos (evasión de impuestos), para descubrir relaciones implícitas existentes entre síntomas y enfermedades, entre características técnicas y diagnóstico del estado de equipos y máquinas, para determinar perfiles de estudiantes "académicamente exitosos" en términos de sus características socioeconómicas, para determinar patrones de compra de los clientes en sus canastas de mercado, entre otras.

Las investigaciones en DCBD, se centraron inicialmente en definir nuevas operaciones de descubrimiento de patrones y desarrollar algoritmos para estas. Investigaciones posteriores se han focalizado en el problema de integrar DCBD con Sistemas Gestores de Bases de Datos (SGBD) ofreciendo como resultado el desarrollo de herramientas DCBD cuyas arquitecturas se pueden clasificar en una de tres categorías: débilmente acopladas, medianamente acopladas y fuertemente acopladas con el SGBD [46].

Una herramienta DCBD debe integrar una variedad de componentes (técnicas

de minería de datos, consultas, métodos de visualización, interfaces, etc.), que juntos puedan eficientemente identificar y extraer patrones interesantes y útiles de los datos almacenados en las bases de datos. De acuerdo a las tareas que desarrollen, las herramientas DCBD se clasifican en tres grupos: herramientas genéricas de tareas sencillas, herramientas genéricas de tareas múltiples y herramientas de dominio específico [37].

En este documento se presenta el trabajo de grado para optar por el título de Ingeniero de Sistemas. Fruto de la presente investigación es el desarrollo de "TariyKDD: Una herramienta genérica de Descubrimiento de Conocimiento en Bases de Datos débilmente acoplada con el SGBD PostgreSQL", en la cual también se implementaron los algoritmos EquipAsso [50, 49, 48] y MateTree [48] para las tareas de Asociación y Clasificación propuestos por Timarán en [48] y sobre los cuales se realizarón ciertas pruebas para medir su rendimiento.

El resto de este documento esta organizado de la siguiente manera. En la siguiente sección se especifica el Tema de la propuesta, se lo enmarca dentro de una línea de investigación y se lo delimita. A continuación se describe el problema objeto de estudio. En la sección 4 se especifican los objetivos generales y específicos del anteproyecto. En la sección 5 se presenta la justificación de la propuesta de trabajo de grado. En la sección 6 se presenta el estado general del arte en el área de integración de DCBD y SGBD. En la sección 7 se desarrolla todo lo concerniente al análisis orientado a objetos UML que se realizo para construir la herramienta y finalmente en la sección 8 se presentan las conclusiones, recomendaciones, referencias bibliográficas y anexos.

Capítulo 2

TEMA

2.1. Titulo

TariyKDD: Una herramienta genérica para el Descubrimiento de Conocimiento en Bases de Datos, débilmente acoplada con el SGBD PostgreSQL.

2.2. Línea de Investigación

El presente trabajo de grado, se encuentra inscrito bajo la **línea de software y manejo de información**, se enmarca dentro del área de las Bases de Datos y específicamente en la subárea de arquitecturas de integración del Proceso de descubrimiento en Bases de datos con Sistemas Gestores de bases de Datos.

2.3. Alcance y Delimitación

TARIYKDD es una herramienta que contempla todas las etapas del proceso DCBD, es decir: Selección, preprocesamiento, transformación, minería de datos y visualización [3, 6, 23]. En la etapa de minería de datos se implementaron las tareas de Asociación y Clasificación. En estas dos tareas, se utilizaron los operadores algebraicos relacionales y primitivas SQL para DCBD, desarrollados por Timarán [51, 48], con los algoritmos EquipAsso [50, 49, 48] y Mate-tree [48]. En la etapa de Visualización se desarrollo una interfaz gráfica que le permite al usuario interac-

tuar de manera fácil con la herramienta.

Para los algoritmos implementados, se hicieron pruebas de rendimiento, utilizando conjuntos de datos reales y se los comparo con los algoritmos Apriori Híbrido [6], FP-Growth [22, 26] y C.4.5 [38, 29].

Capítulo 3

PROBLEMA OBJETO DE ESTUDIO

3.1. Descripción del Problema

Muchos investigadores [9, 8, 22, 42] han reconocido la necesidad de integrar los sistemas de descubrimiento de conocimiento y bases de datos, haciendo de esta un área activa de investigación. La gran mayoría de herramientas de DCBD tienen una arquitectura débilmente acoplada con un Sistema Gestor de Bases de Datos [46].

Algunas herramientas como Alice [41], C5.0 RuleQuest [27], Qyield [10], CoverStory [31] ofrecen soporte únicamente en la etapa de minería de datos y requieren un pre y un post procesamiento de los datos. Hay una gran cantidad de este tipo de herramientas [30], especialmente para clasificación apoyadas en árboles de decisión, redes neuronales y aprendizaje basado en ejemplos. El usuario de este tipo de herramientas puede integrarlas a otros módulos como parte de una aplicación completa [37].

Otros ofrecen soporte en más de una etapa del proceso de DCBD y una variedad de tareas de descubrimiento, típicamente, combinando clasificación, visualización, consulta y clustering, entre otras [37]. En este grupo están Clementine [43], DBMiner [21, 20, 18], DBLearn [19], Data Mine [26], IMACS [7], Intelligent Miner [11], Quest [4] entre otras. Una evaluación de un gran número de herramientas de este tipo se puede encontrar en [14].

Todas estas herramientas necesitan de la adquisición de costosas licencias para

su utilización. Este hecho limita a las pequeñas y medianas empresas u organizaciones, al acceso de herramientas DCBD para la toma de decisiones, que inciden directamente en la obtención de mayores ganancias y en el aumento de su competitividad.

Por esta razón, se plantea el desarrollo de una herramienta genérica de DCBD, débilmente acoplada, bajo software libre, que permita el acceso a este tipo de herramientas sin ningún tipo de restricciones, a las pequeñas y medianas empresas u organizaciones de nuestro país o de cualquier parte del mundo.

3.2. Formulación del Problema

¿El desarrollo de una herramienta genérica para el Descubrimiento de Conocimiento en bases de datos débilmente acoplada con el SGBD PostgreSQL bajo software libre, facilitará a las pequeñas y medianas empresas la toma de decisiones?

Capítulo 4

OBJETIVOS

4.1. Objetivo General

Desarrollar una herramienta genérica para el Descubrimiento de Conocimiento en bases de datos débilmente acoplada con el sistema gestor de bases de datos PostgreSQL, bajo los lineamientos del Software Libre, que facilite la toma de decisiones a las pequeñas y medianas empresas u organizaciones de nuestro país y de cualquier parte del mundo.

4.2. Objetivos Específicos

1. Estudiar y Analizar diferentes herramientas DCBD débilmente acopladas con un SGBD.

2. Analizar, diseñar y desarrollar programas que permitan la selección, preprocesamiento y transformación de datos.

3. Analizar, diseñar y desarrollar programas que implementen los operadores algebraicos y primitivas SQL para las tareas de Asociación y Clasificación de datos.

4. Analizar, diseñar y desarrollar programas que implementen los algoritmos EquipAsso, MateTree, Apriori Híbrido, FP-Growth y C4.5.

5. Analizar, diseñar y desarrollar programas que permitan visualizar de manera gráfica las reglas de asociación y clasificación.

6. Integrar todos los programas desarrollados en una sola herramienta para DCBD.

7. Implementar la conexión de la herramienta DCBD con el SGBD PostgreSQL.

8. Obtener conjuntos de datos reales para la realización de las pruebas con la herramienta DCBD.

9. Realizar las pruebas y analizar los resultados con la herramienta DCBD débilmente acoplada con PostgreSQL.

10. Realizar las pruebas de rendimiento con los diferentes algoritmos implementados.

11. Dar a conocer los resultados del proyecto a través de la publicación de un artículo en una revista o conferencia nacional o internacional.

Capítulo 5

JUSTIFICACIÓN

Todas las herramientas de DCBD o comúnmente conocidas como herramientas de minería de datos sirven en las organizaciones para apoyar la toma de decisiones de tipo semiestructurado, única o que cambian rápidamente, y que no es fácil especificar por adelantado. Es evidente que por su diseño, estas herramientas tienen mayor capacidad analítica que otras. Están construidas explícitamente con diversos modelos para obtener patrones insospechados a partir de los datos. Apoyan la toma de decisiones al permitir a los usuarios extraer información útil que antes estaba enterrada en montañas de datos. Diversas herramientas de minería de datos disponibles en el mercado ofrecen diferentes tipos de arquitecturas que determinan, en alguna medida, su versatilidad y su costo. Por lo general todas estas son demasiado costosas, necesitan de licenciamiento para su uso y de software específico.

El desarrollo de TARIYKDD, como una herramienta DCBD bajo licencia pública GNU, permitirá que empresas u organizaciones que por su tamaño no puedan acceder a herramientas DCBD propietarias, utilicen esta tecnología para mejorar la toma de decisiones, maximicen sus ganancias con decisiones acertadas y eleven su poder competitivo, ya que el ritmo actual del mundo y la globalización así lo requieren.

Por otra parte, TARIYKDD se convierte en otro aporte más en el área del Descubrimiento de Conocimiento en bases de datos que la Universidad de Nariño, a través de su programa de Ingeniería de Sistemas, hace al mundo, contribuyendo a la investigación científica y al desarrollo de la región y del país.

Por ser TARIYKDD una herramienta genérica de DCBD, puede utilizarse en diferentes campos como la industria, la banca, la salud y la educación, entre otros.

Capítulo 6

MARCO TEORICO

6.1. El Proceso de Descubrimiento de Conocimiento en Bases de Datos - DCBD

El proceso de DCBD es el proceso que utiliza métodos de minería de datos (algoritmos) para extraer (identificar) patrones que evaluados e interpretados, de acuerdo a las especificaciones de medidas y umbrales, usando una base de datos con alguna selección, preprocesamiento, muestreo y transformación, se obtiene lo que se piensa es conocimiento [13].

El proceso de DCBD es interactivo e iterativo, involucra numerosos pasos con la intervención del usuario en la toma de muchas decisiones y se resumen en las siguientes etapas:

- Selección.
- Preprocesamiento / Data cleaning.
- Transformación / Reducción.
- Minería de Datos (Data Mining).
- Interpretación / evaluación.

6.2. Arquitecturas de Integración de las Herramientas DCBD con un SGBD

Las arquitecturas de integración de las herramientas DCBD con un SGBD se pueden ubicar en una de tres tipos: herramientas débilmente acopladas, medianamente acopladas y fuertemente acopladas con un SGBD [46].

Una arquitectura es débilmente acoplada cuando los algoritmos de Minería de Datos y demás componentes se encuentran en una capa externa al SGBD, por fuera del núcleo y su integración con este se hace a partir de una interfaz [46].

Una arquitectura es medianamente acoplada cuando ciertas tareas y algoritmos de descubrimiento de patrones se encuentran formando parte del SGBD mediante procedimientos almacenados o funciones definidas por el usuario [46].

Una arquitectura es fuertemente acoplada cuando la totalidad de las tareas y algoritmos de descubrimiento de patrones forman parte del SGBD como una operación primitiva, dotándolo de las capacidades de descubrimiento de conocimiento y posibilitándolo para desarrollar aplicaciones de este tipo [46].

Por otra parte, de acuerdo a las tareas que desarrollen, las herramientas DCBD se clasifican en tres grupos: herramientas genéricas de tareas sencillas, herramientas genéricas de tareas múltiples y herramientas de dominio específico.

Las Herramientas genéricas de tareas sencillas principalmente soportan solamente la etapa de minería de datos en el proceso de DCBD y requieren un pre y un post procesamiento de los datos. El usuario final de estas herramientas es típicamente un consultor o un desarrollador quien podría integrarlas con otros módulos como parte de una completa aplicación.

Las Herramientas genéricas de tareas múltiples realizan una variedad de tareas de descubrimiento, típicamente combinando clasificación, asociación, visualización, clustering, entre otros. Soportan diferentes etapas del proceso de DCBD. El usuario final de estas herramientas es un analista quien entiende la manipulación de los datos.

Finalmente, las Herramientas de dominio específico, soportan descubrimiento solamente en un dominio específico y hablan el lenguaje del usuario final, quien necesita conocer muy poco sobre el proceso de análisis.

6.3. Implementación de Herramientas DCBD débilmente acopladas con un SGBD

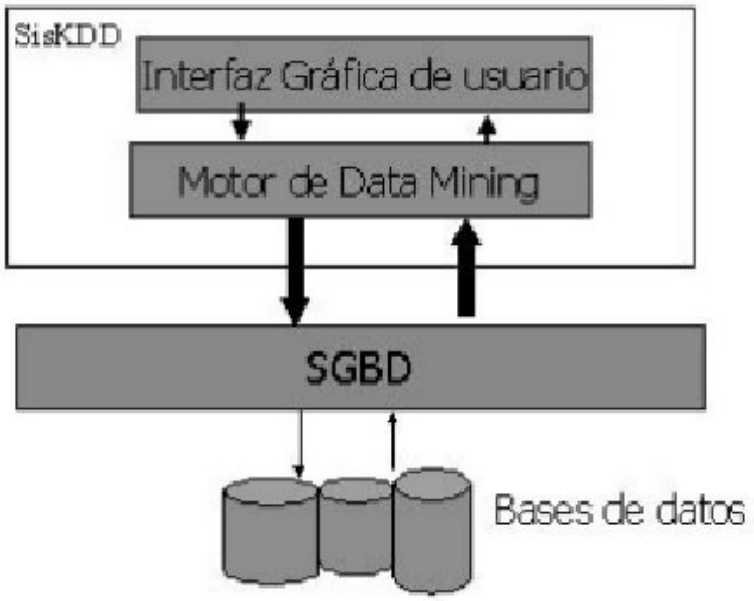

Figura 6.1: Arquitectura DCBD débilmente acoplada

La implementación de herramientas DCBD débilmente acopladas con un SGBD se hace a través de SQL embebido en el lenguaje anfitrión del motor de minería de datos [5]. Los datos residen en el SGBD y son leídos registro por registro a través de ODBC, JDBC o de una interfaz de cursores SQL. La ventaja de esta arquitectura es su portabilidad. Sus principales desventajas son la escalabilidad y el rendimiento. El problema de escalabilidad consiste en que las herramientas y aplicaciones bajo este tipo de arquitectura, cargan todo el conjunto de datos en memoria, lo que las limita para el manejo de grandes cantidades de datos. El bajo rendimiento se debe a que los registros son copiados uno por uno del espacio de direccionamiento de la base de datos al espacio de direccionamiento de la aplicación de minería de datos [8, 24] y estas operaciones de entrada/salida, cuando se manejan grandes volúmenes de datos, son bastante costosas, a pesar de la optimización de lectura por bloques presente en muchos SGBD (Oracle, DB2, Informix, PostgreSQL.) donde un bloque de tuplas puede ser leído al tiempo (figura 1).

6.4. Estado del Arte

En el desarrollo de nuestro trabajo de grado realizamos un estudio de herramientas de minería de datos elaboradas por otras universidades y centros de investigación que nos permitieron ver el estado actual de este tipo de aplicaciones. A continuación se describen las herramientas estudiadas.

6.4.1. WEKA - Waikato Environment for Knowledge Analysis

WEKA es una herramienta libre de minería de Datos realizada en el departamento de Ciencias de la Computación de la Universidad de Waikato en Hamilton, Nueva Zelanda. Los principales gestores de este proyecto son Ian H. Waitten y Eibe Frank autores del libro Data Mining: Practical Machine Learning Tools and Techniques with Java Implementations [52] cuyo octavo capítulo sirve como tutorial de Weka y es libremente distribuido junto con el ejecutable y el código fuente. Las ultimas versiones estables de Weka pueden ser descargadas de la página oficial del Proyecto [17].

La implementación de la herramienta fue hecha bajo la plataforma Java utilizando la versión 1.4.2 de su máquina virtual. Al ser desarrollada en Java, Weka posee todas las características de una herramienta orientada a objetos con la ventaja de ser multiplataforma, la ultima versión (3.4) ha sido evaluada bajo ambientes GNU/Linux, Macintosh y Windows siguiendo una arquitectura Cliente/Servidor lo que permite ejecutar ciertas tareas de manera distribuida.

La conexión a la fuente de datos puede hacerse directamente hacia un archivo plano, considerando varios formatos como C4.5 y CVS aunque el formato oficial es el ARFF que se explica más adelante, o a través de un driver JDBC hacia diferentes Sistemas Gestores de Bases de Datos(SGBD).

Entre las principales características de Weka esta su modularidad la cual se fundamenta en un estricto y estandarizado formato de entrada de datos que denominan ARFF (Atributte - Relation File Format), a partir de este tipo de archivo todos los algoritmos de minería implementados en Weka son trabajados. Nuevas implementaciones y adiciones deben ajustarse a este formato.

El formato ARFF consiste en una archivo sencillo de texto que funciona a partir de etiquetas, similar a XML, y que debe cumplir con dos secciones: una cabecera y un conjunto de datos. La cabecera contiene las etiquetas del nombre de la relación

(@relation) y los atributos (@atributte), que describen los tipos y el orden de los datos. La sección datos (@data) en un archivo ARFF contiene todos los datos del conjunto que se quiere evaluar separados por comas y en el mismo orden en el que aparecen en la sección de atributos [15].

La conexión al SGBD se hace en primera instancia a través de una interfaz gráfica de usuario construyendo una sentencia SQL siguiendo un modelo relacional pero a partir de construida la tabla a minar se trabaja en adelante fuera de línea a través de flujos (streams) que se comunican con archivos en disco duro.

WEKA cubre gran parte del proceso de Descubrimiento de Conocimiento, las características especificas de minería de datos que se pueden ver son preprocesamiento y análisis de datos, clasificación, clustering, asociación y visualización de resultados.

WEKA posee una rica colección de algoritmos que soportan el proceso de minería que aplican diversas metodologías como arboles de decisión, conjuntos difusos, reglas de inducción, métodos estadísticos y redes bayesianas.

Por poseer diferentes modos de interfaces gráficas, WEKA se puede considerar una herramienta orientada al usuario aunque cabe aclarar que exige un buen dominio de conceptos de minería de datos y del objeto de análisis. Muchas tareas se encuentran soportadas aunque no del todo automatizadas por lo que el proceso de descubrimiento debe ser guiado aún por un analista.

Un análisis completo de las características de WEKA con respecto a otras aplicaciones presentes en el mercado se puede encontrar en [14].

6.4.2. ADaM - Algorithm Development and Mining System

El proyecto ADaM es desarrollado por el Centro de Sistemas y Tecnologías de la Información de la Universidad de Alabama en Huntsville, Estados Unidos. Este sistema es usado principalmente para aplicar técnicas de minería a datos científicos obtenidos de sensores remotos [45].

ADaM es un conjunto de herramientas de minería y procesamiento de imágenes que consta de varios componentes interoperables que pueden usarse en conjunto para realizar aplicaciones en la solución de diversos problemas. En la actualidad,

ADaM (en su versión 4.0.2) cubre cerca de 120 componentes [44] que pueden ser configurados para crear procesos de minería personalizados. Nuevos componentes pueden fácilmente integrarse a un sistema o proceso existente.

ADaM 4.0.2 provee soporte a través del uso de componentes autónomos dentro de una arquitectura distribuida. Cada componente esta desarrollado en C, C++ u otra interfaz de programación de aplicaciones y entrega un ejecutable a modo de script donde cada componente recibe sus parámetros a través de línea de comandos y arroja los resultados en ficheros que pueden ser utilizados a su vez por otros componentes ADaM. Eventualmente se ofrece Servicios Web de algunos componentes por lo que se puede acceder a ellos a través de la Web.

Lastimosamente el acceso a fuentes de datos es limitada no ofreciendo conexión directa hacia un sistema gestor de bases de datos. ADaM trabaja generalmente con ficheros ARFF [15] que son generados por sus componentes.

Dentro de las herramientas ofrecidas por ADaM encontramos soporte al preprocesamiento de datos y de imágenes, clasificación, clustering y asociación. La visualización y análisis de resultados se deja para ser implementado por el sistema que invoca a los componentes. ADaM 4.0.2 ofrece un amplio conjunto de herramienta que implementa diversos metodologías dentro del área del descubrimiento de conocimiento. Existen módulos que implementan árboles de decisión, reglas de asociación, métodos estadísticos, algoritmos genéticos y redes bayesianas.

ADaM 4.0.2 no soporta una interfaz gráfica de usuario, se limita a ofrecer un conjunto de herramientas para ser utilizadas en la construcción de sistemas que cubran diferentes ámbitos. Por tal motivo, es necesario un buen conocimiento de los conceptos de minería de datos, a parte de fundamentos en el análisis y procesamiento de imágenes. No obstante, ADaM ofrece un muy buen soporte a sistemas donde se busque descubrimiento de conocimiento, un ejemplo de la implementación de ADaM puede verse en [2] donde se utilizan componentes ADaM en el análisis e interpretación de imágenes satelitales de ciclones para estimar la máxima velocidad de los vientos.

6.4.3. Orange - Data Mining Fruitful and Fun

Construído en el Laborátorio de Inteligencia Artifícial de la Fácultad de Computación y Ciencias de la Información de la Universidad de Liubliana, en Eslovenia y ya que fué liberado bajo la Licencia Pública General (GPL) puede ser descargado desde su sitio oficial [33].

Orange [12] en su núcleo es una librería de objetos C++ y rutinas que incluyen entre otros, algoritmos estandar y no estandar de Minería de Datos y Aprendizaje Maquinal, además de rutinas para la entrada y manipulación de datos. Orange provee un ambiente para que el usuario final pueda acceder a la herramienta a través de scripts hechos en Python y que se encuentran un nivel por encima del núcleo en C++. Entonces el usuario puede incorporar nuevos algoritmos o utilizar código ya elaborado y que le permiten cargar, limpiar y minar datos, así como también imprimir árboles de desición y reglas de asociación.

Otra característica de Orange es la inclusión de un conjunto de Widgets gráficos que usan métodos y módulos del núcleo central (C++) brindando una intefáz agradable e intuitiva al usuario.

Los Widgets de la interfaz gráfica y los módulos en Python incluyen tareas de Minería de Datos desde preprocesamiento hasta modelamiento y evaluación. Entre otras funciones estos componentes poseen técnicas para:

Entrada de datos Orange proporciona soporte para varios formatos populares de datos. Como por ejemplo:

- **.tab** Formato nativo de Orange. La primera línea tiene los nombres de los atributos, la segunda línea dice cual es el tipo de datos de cada columna (discreto o continuo) y en adelante se encuentran los datos separados a través de tabuladores.
- **.c45** Estos archivos están compuestos por dos archivos uno con extención .names que tiene los nombres de las columnas separados por comas y otro .data con los datos separados también con comas.

Manipulación de datos y preprocesamiento Entre las tareas que Orange incluye en este apartado tenemos visualización gráfica y estadística de datos, procesamiento de filtros, discretización y construcción de nuevos atributos entre otros métodos.

Minería de Datos y Aprendizaje Máquinal Dentro de esta rama Orange incluye variedad de algoritmos de asociación, clasificación, regresión logistica, regresión lineal, árboles de regresión y acercamientos basados en instancias (instance-based approaches).

Contenedores Para la calibración de la predicción de probabilidades de modelos de clasificación.

Métodos de evaluación Que ayudan a medir la exactiud de un clasificador.

Podría catalogarse como una falencia el hecho de que Orange no incluya un modulo para la conexión a un Sistema Gestor de Bases de Datos, pero si se revisa bien la documentación de los modulos se encuentra que ha sido desarrollado uno para la conexión a MySQL (orngMySQL [34]). El modulo provee una entrada a MySQL a través de este modulo y de sencillos comandos en Python los datos de las tablas pueden transferidos desde y hacia MySQL. Así mismo programadores independientes han desarrollado varios modulos entre los que se destaca uno para algoritmos de Inteligencia Artificial.

Dentro de las herramientas de Minería de Datos, Orange podría catalogarse como una Herramienta Genérica Multitarea. Estas herramientas realizan una variedad de tareas de descubrimiento, típicamente combinando clasificación, asociación , visualización, clustering, entre otros. Soportan diferentes etapas del proceso de DCBD. El usuario final de estas herramientas es un analista quien entiende la manipulación de los datos.

Orange funciona en varias plataformas como Linux, Mac y Windows y desde su sitio web [33] se puede descargar toda la documentación disponible para su instalación. Al instalar Orange el usuario puede acceder a una completa información de la herramienta, así como a prácticos tutoriales y manuales que permiten familiarizarse con la misma. Su sitio web [33] incluye tutoriales básicos sobre Python y Orange, manuales para desarrolladores más avanzados y además tener acceso a los foros de Orange en donde se despejan todos los interrogantes sobre esta herramienta.

En si Orange esta hecho para usuarios experimentados e investigadores en aprendizaje maquinal con la ventaja de que el software fue liberado con licencia GPL, por tanto cualquier persona es libre de desarrollar y probar sus propios algoritmos reusando tanto código como sea posible.

6.4.4. TANAGRA - A Free Software for Research and Academic Purposes

TANAGRA [39] es software de Minería de Datos con própositos acádemicos e investigativos, desarrollado por Ricco Rakotomalala, miembro del Equipo de Investigación en Ingeniería del Conocimiento (ERIC - Equipe de Recherche en Ingénierie des Connaissances [1]) de la Universidad de Lyon, Francia. Conjuga varios métodos de Minería de Datos como análisis exploratorio de datos, aprendizaje estadístico

y aprendizaje maquinal.

Este proyecto es el sucesor de SIPINA, el cuál implementa varios algoritmos de aprendizaje supervisado, especialmente la construcción visual de árboles de desición. TANAGRA es más potente, contiene además de algunos paradigmas supervisados, clustering, análisis factorial, estadística paramétrica y no-paramétrica, reglas de asociación, selección de características y construcción de algoritmos.

TANAGRA es un proyecto open source, así que cualquier investigador puede acceder al código fuente y añadir sus propios algoritmos, en cuanto este de acuerdo con la licencia de distribución.

El principal próposito de TANAGRA es proponer a los investigadores y estudiantes una arquitectura de software para Minería de Datos que permita el análisis de datos tanto reales como sintéticos.

El segundo próposito de TANAGRA es proponer a los investigadores una arquitectura que les permita añadir sus propios algoritmos y métodos, para así comparar sus rendimientos. TANAGRA es más una plataforma experimental, que permite ir a lo esencial y obviarse la programación del manejo de los datos.

El tercero y último próposito va dirigido a desarrolladores novatos y consiste en difundir una metodología para construir este tipo de software con la ventaja de tener acceso al código fuente, de esta forma un desarrollador puede ver como ha sido construido el software, cuales son los problemas a evadir, cuales son los principales pasos del proyecto y que herramientas y librerías usar.

Lo que no incluye TANAGRA es lo que constituye la fuerza del software comercial en el campo de la Minería de Datos: Grandes fuentes de datos, acceso directo a bodegas y bases de datos (Data Warehouses y Data Bases) así como la aplicación de data cleaning.

Para realizar trabajos de Minería de Datos con Tanagra, se deben crear esquemas y diagramas de flujo y utilizar sus diferentes componentes que van desde la visualización de datos, estadísticas, construcción y selección de instancias, regresión y asociación entre otros.

El formato del conjunto de datos de Tanagra es un archivo plano (.txt) separado por tabulaciones, en la primera línea tiene un encabezado con el nombre de

los atributos y de la clase, a continuación vienen los datos con el mismo formato de separación con tabuladores. Tanagra también acepta conjuntos de datos generados en Excel y uno de los estandares en Minería de Datos, el formato de WEKA (archivos .arff). Tanagra permite cargar un solo conjunto de datos.

A medida que se trabaja con TANAGRA y se van generando reportes de resultados, existe la posibilidad de guardarlos en formato HTML.

La paleta de componentes de Tanagra tiene implementaciones de tareas de Minería de Datos que van desde preprocesamiento hasta modelamiento y evaluación. Entre otras TANAGRA permite usar técnicas para:

Entrada de datos Con soporte para varios formatos populares de datos.

Manipulación de datos y preprocesamiento Como muestreo, filtrado, discretización y construcción de nuevos atributos entre otros métodos.

Construcción de Modelos Métodos para la construcción de modelos de clasificación, que incluyen árboles de clasificación, clasificadores Naive-Bayes y regresión logística.

Métodos de regresión Como regresión multiple lineal.

Métodos de evaluación Utilizados para medir la exactitud de un clasificador.

Al ser TANAGRA un proyecto Open Source es facilmente descargable desde su sitio web [40], el cual contiene tutoriales, referencias y conjuntos de datos.

6.4.5. AlphaMiner

AlphaMiner es desarrollado por el E-Business Technology Institute (ETI) de la Universidad de Hong Kong [28] bajo el apoyo del Fondo para la Innovación y la Tecnología (ITF) [35] del Gobierno de la Región Especial Administrativa de Hong Kong (HKSAR).

AlphaMiner es un sistema de Minería de Datos, Open Source, desarrollado en java de propósito general pero más enfocado al ambiente empresarial. Implementa algoritmos de asociación, clasificación, clustering y regresión logística. Posee una gama amplia de funcionalidad para el usuario, al realizar cualquier proceso minero dando la posibilidad al usuario de escoger los pasos del proceso KDD que más se ajusten a sus necesidades, por medio de nodos que el analista integra a un árbol

KDD siguiendo la metodología Drag & Drop. Es por eso que el proceso KDD en AlphaMiner no sigue un orden estructurado, sino que sigue la secuencia brindada por el propósito u objetivo del analista, ademas brinda la visualicion estadística de distintas maneras, permitiendo un analisis más preciso por parte del analista.

El principal objetivo de AlphaMiner es la inteligencia Comercial Económica o Inteligencia de Negocios (BI - Business Inteligence) es uno de los medios más importantes para que las compañías tomen las decisiones comerciales más acertadas. Las soluciones de BI son costosas y sólo empresas grandes pueden permitirse el lujo de tenerlas. Por tanto las compañías pequeñas tienen una gran desventaja. AlphaMiner proporciona las tecnologías de BI de forma económica para dichas empresas para que den soporte a sus decisiones en el ambiente de negocio cambiante rápido.

AlphaMiner tiene dos componentes principales:

1. Una base de conocimiento.
2. Un árbol KDD, que permite integrar nodos artefacto que proporcionan varias funciones para crear, revisar, anular, interpretar y argumentar los distintos análisis de datos

AlphaMiner implementa los siguientes pasos del proceso KDD:

1. Acceso de diferentes formas a las fuentes datos.
2. Exploración de datos de diferentes maneras.
3. Preparación de datos.
4. Vinculación de los distintos modelos mineros.
5. Análisis a partir de los modelos.
6. Despliegue de modelos al ambiente empresarial.

La característica más importante de AlphaMiner, es su capacidad para almacenar los datos después del núcleo minero en una base de conocimiento, que puede ser reutilizada. Esta función aumenta su utilidad significativamente, y brinda un gran apoyo logístico al nivel estratégico de una empresa. Aventajando cualquier sistema tradicional de manipulación de datos. AlphaMiner proporciona una funcionalidad adicional para construir los modelos de Minería de Datos, conformando

una sinergia entre los distintos algoritmos de minería.

AlphaMiner se asemeja a Tariy en varias características, por un lado el lenguaje de programación en el que fue implementado es JAVA, por otro lado es Open Source, tiene conectividad JDBC a los distintos gestores de bases de datos, además de conectividad con archivos de Excel. Otra semejanza es el tipo de formato que presenta para el manejo de tablas unívaluadas en algoritmos de asociación, específicamente en bases de datos de supermercados, ya que después de escoger los campos de dicha base de datos se la lleva a un formato de dos columnas una para la correspondiente transacción y otra para el ítem.

6.4.6. YALE - Yet Another Learning Environment

YALE [53, 32] es un ambiente para la realización de experimentos de aprendizaje maquinal y Minería de Datos. A través de un gran número de operadores se pueden crear los experimentos, los cuales pueden ser anidados arbitrariamente y cuya configuración es descrita a través de archivos XML que pueden ser fácilmente creados por medio de una interfaz gráfica de usuario. Las aplicaciones de YALE cubren tanto investigación como tareas de Minería de Datos del mundo real.

Desde el año 2001 YALE ha sido desarrollado en la Unidad de Inteligencia Artificial de la Universidad de Dortmund [36] en conjunto con el Centro de Investigación Colaborativa en Inteligencia Computacional (Sonderforschungsbereich 531).

El concepto de operador modular permite el diseño de cadenas complejas de operadores anidados para el desarrollo de un gran número de problemas de aprendizaje. El manejo de los datos es transparente para los operadores. Ellos no tienen nada que ver con el formato de datos o con las diferentes presentaciones de los mismos. El kernel de Yale se hace a cargo de las transformaciones necesarias. Yale es ampliamente usada por investigadores y compañías de Minería de Datos.

Modelando Procesos De Descubrimiento De Conocimiento Como Arboles De Operadores

Los procesos de descubrimiento de conocimiento son vistos frecuentemente como invocaciones secuenciales de métodos simples. Por ejemplo, después de cargar datos, uno podría aplicar un paso de preprocesamiento seguido de una invocación a un método de clasificación. El resultado en este caso es modelo aprendido, el

cual puede ser aplicado a datos nuevos o no revisados todavía. Una posible abstracción de estos métodos simples es el concepto de operadores. Cada operador recibe su entrada, desempeña una determinada función y entrega una salida. Desde ahi, el método secuencial de invocaciones corresponde a una cadena de operadores. Aunque este modelo de cadena es suficiente para la realización de muchas tareas básicas de descubrimiento de conocimiento, estas cadenas planas son a menudo insuficientes para el modelado de procesos de descubrimiento de conocimiento más complejas.

Una aproximación común para la realización del diseño de experimentos más complejos es diseñar las combinaciones de operadores como un gráfico direccionado. Cada vertice del gráfico corresponde a un operdor simple. Si dos operadores son conectados, la salida del primer operador será usada como entrada en el segundo operador. Por un lado, diseñar procesos de descubrimiento de conocimiento con la ayuda de gráficosdireccionados es muy poderoso. Por el otro lado, existe una desventaja principal: debido a la pérdida de restricciones y la necesidad de ordenar topológicamente el diseño de experimentos es menudo poco intuitivo y las validaciones automáticas son difíciles de hacer.

Yale ofrece un compromiso entre la simplicidad de cadenas de operadores y la potencia de los gráficos direccionados a través del modelamiento de procesos de descubrimiento de conocimiento por medio de árboles de operadores. Al igual que los lenguajes de programación , el uso de árboles de operadores permite el uso de conceptos como ciclos, condiciones, u otros esquemas de aplicación. Las hojas en el árbol de operadores corresponden a pasos sencillos en el proceso modelado como el aprendizaje de un modelo de predicción ó la aplicación de un filtro de preprocesamiento. Los nodos interiores del árbol corresponden a más complejos o abstractos pasos en el proceso. Este es a menudo necesario si los hijos deben ser aplicados varias veces como, por ejemplo, los ciclos. En general, los nodos de los operadores internos definen el flujo de datos a través de sus hijos. La raiz del árbol corresponde al experimento completo.

Qué Puede Hacer YALE?

YALE provee mas de 200 operadores incluyendo:

Algoritmos de aprendizaje maquinal Un gran número de esquemas de aprendizaje para tareas de regresión y clasificación incluyendo Máquinas para el Soporte de Vectores (SVM), árboles de decisión e inductores de reglas, operadores Lazy, operadores Bayesianos y operadores Logísticos. Varios oper-

adores para la minería de reglas de asociación y Clustering son también parte de YALE. Además, se adicionaron varios esquemas de Meta Aprendizaje.

Operadores WEKA Todos los esquemas de aprendizaje y evaluadores de atributos del ambiente de aprendizaje WEKA también están disponibles y pueden ser usados como cualquier otro operador de YALE.

Capítulo 7

DESARROLLO DEL PROYECTO

7.1. Análisis UML

7.1.1. Funciones

Ref#	**Función**	**Cat.**	**Atributo**	**Detalles y Restricciones**	**Cat.**
R1	Ejecutar la herramienta.	Evidente.	Interfaz.		Obligatorio.
R1.1	Mostrar interfaz gráfica de la herramienta	Evidente.	Interfaz.	Pestañas desplegables.	Obligatorio.
R2	Mostrar mensajes de ayuda, respuesta o error cuando sea necesario.	Evidente.	Interfaz.	Cuadros de dialogo, barra de estado.	Deseable.
R3	Permitir el establecimiento de conexiones.	Evidente.	Interfaz.		Obligatorio.
R3.1	Autorizar conexiones a Bases de Datos.	Evidente.	Interfaz.		Opcional.
R3.1.1	Permitir la selección de atributos.	Evidente.	Interfaz.		Obligatoria.
R3.1.2	Cargar el conjunto de datos.	Oculto.	Información	Establecer relaciones correctas	Obligatorio.
R3.2	Autorizar conexiones a Archivos Planos.	Evidente.	Interfaz.		Opcional.

Ref#	Función	Cat.	Atributo	Detalles y Restricciones	Cat.
R3.2.1	Recorrer archivo y cargar datos.	Oculto.	Información.	El archivo debe cumplir con el formato ARFF.	Obligatorio.
R4	Permitir la aplicación de filtros	Evidente	Interfaz		Opcional
R4.1	Permitir remover datos nulos	Evidente	Interfaz		Opcional
R4.2	Permitir actualizar datos nulos	Evidente	Interfaz		Opcional
R4.3	Permitir seleccionar un conjunto de registros	Evidente	Interfaz		Opcional
R4.4	Permitir seleccionar un conjunto de registros según un atributo dado.	Evidente	Interfaz		Opcional
R4.5	Permitir reducir el rango de los datos	Evidente	Interfaz		Opcional
R4.6	Permitir codificar los datos	Evidente	Interfaz		Opcional
R4.7	Permitir reemplazar un valor determinado	Evidente	Interfaz		Opcional
R4.8	Permitir seleccionar una muestra del conjunto de datos	Evidente	Interfaz		Opcional
R4.9	Permitir discretizar valores continuos	Evidente	Interfaz		Opcional
R5	Aplicar técnicas de Minería de Datos	Evidente	Interfaz		Obligatorio
R5.1	Proveer algoritmos que cumplan tareas de asociación	Evidente	Interfaz		Opcional
R5.1.1	Implementar algoritmo Apriori	Evidente	Interfaz		Opcional
R5.1.2	Implementar algoritmo FPGrowth	Evidente	Interfaz		Opcional
R5.1.3	Implementar algoritmo EquipAsso	Evidente	Interfaz		Opcional

Ref#	Función	Cat.	Atributo	Detalles y Restricciones	Cat.
R5.2	Proveer algoritmos que cumplan tareas de clasificación	Evidente	Interfaz		Opcional
R5.2.1	Implementar algoritmo C4.5	Evidente	Interfaz		Opcional
R5.2.2	Implementar algoritmo MateBy	Evidente	Interfaz		Opcional
R6	Permitir visualización de reglas	Evidente	Interfaz		Obligatorio
R6.1	Organizar las reglas de asociación a través de una tabla	Evidente	Interfaz		Obligatorio
R6.2	Organizar las reglas de clasificación a través de un árbol	Evidente	Interfaz		Obligatorio

7.1.2. Diagramas de Casos de Uso

Tariy

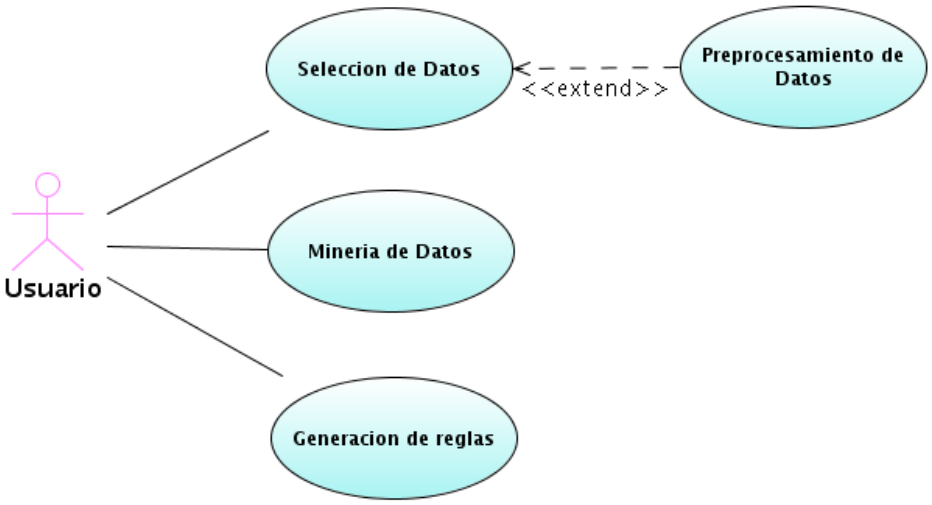

Figura 7.1: Diagrama de caso de uso Tariy

Módulo de Selección

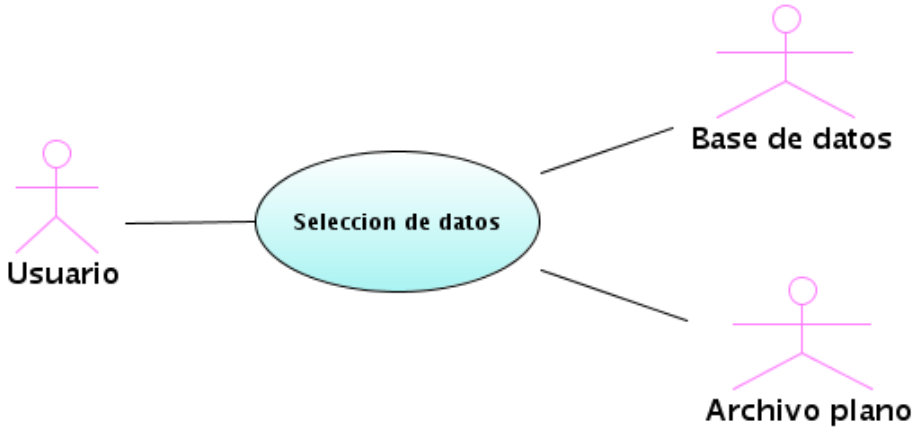

Figura 7.2: Módulo de Selección

Módulo de Conexión a Base de Datos

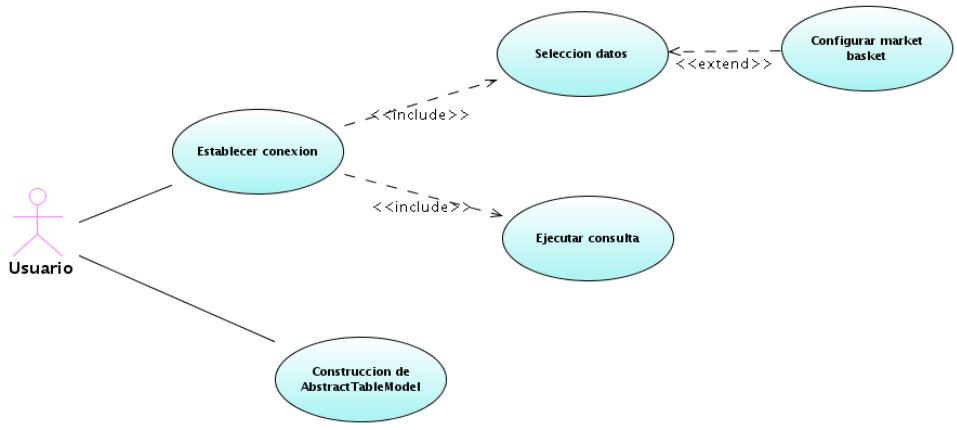

Figura 7.3: Base de Datos

Módulo de Conexión a Archivo Plano

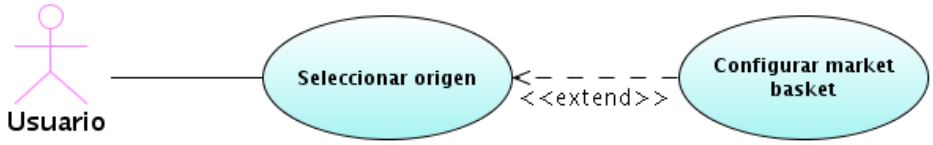

Figura 7.4: Archivo Plano

Módulo de Preprocesamiento

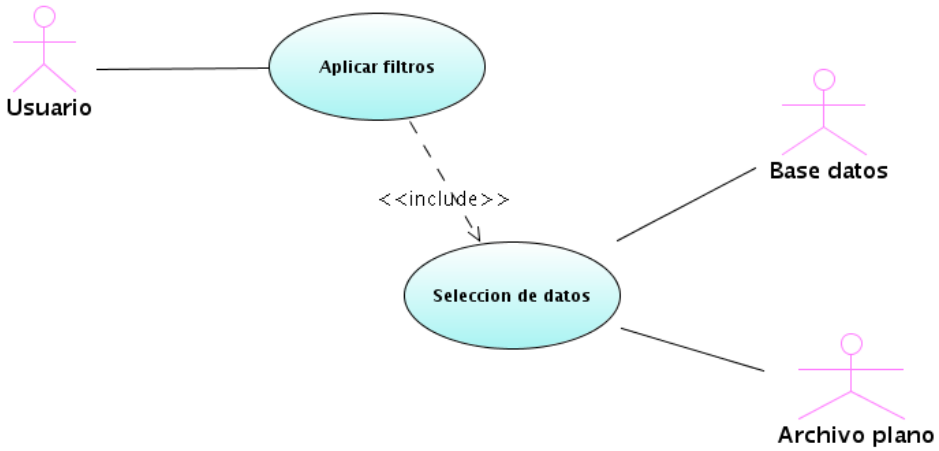

Figura 7.5: Preprocesamiento

Módulo de Minería de Datos

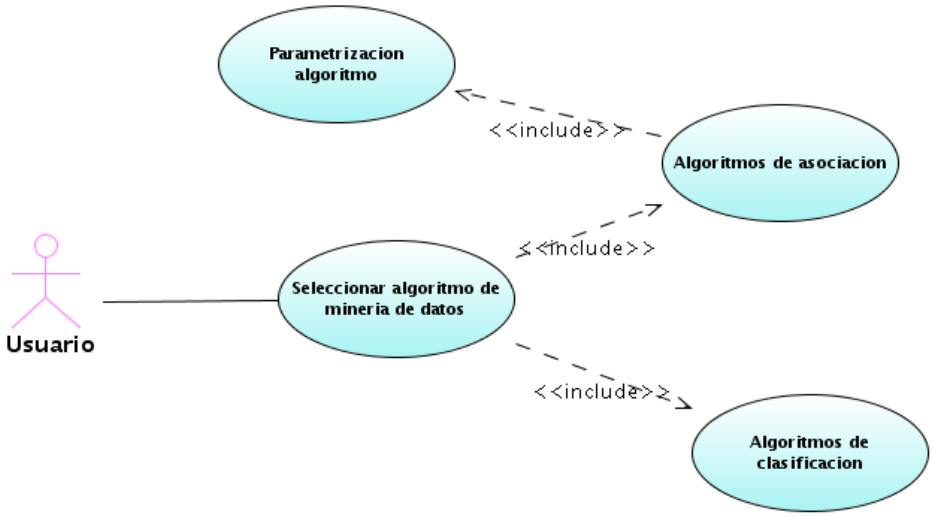

Figura 7.6: Minería de Datos

Módulo de Reglas

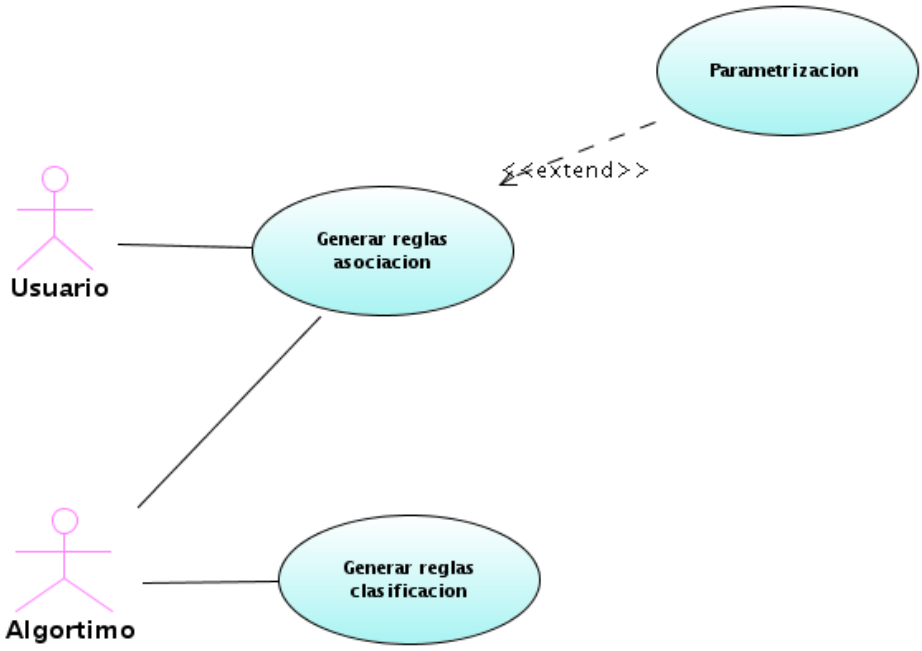

Figura 7.7: Reglas

7.1.3. Diagramas de Secuencia

Clase Apriori

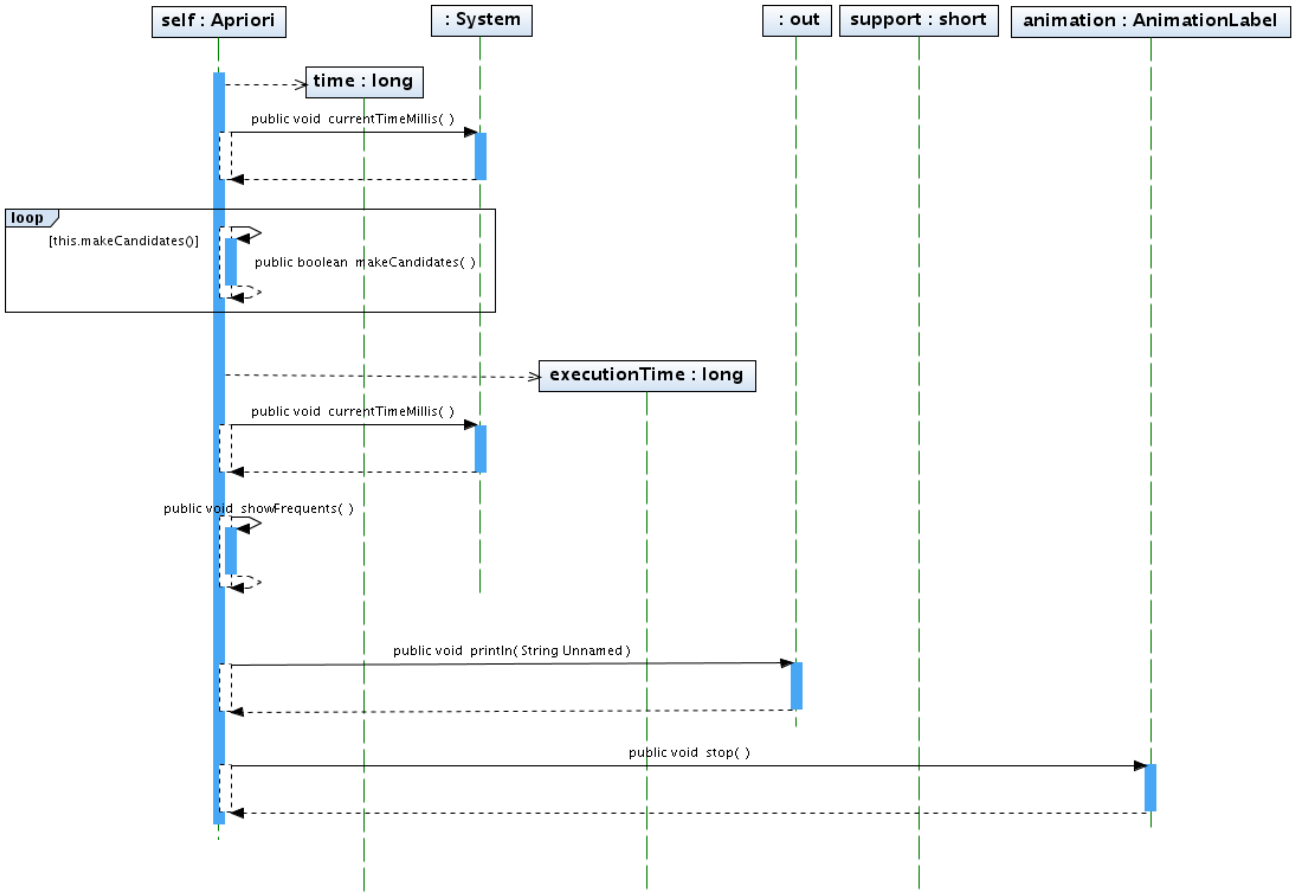

Figura 7.8: run

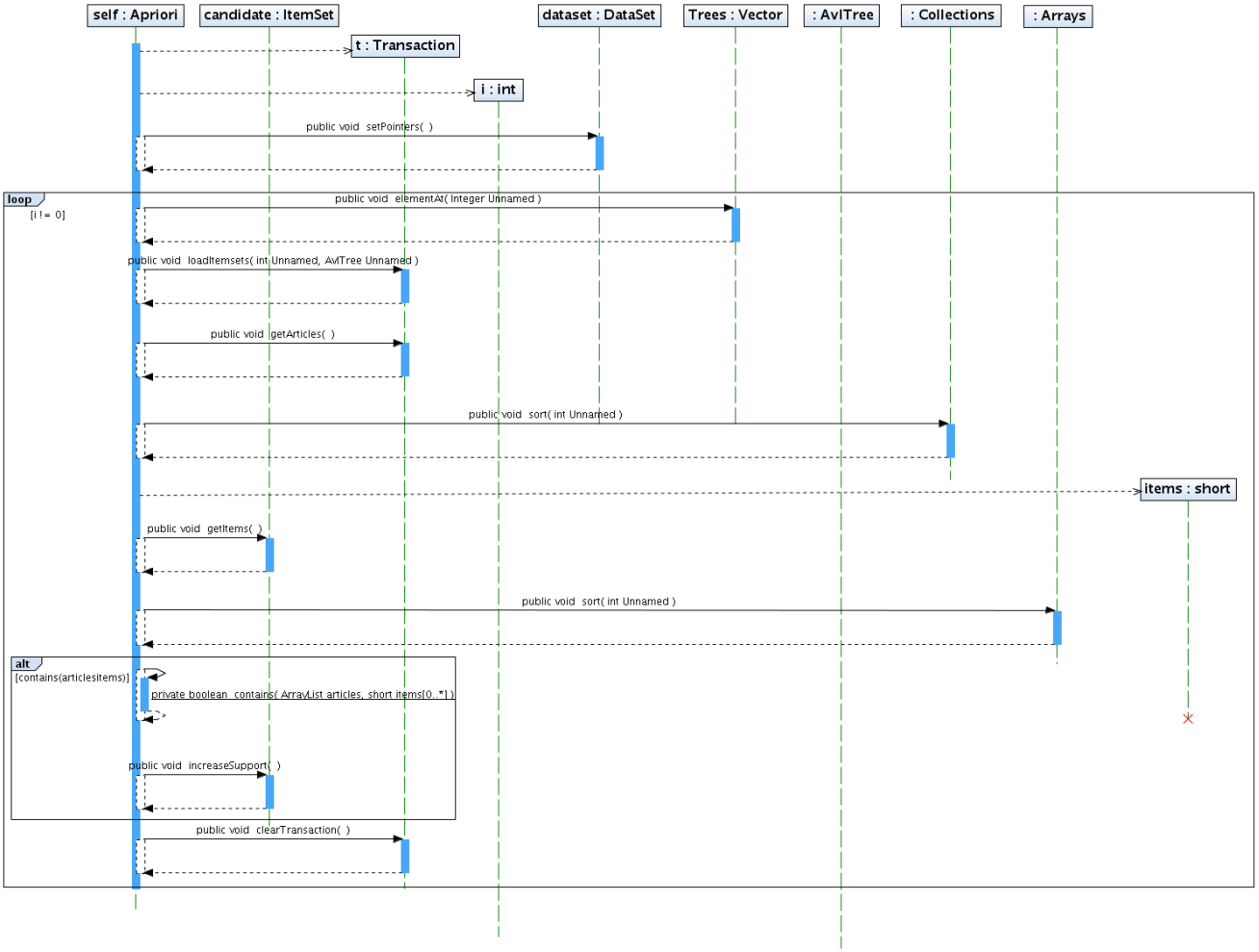

Figura 7.9: increaseSupport

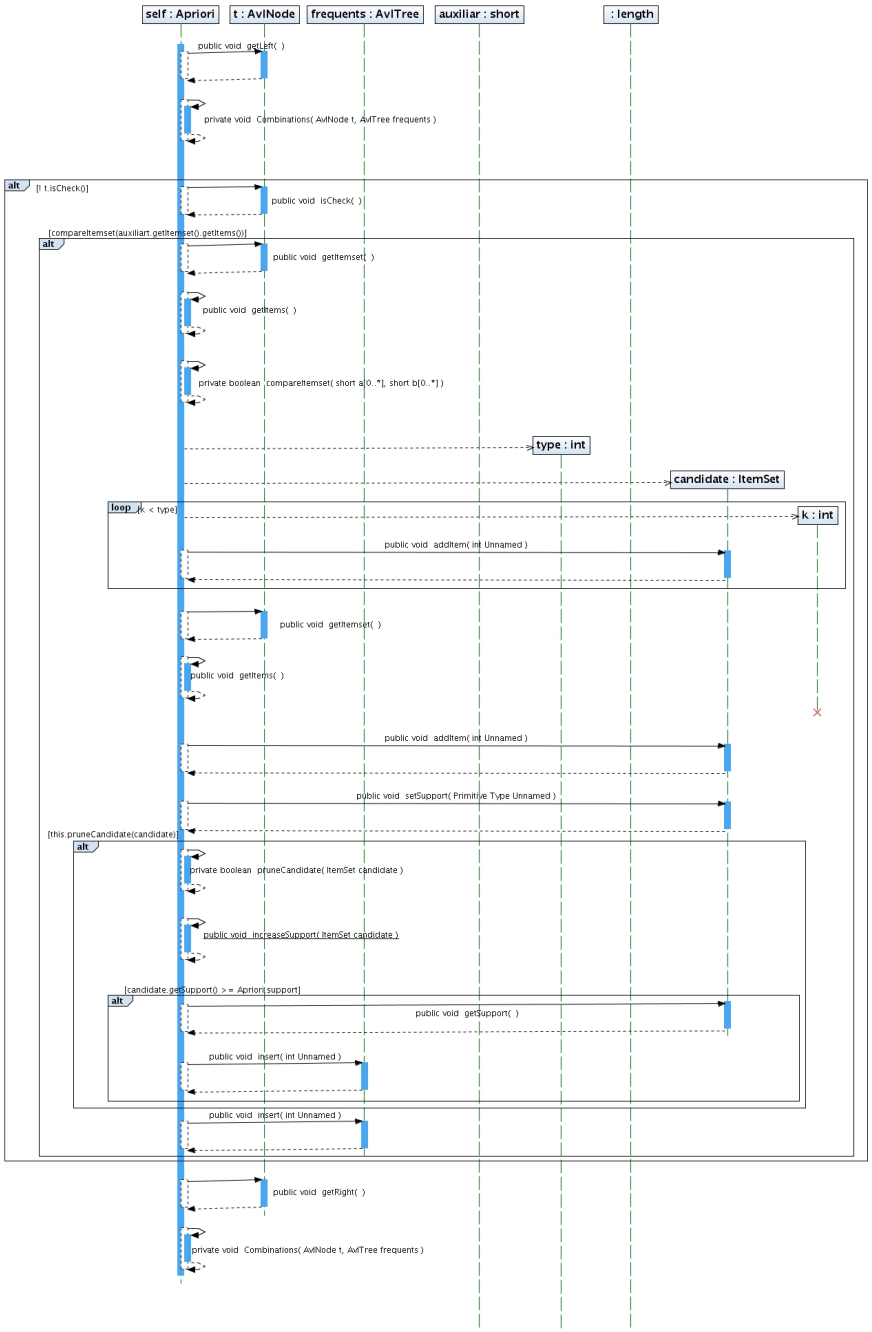

Figura 7.10: Combinations

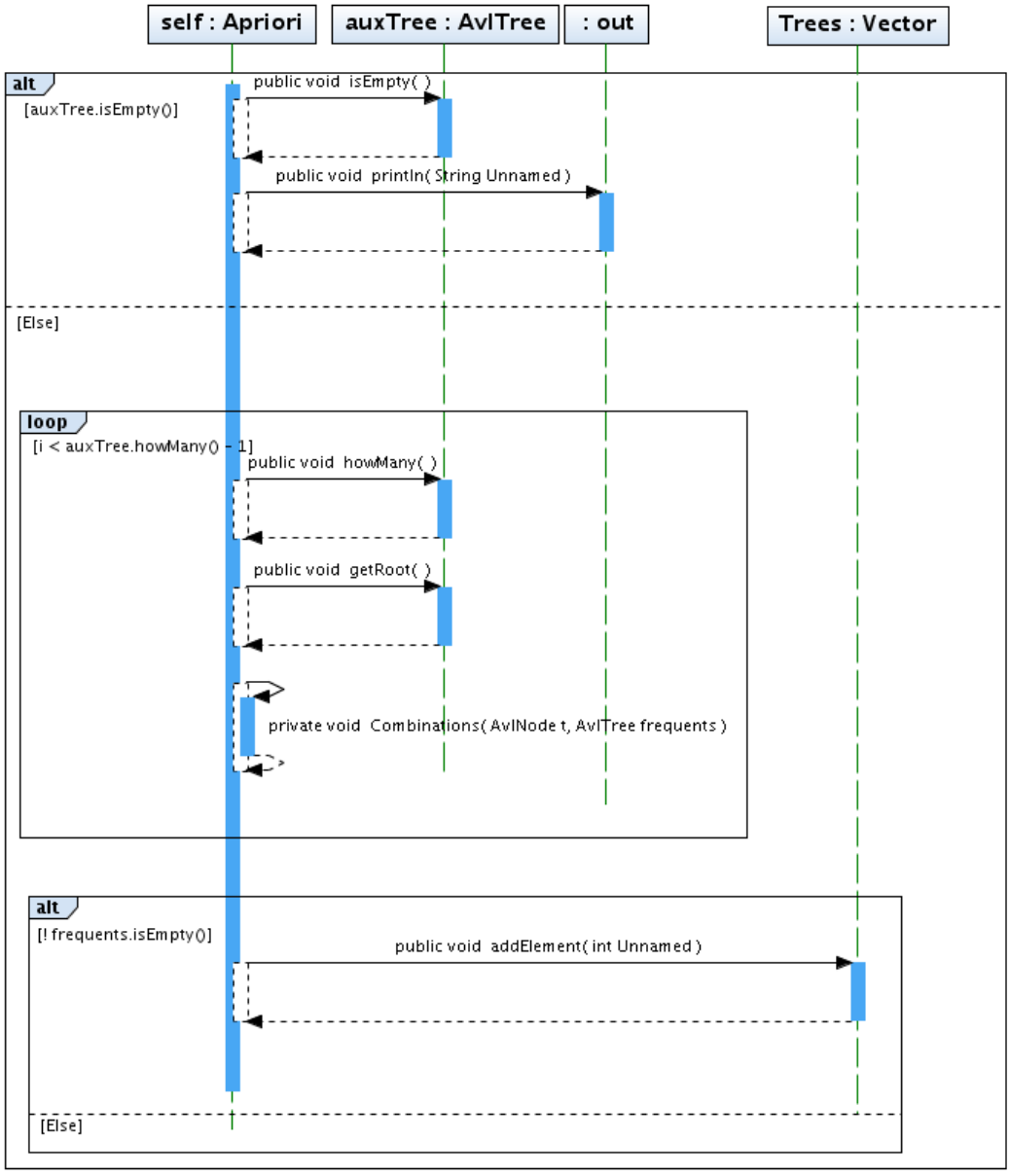

Figura 7.11: makeCandidates

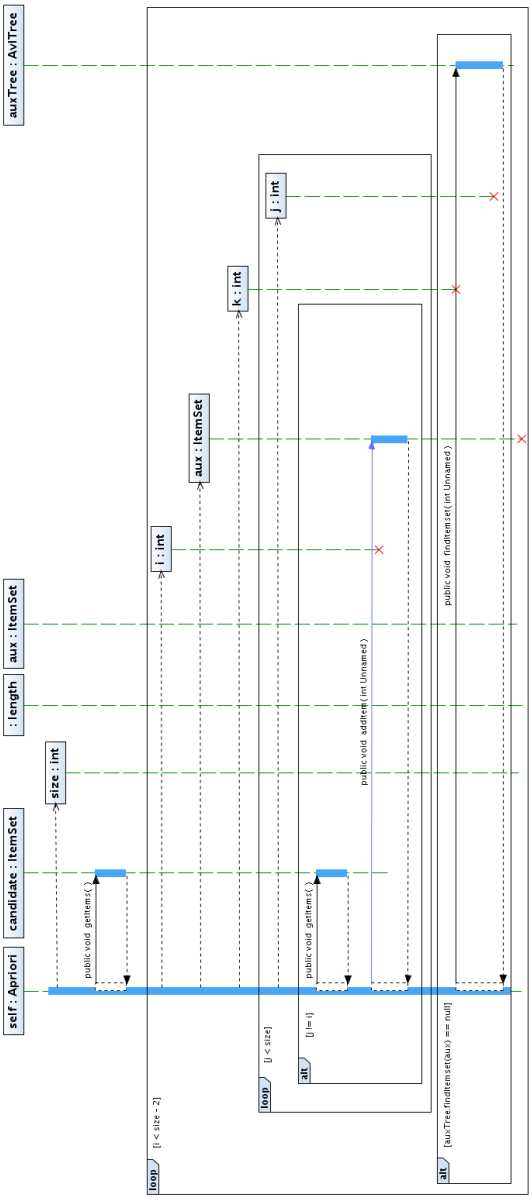

Figura 7.12: pruneCandidates

Clase EquipAsso

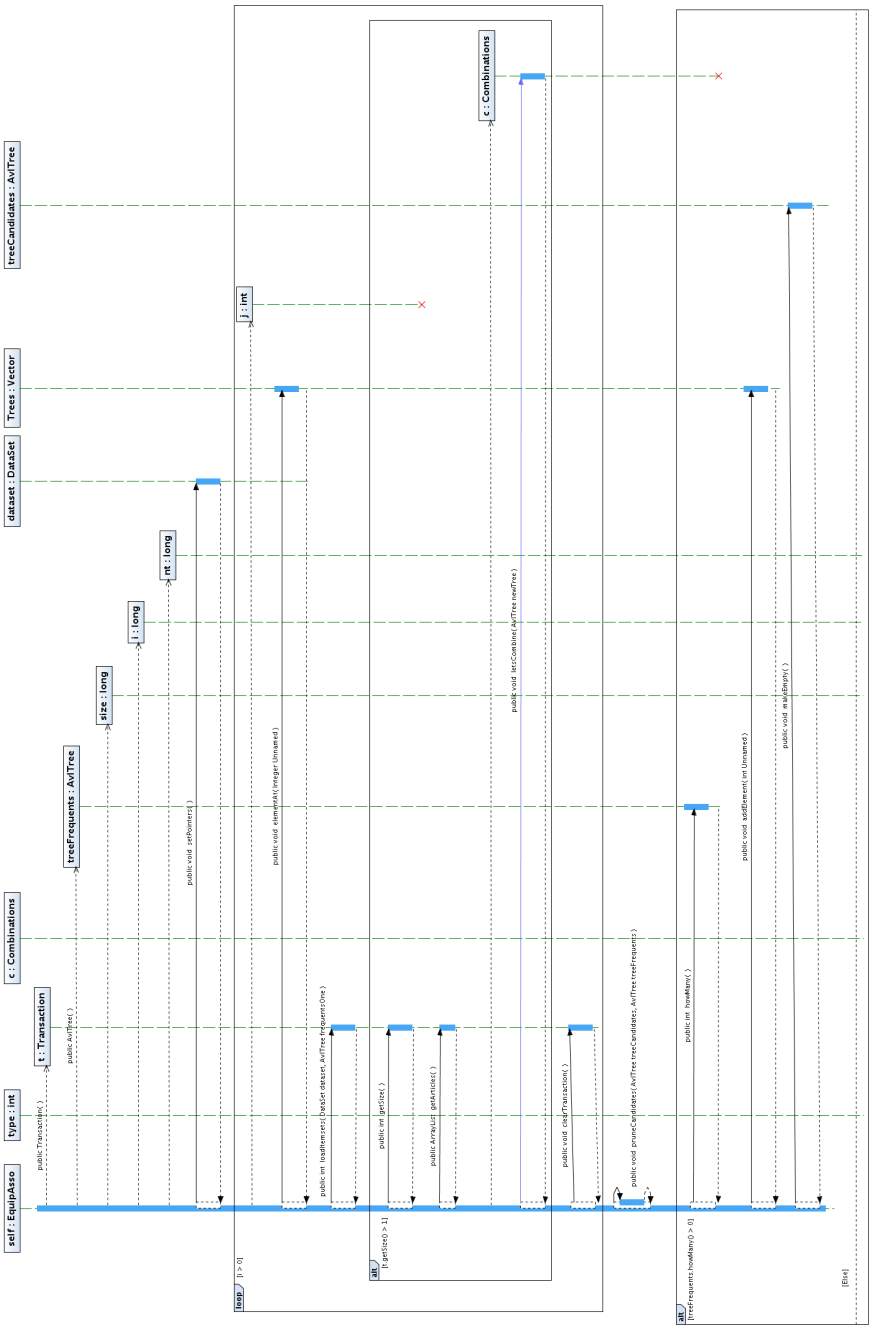

Figura 7.13: findInDataSet

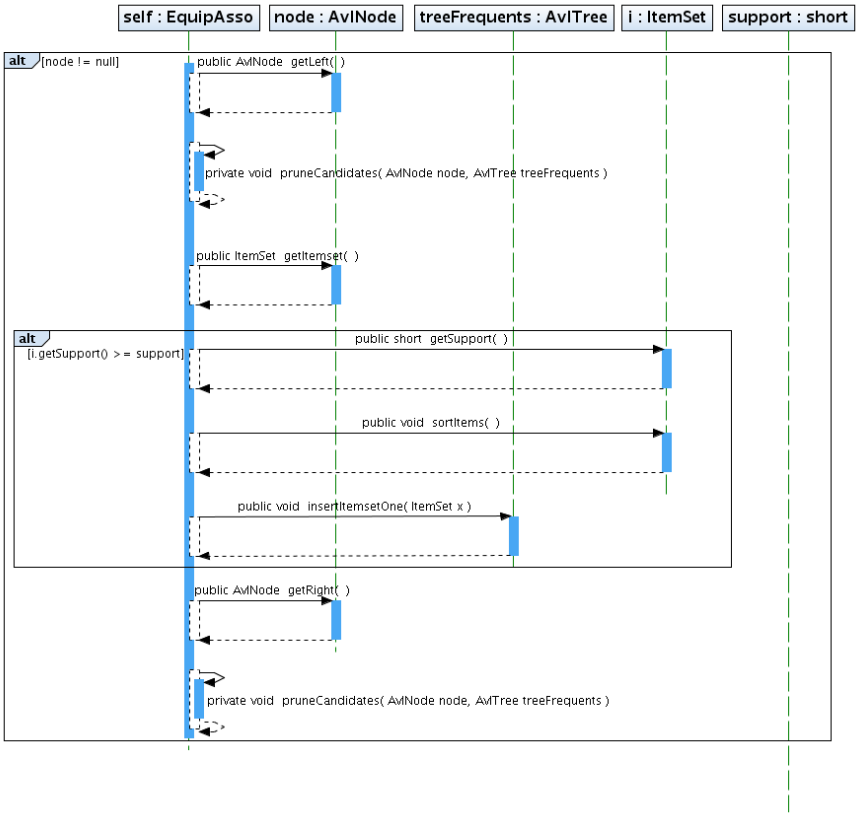

Figura 7.14: pruneCandidate-recursive

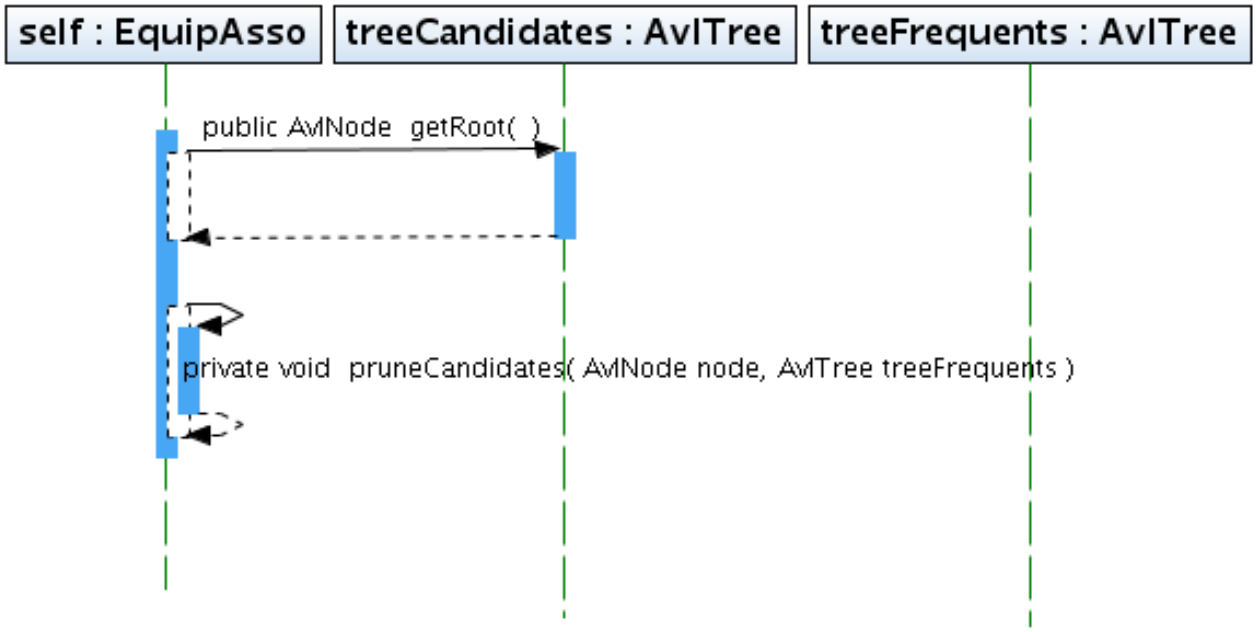

Figura 7.15: pruneCandidate-recursive

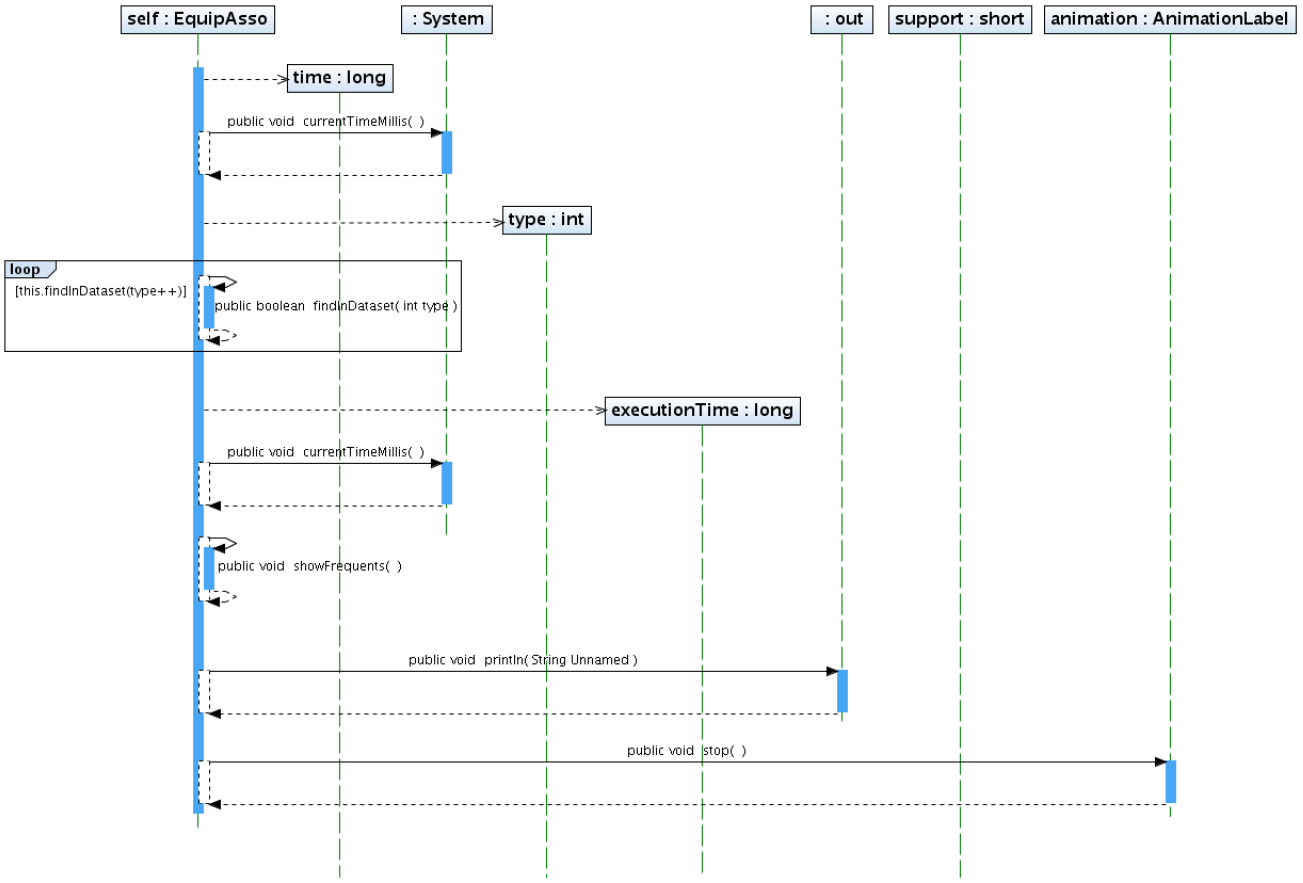

Figura 7.16: run

Clase ItemSet

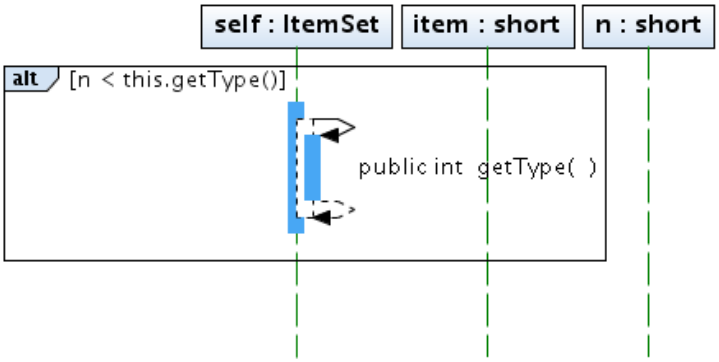

Figura 7.17: addItems

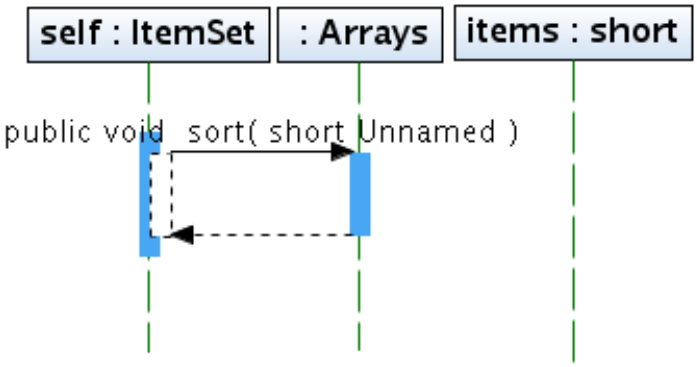

Figura 7.18: sortItems

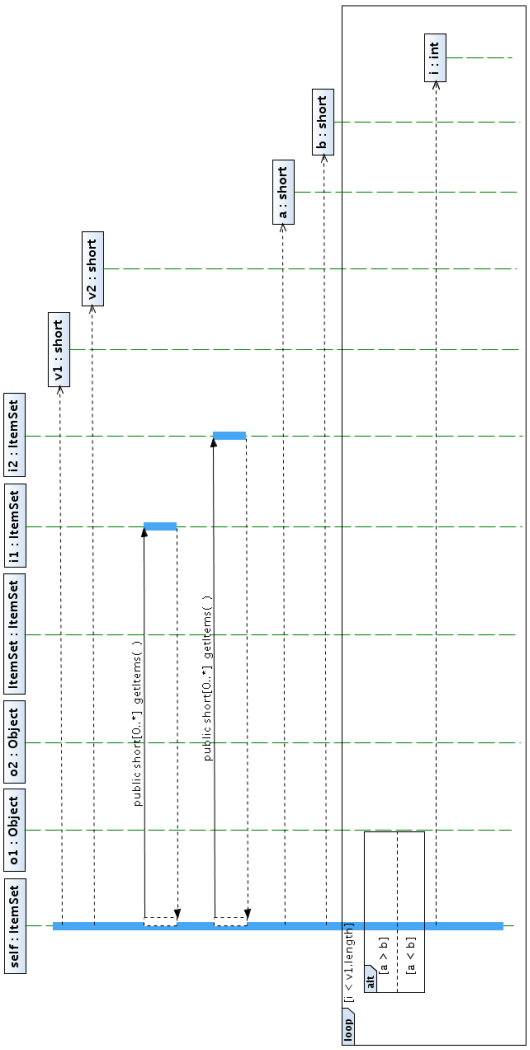

Figura 7.19: compare

Clase DataSet

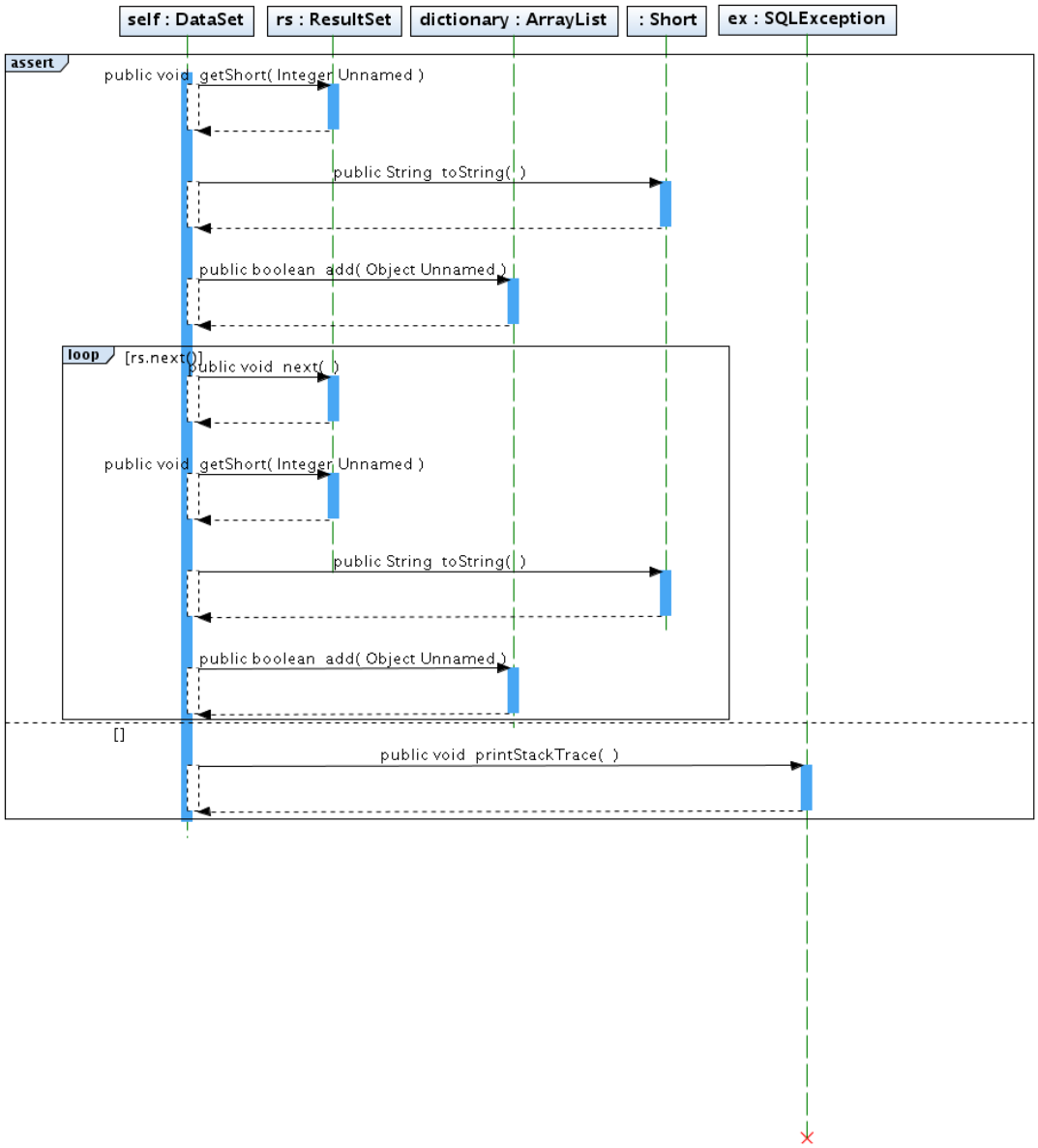

Figura 7.20: buildDictionary

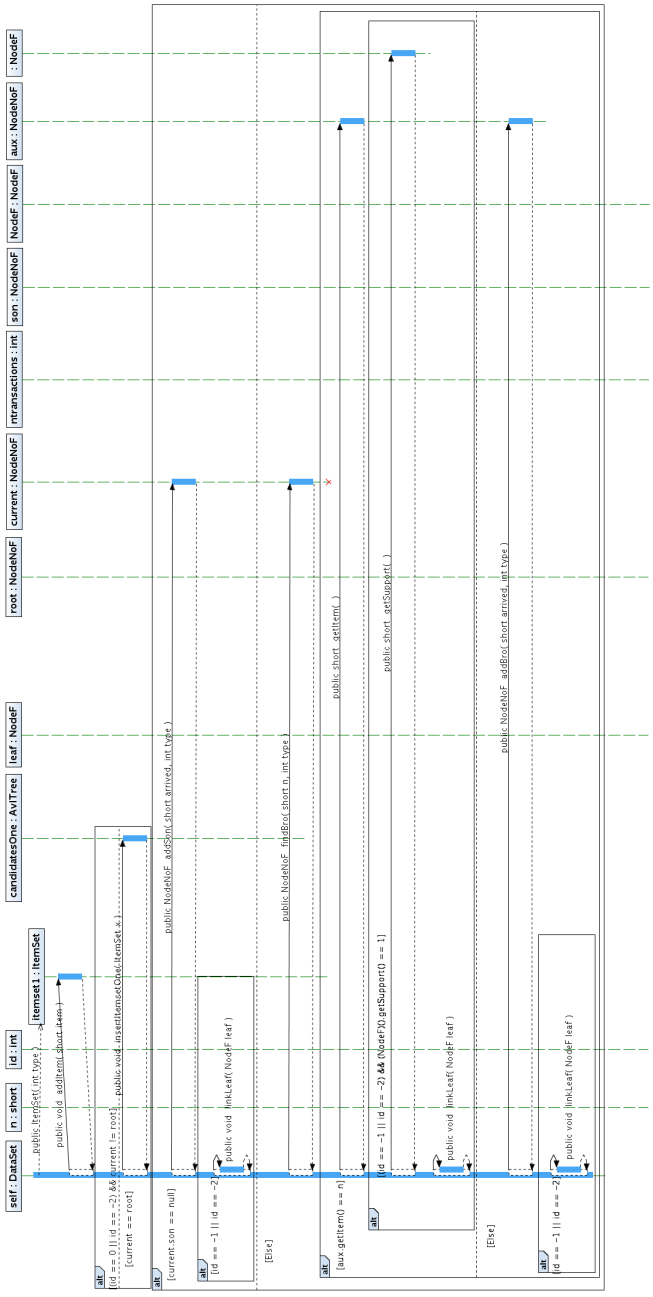

Figura 7.21: buildNTree

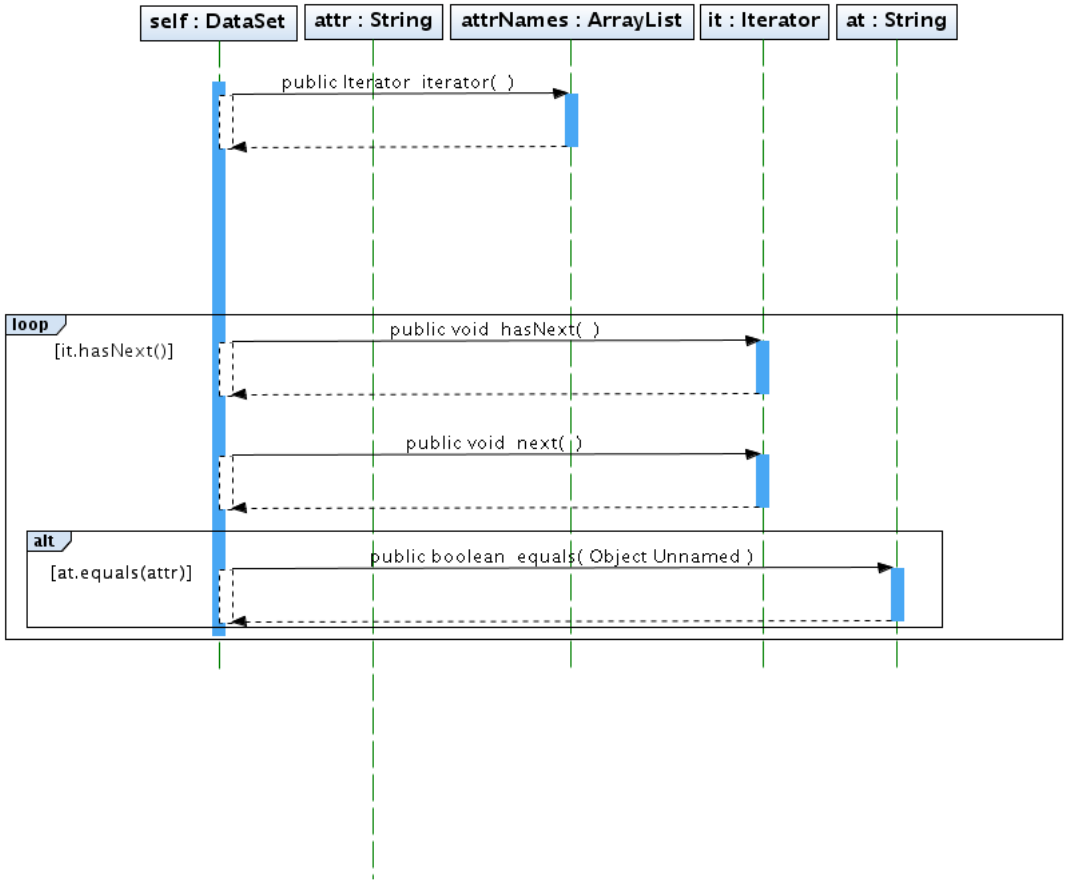

Figura 7.22: findAttrName

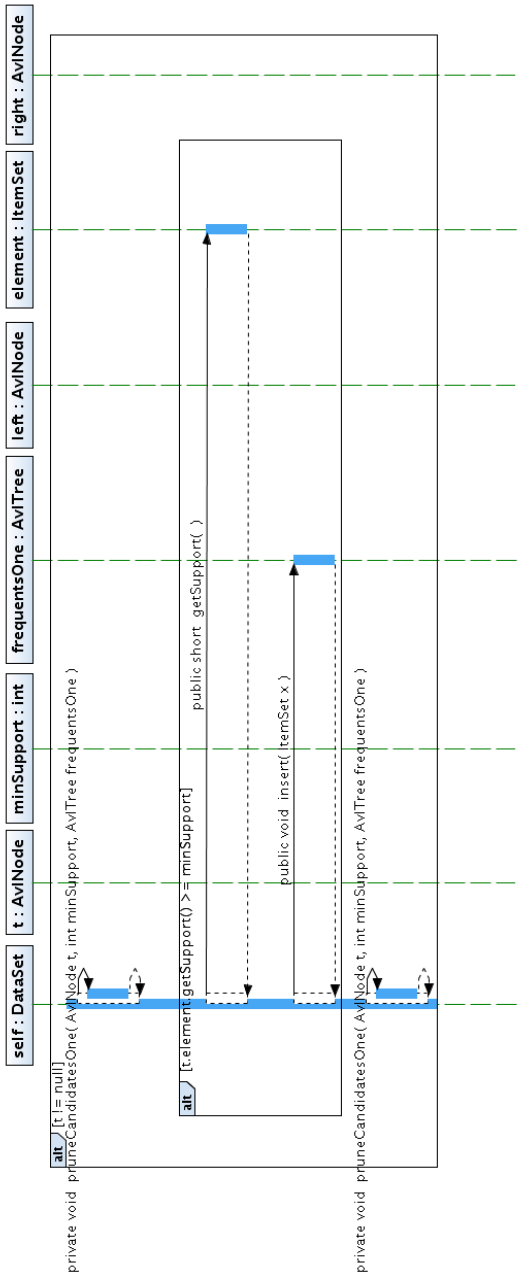

Figura 7.23: pruneCandidatesOne

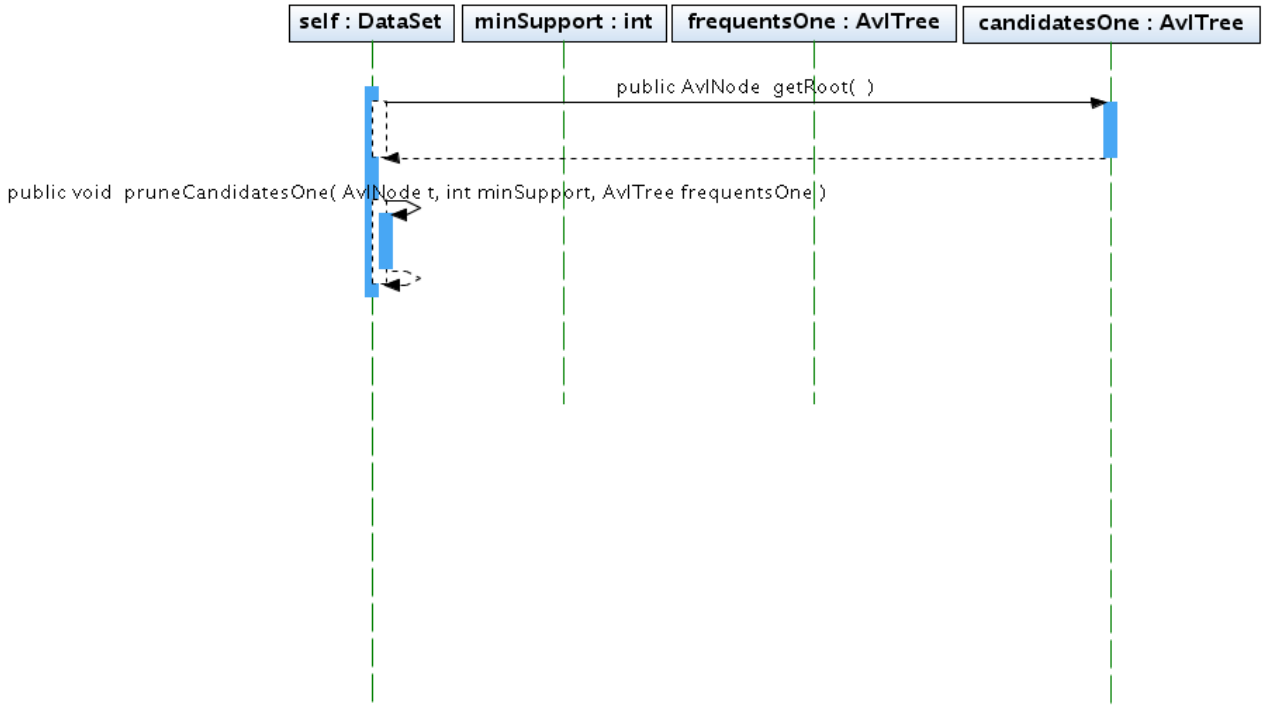

Figura 7.24: public-pruneCandidatesOne

Clase FPGrowth

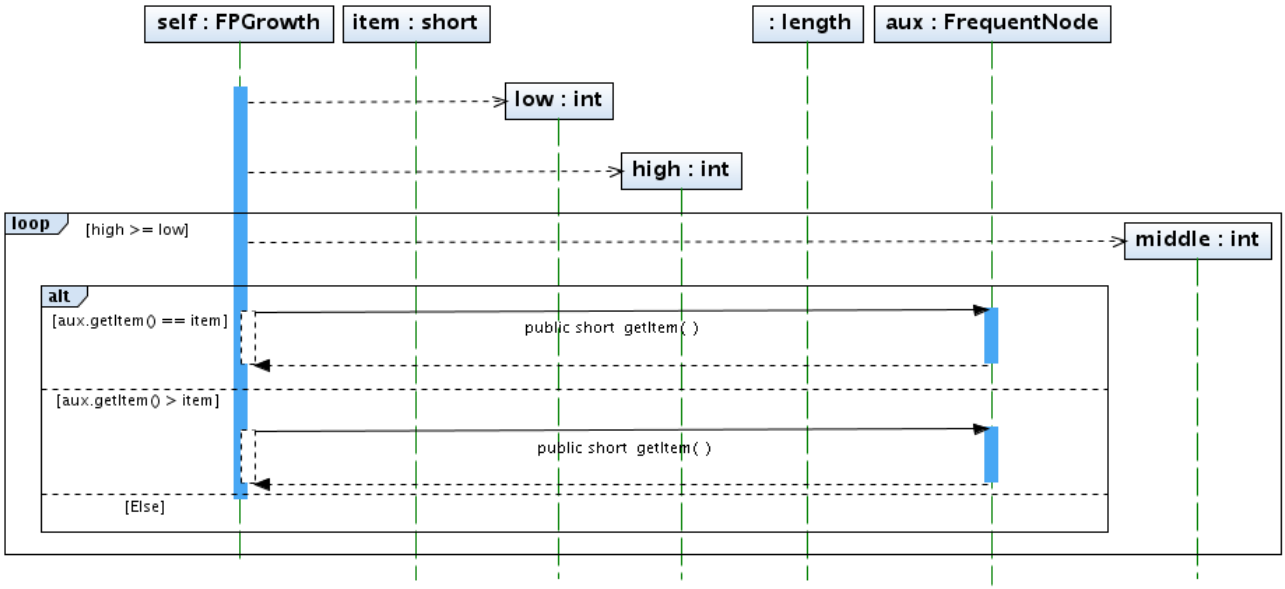

Figura 7.25: buildFrequentsNodes

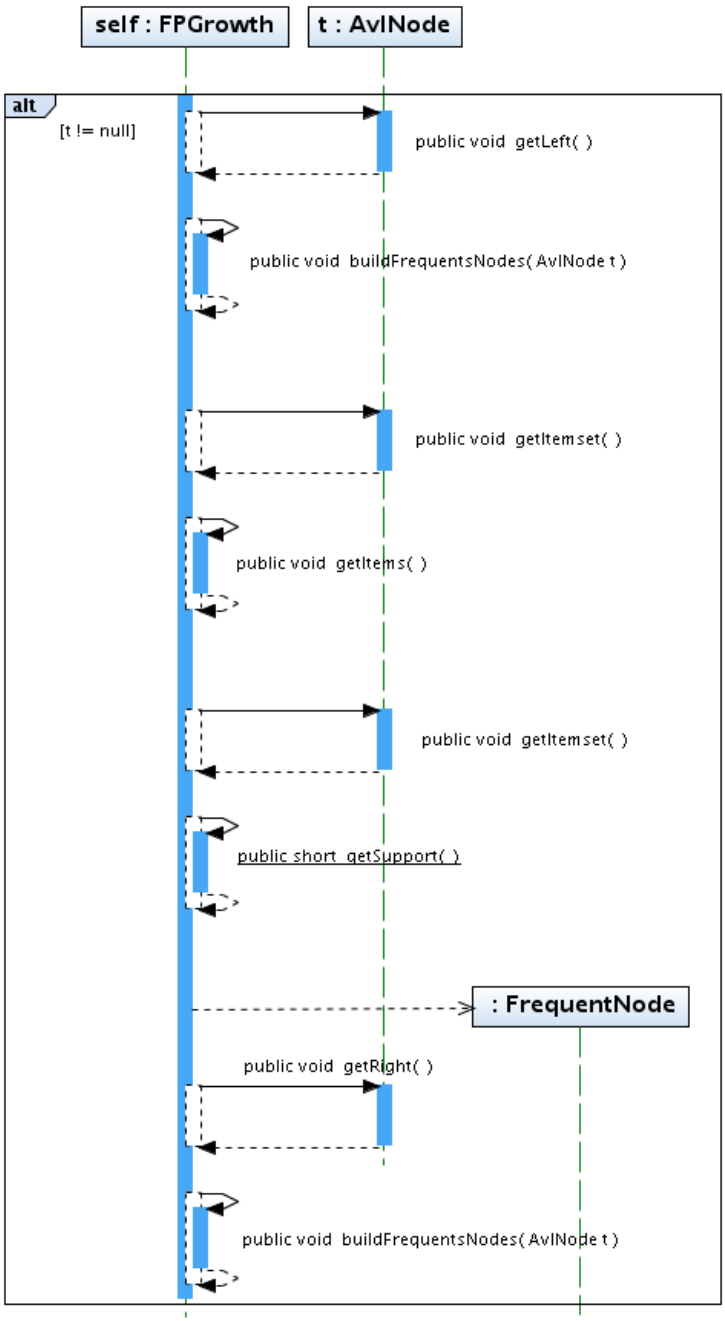

Figura 7.26: findNode

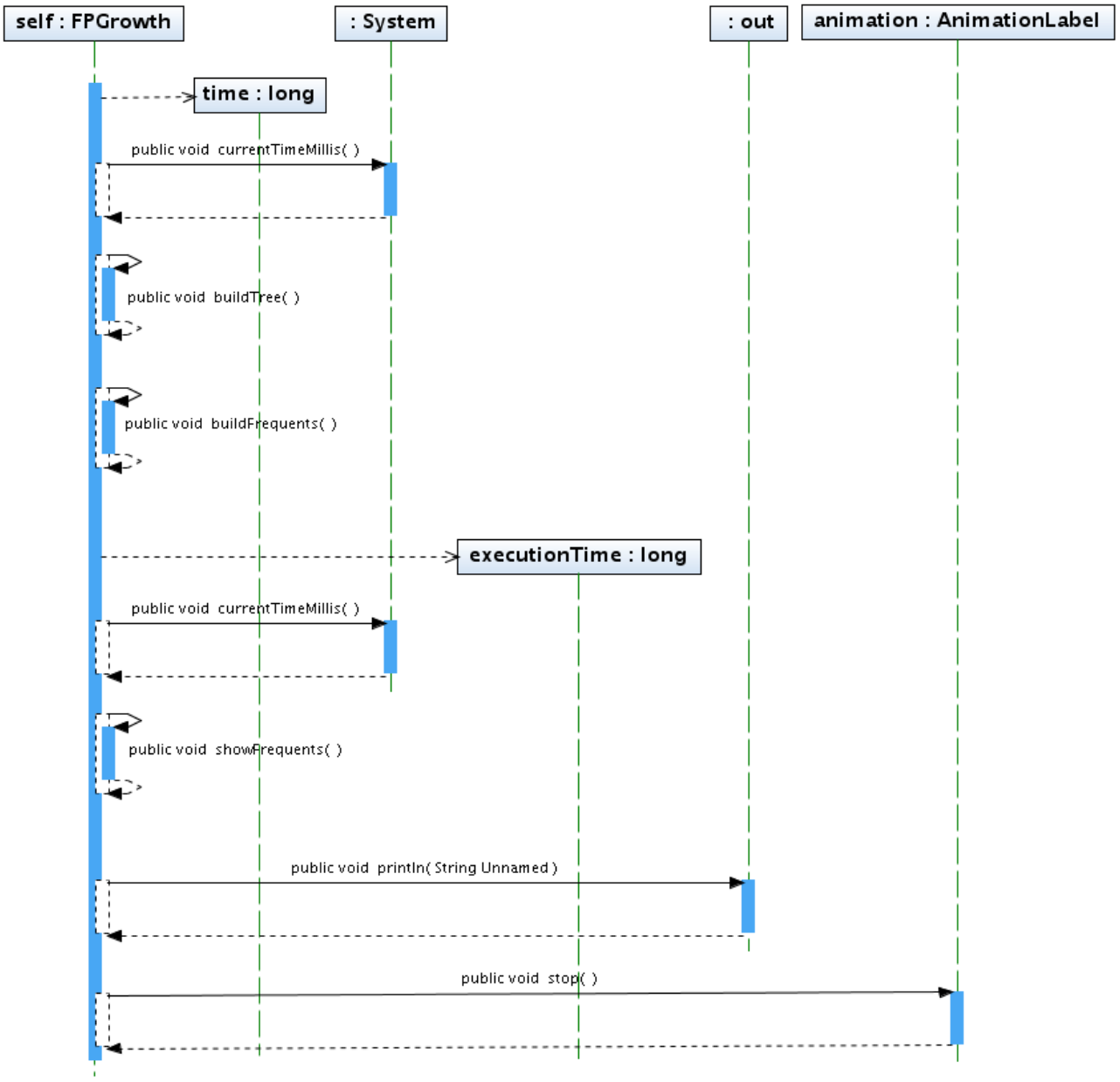

Figura 7.27: run

Clase AvlTree

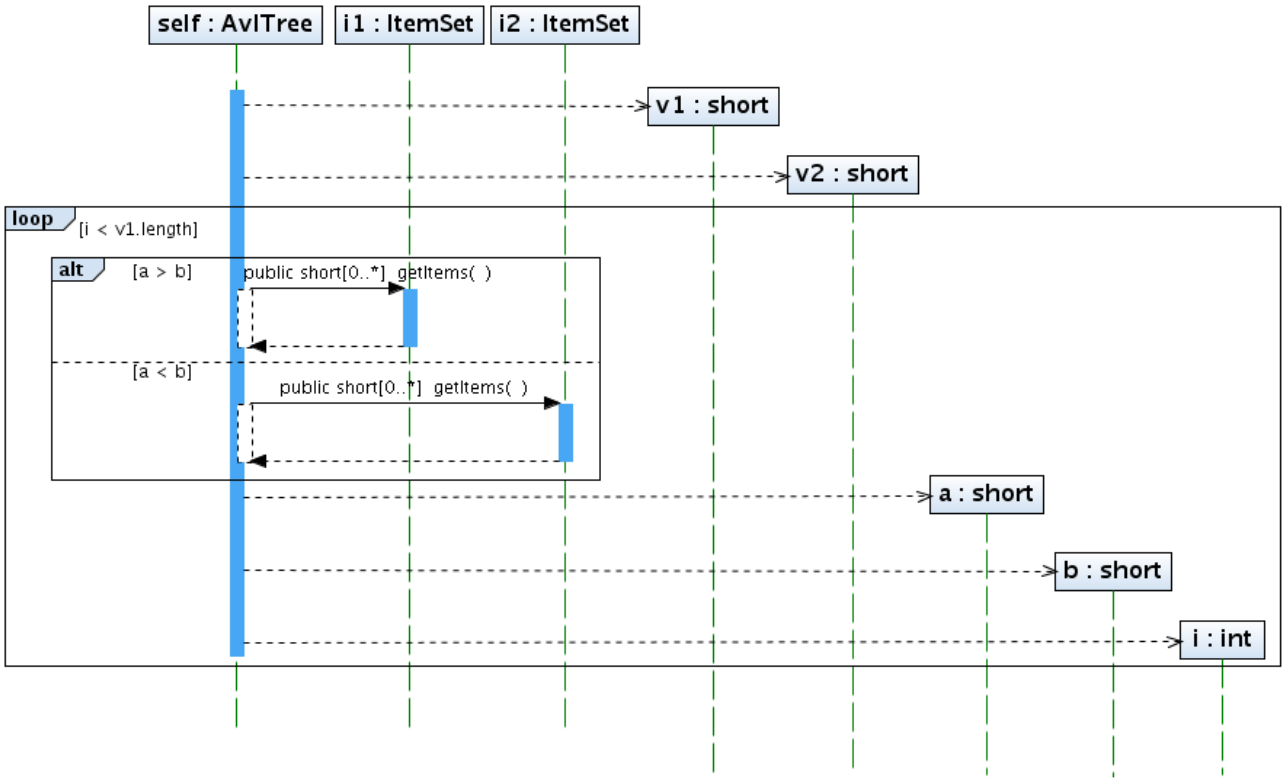

Figura 7.28: compareItemSet

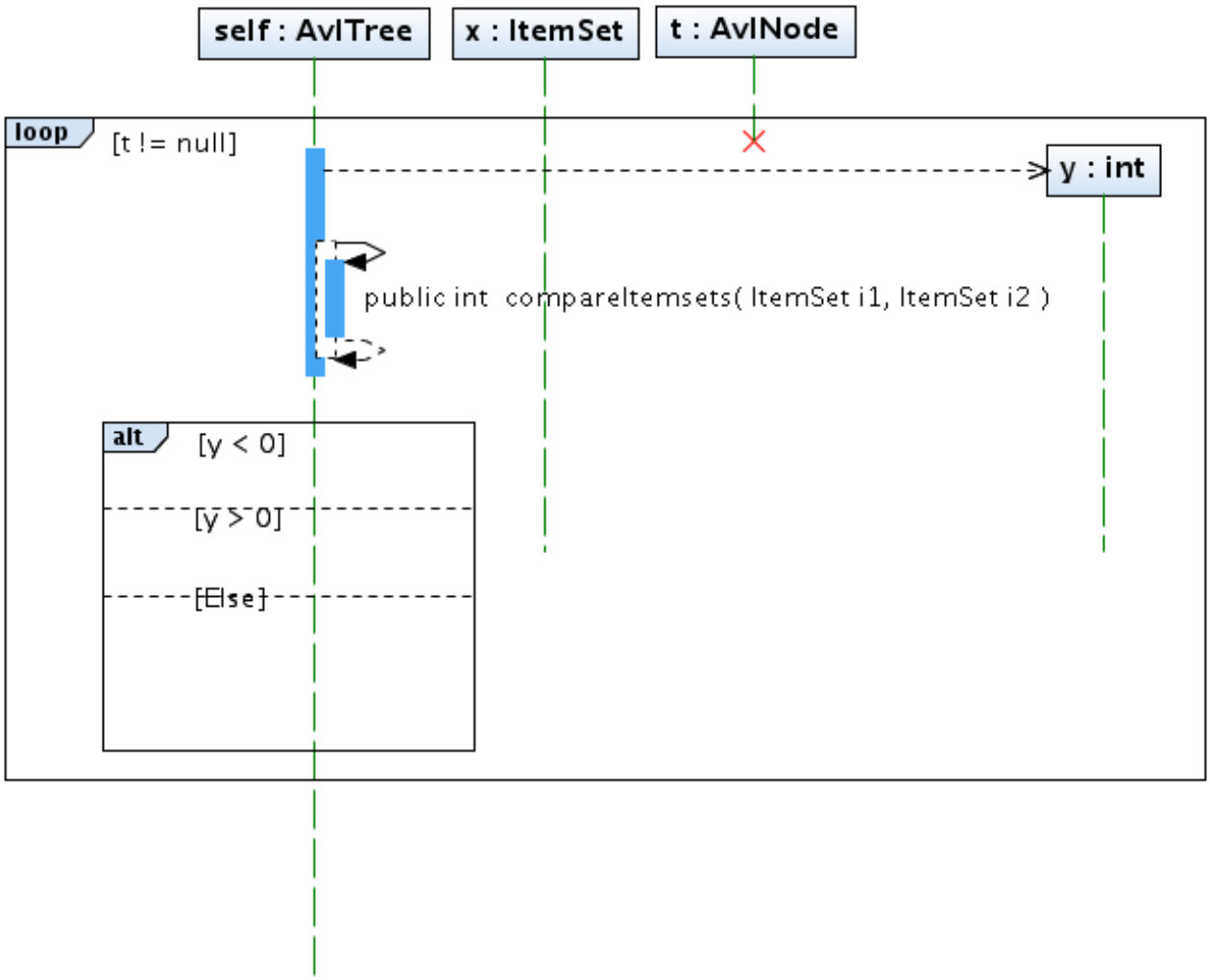

Figura 7.29: find

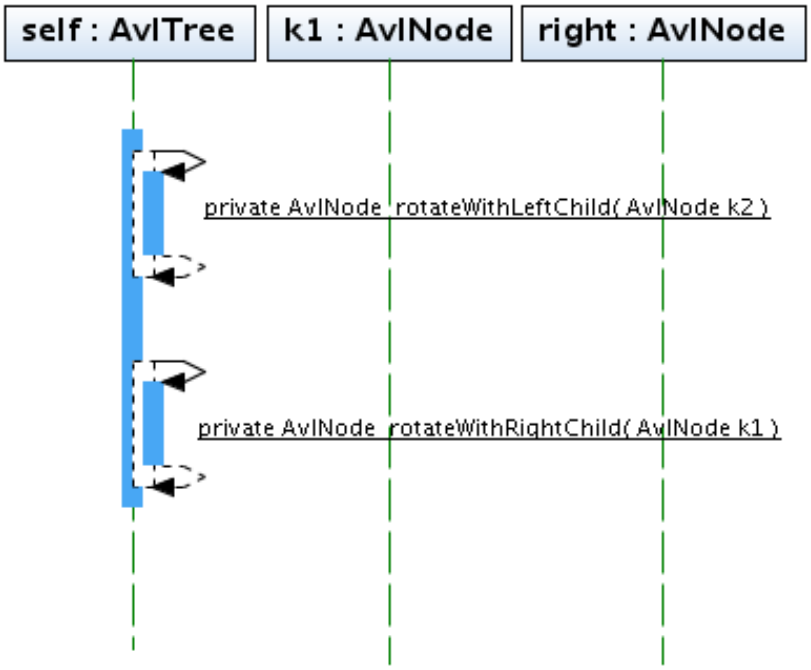

Figura 7.30: doubleWithRightChild

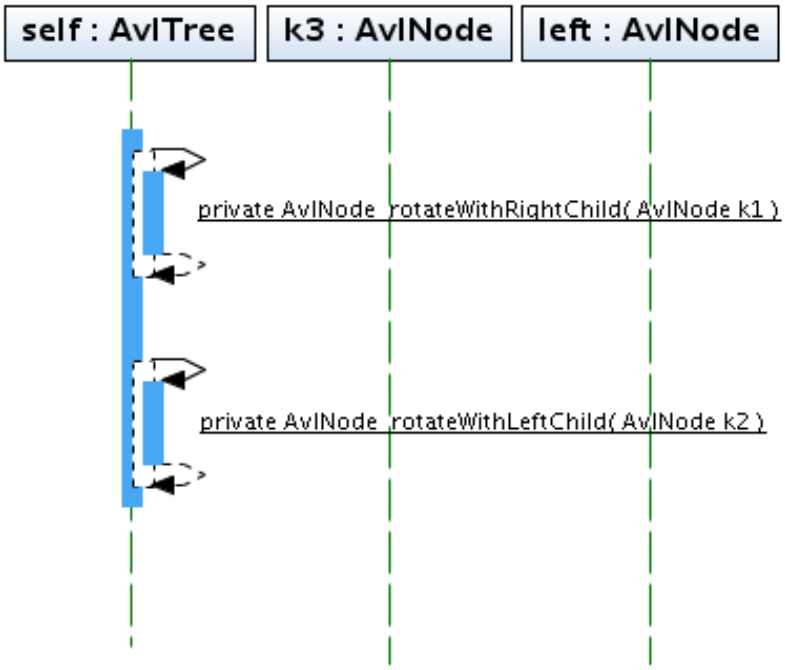

Figura 7.31: doubleWithLeftChild

Clase Transaction

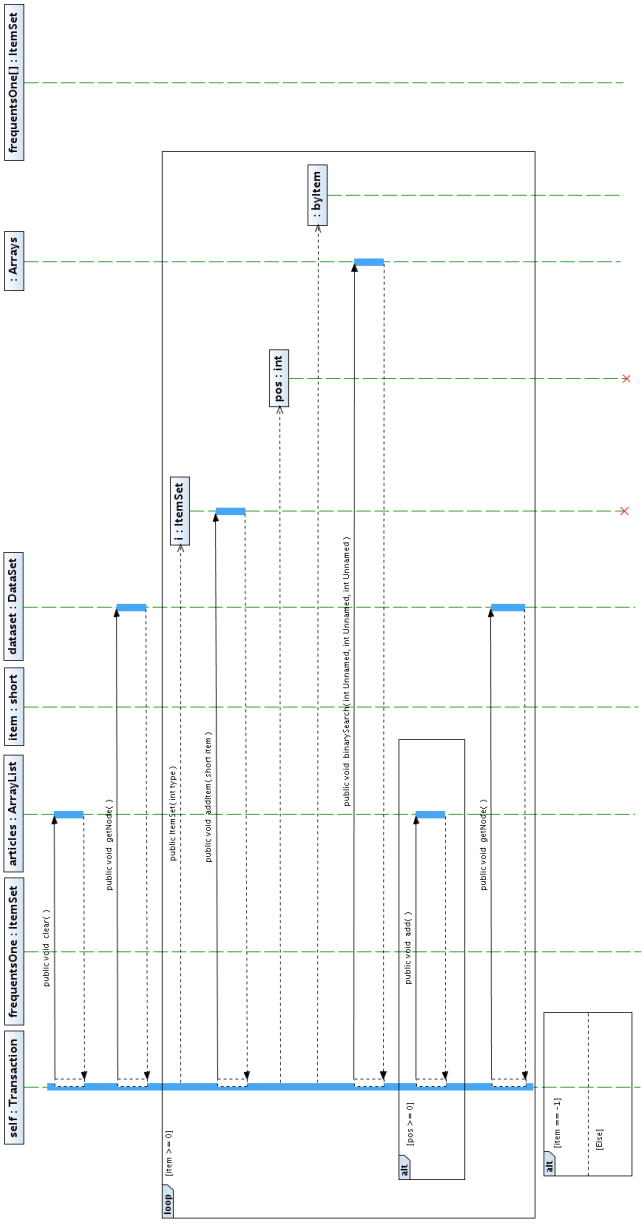

Figura 7.32: loadItemset

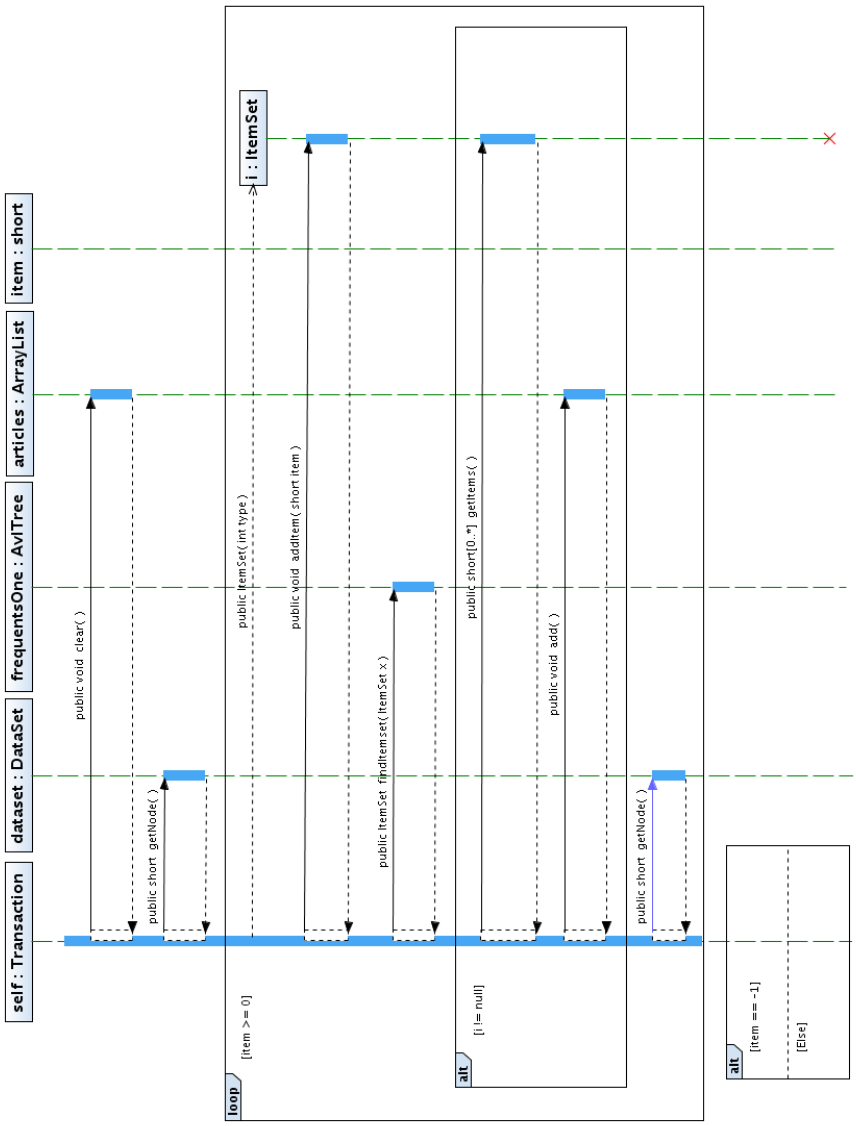

Figura 7.33: public-loadItemset

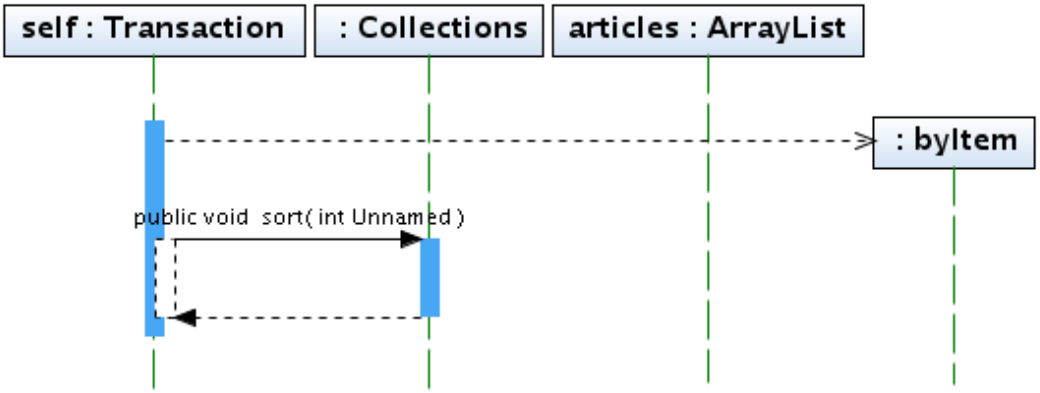

Figura 7.34: sortByItem

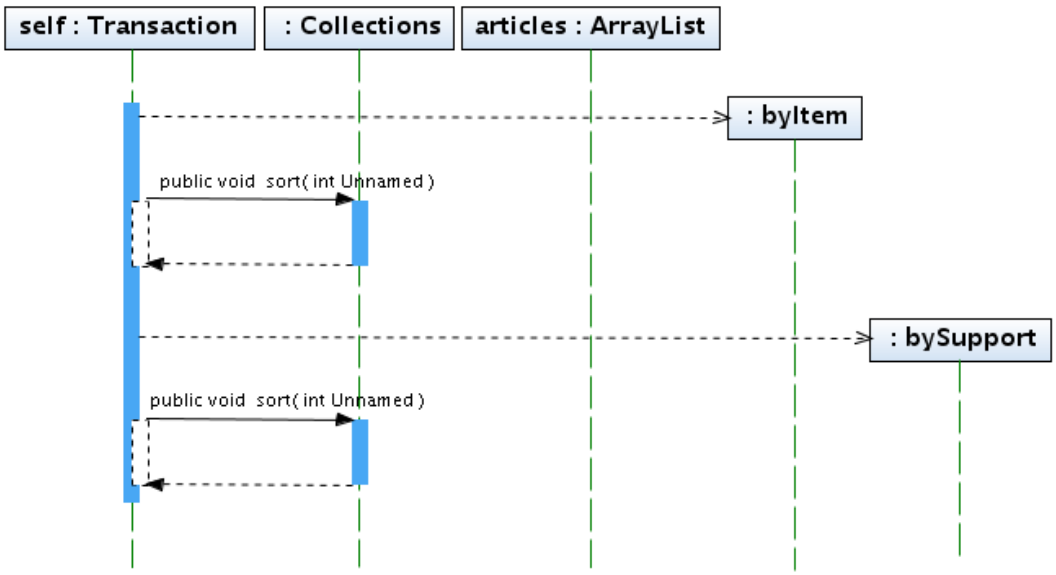

Figura 7.35: sortBySupport

Clase NodeF

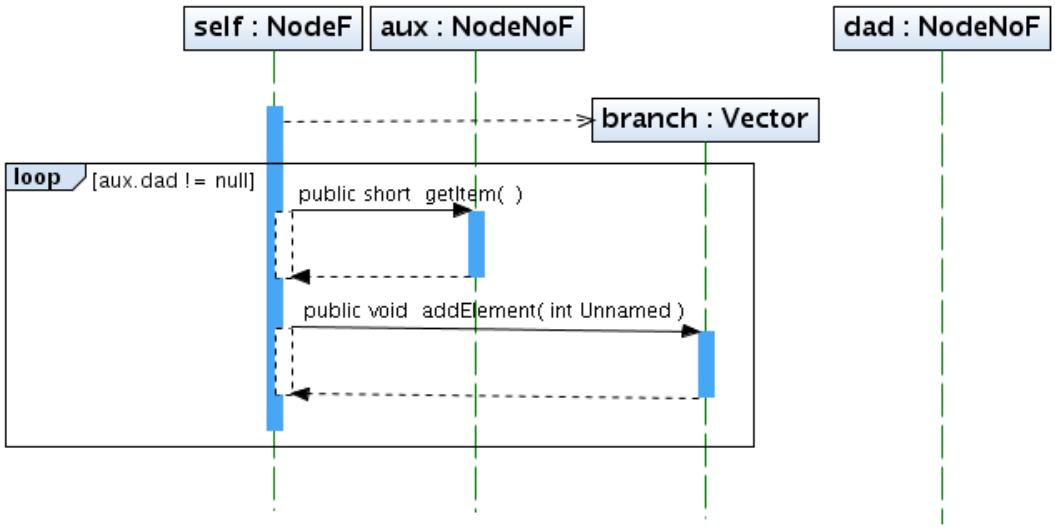

Figura 7.36: getBranch

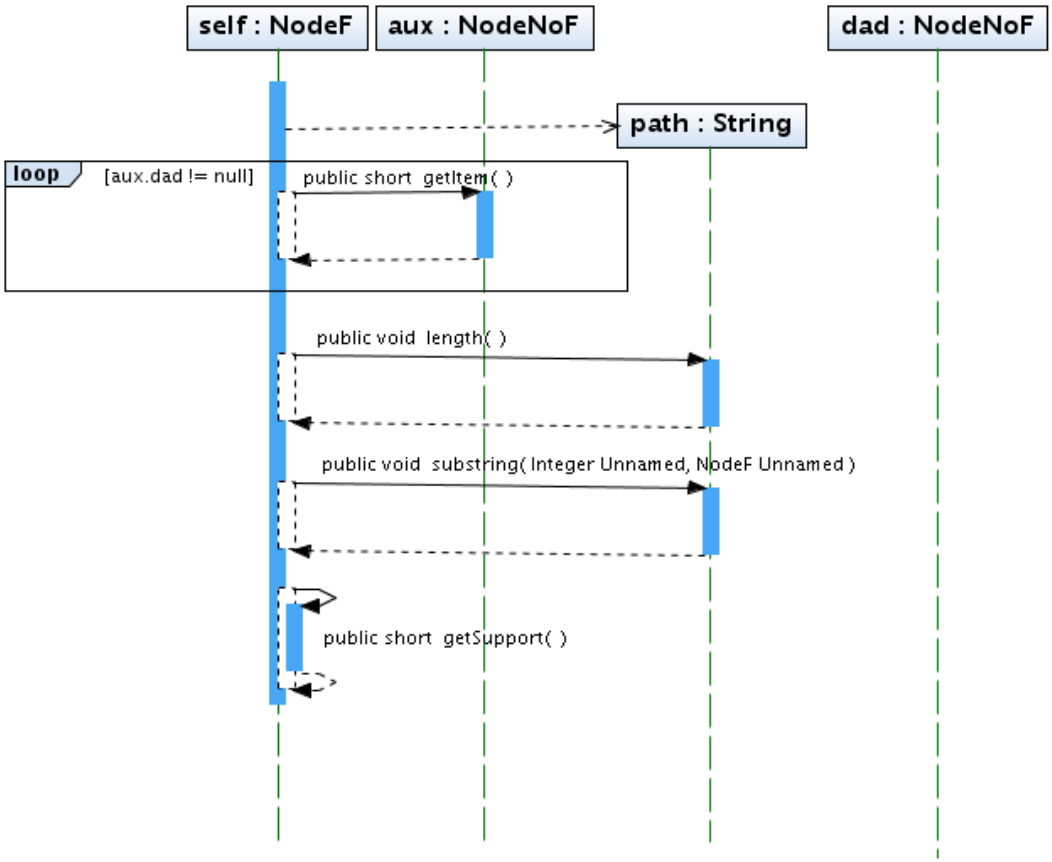

Figura 7.37: getPath

Clase NodeNoF

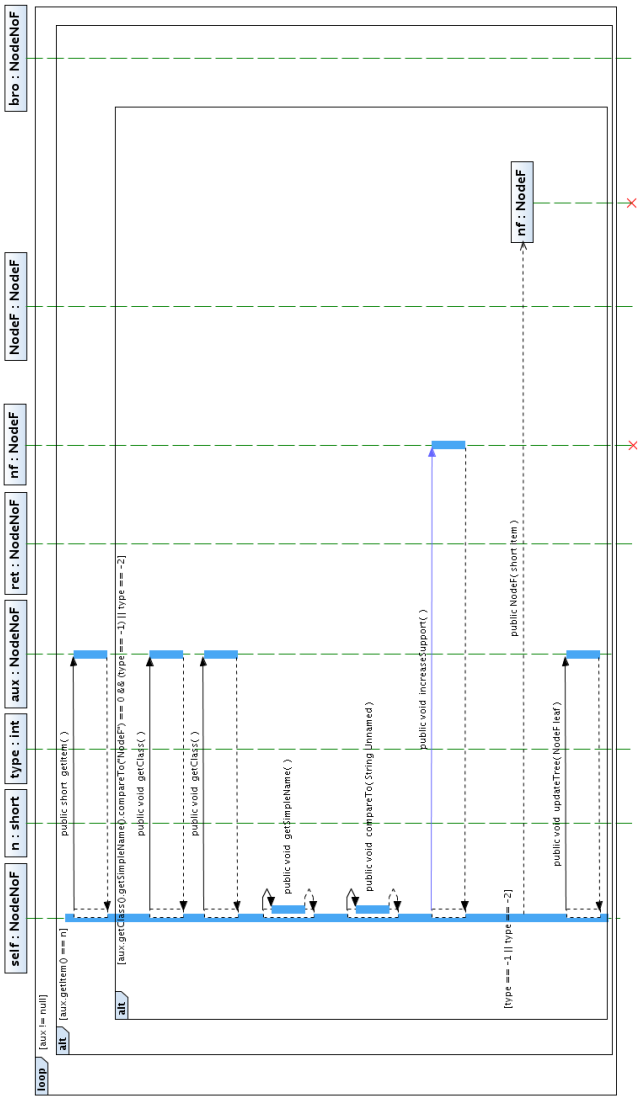

Figura 7.38: findBro

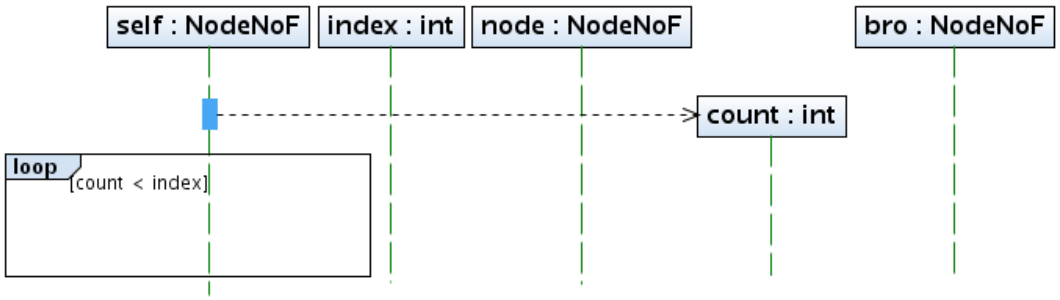

Figura 7.39: getChild

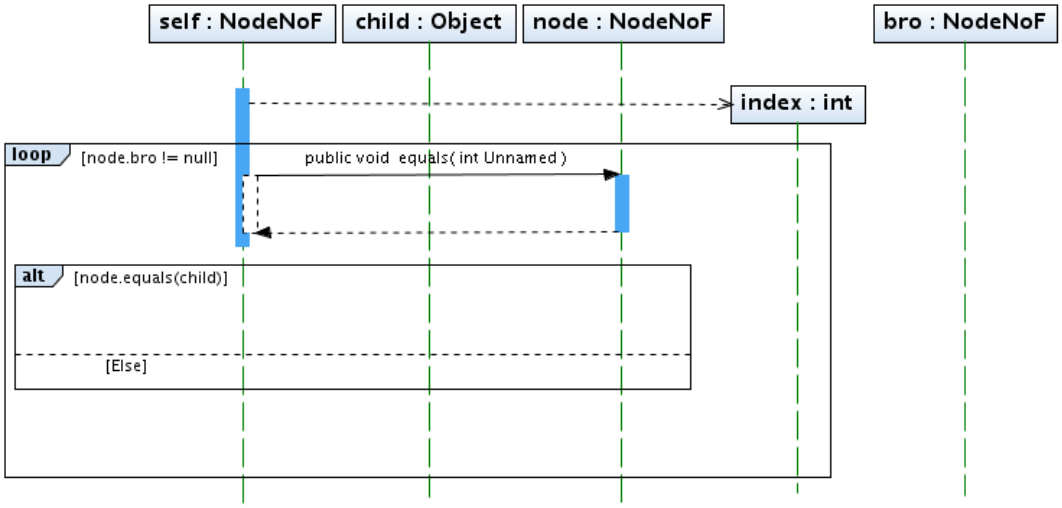

Figura 7.40: getIndexOfChild

Clase C45

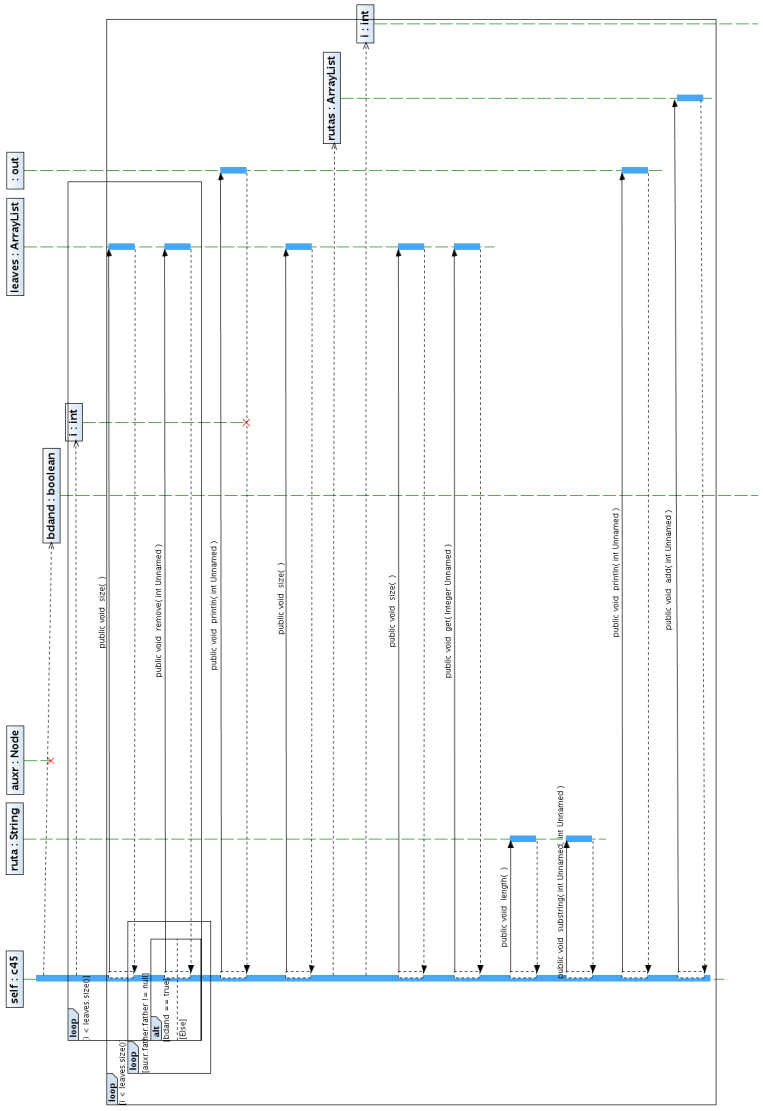

Figura 7.41: C45Rules

Clase myHasMap

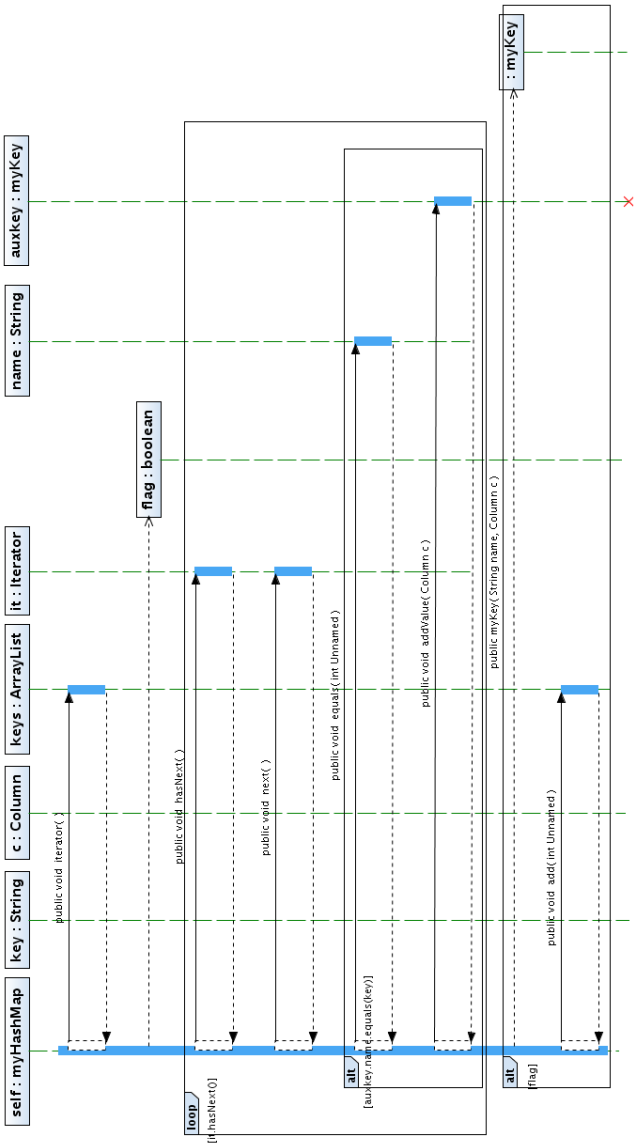

Figura 7.42: addColumn

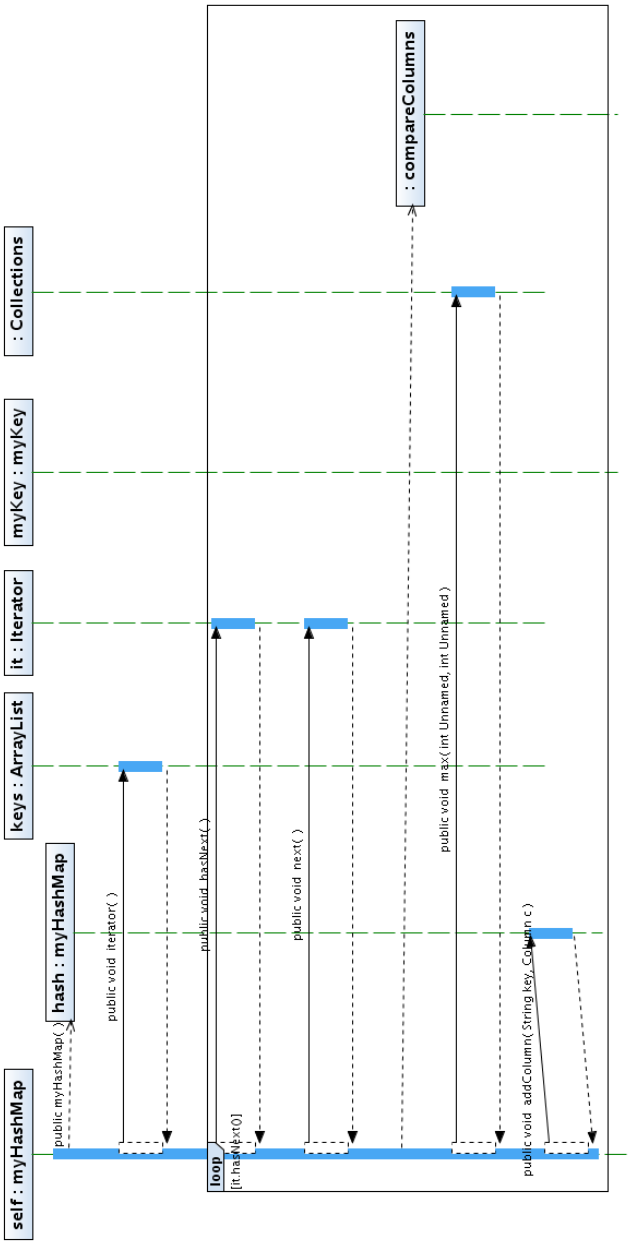

Figura 7.43: SearchColumn

Clase Route

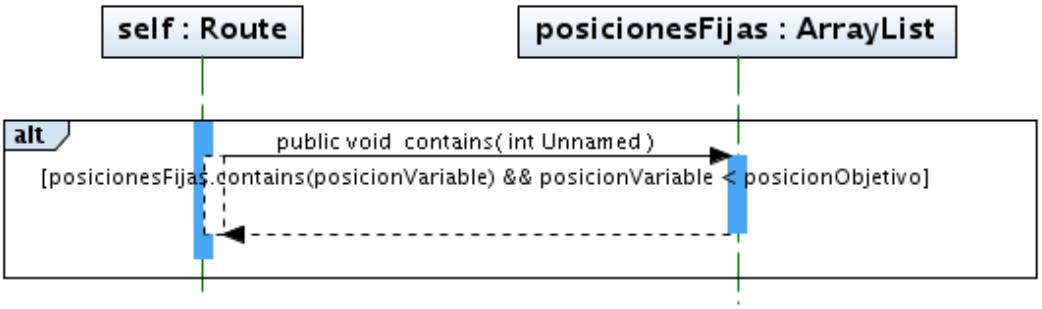

Figura 7.44: avanceVariable

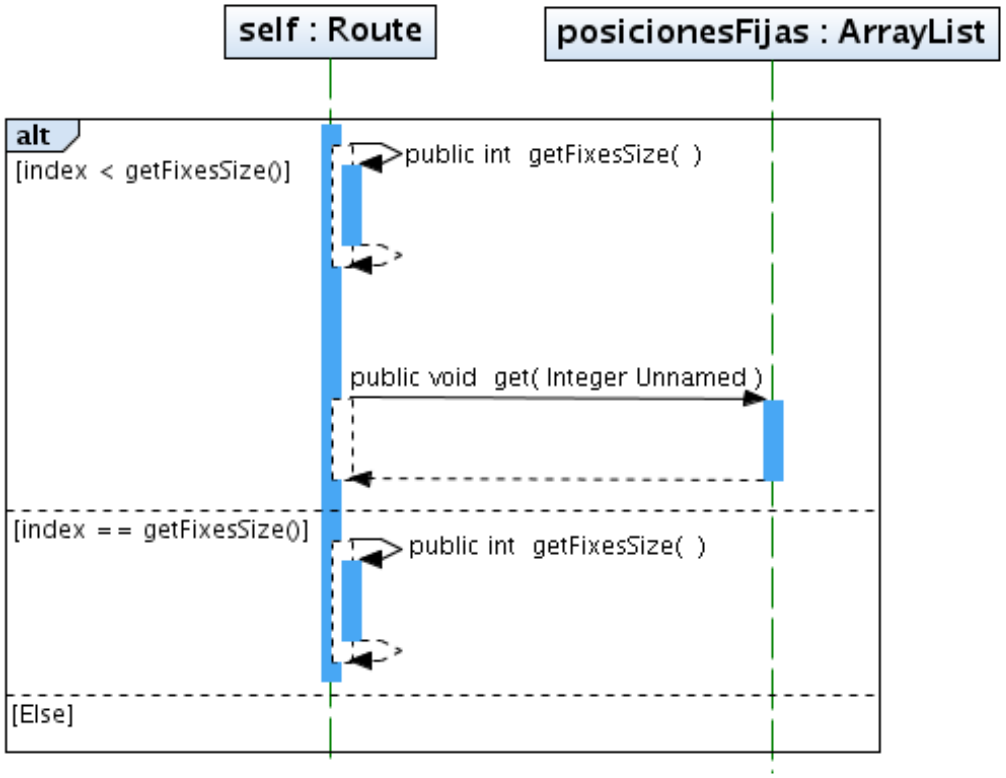

Figura 7.45: getIndex

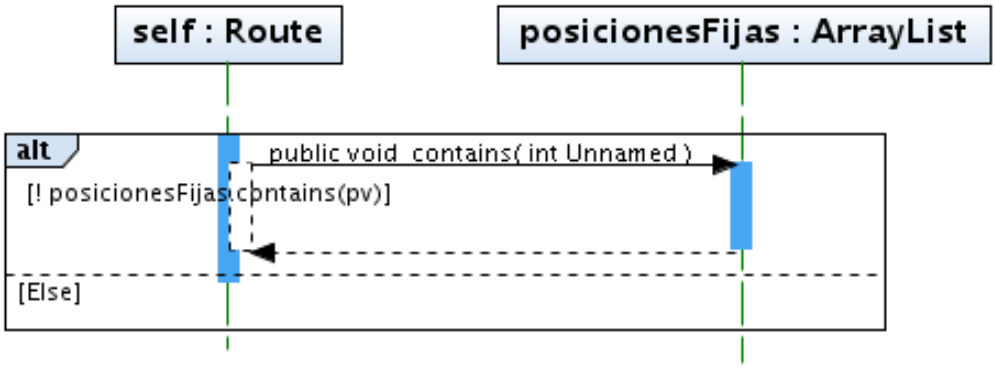

Figura 7.46: setPoscicionVariable

Clase TreeCounter

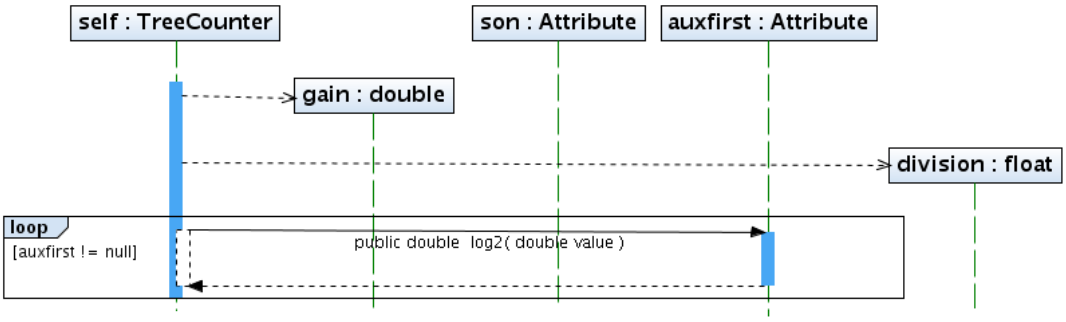

Figura 7.47: firstGain

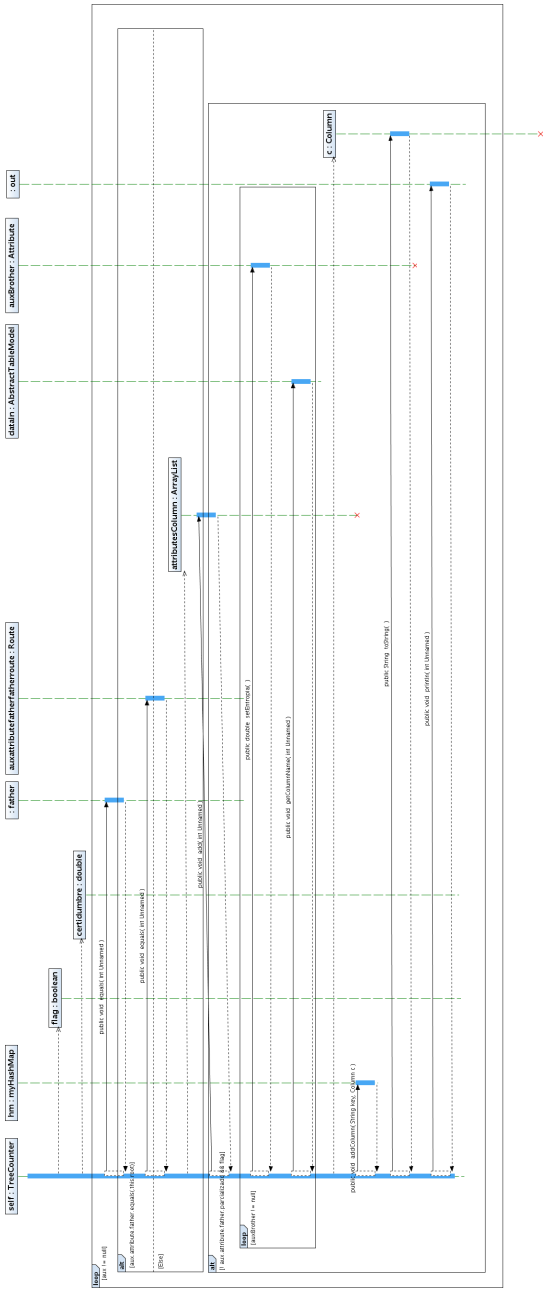

Figura 7.48: gain

Clase Attribute

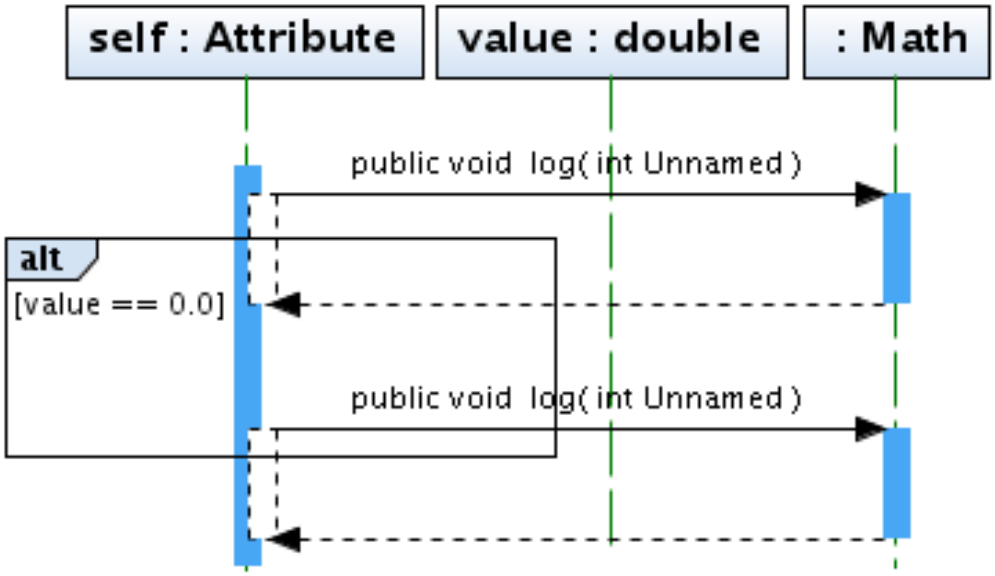

Figura 7.49: log2

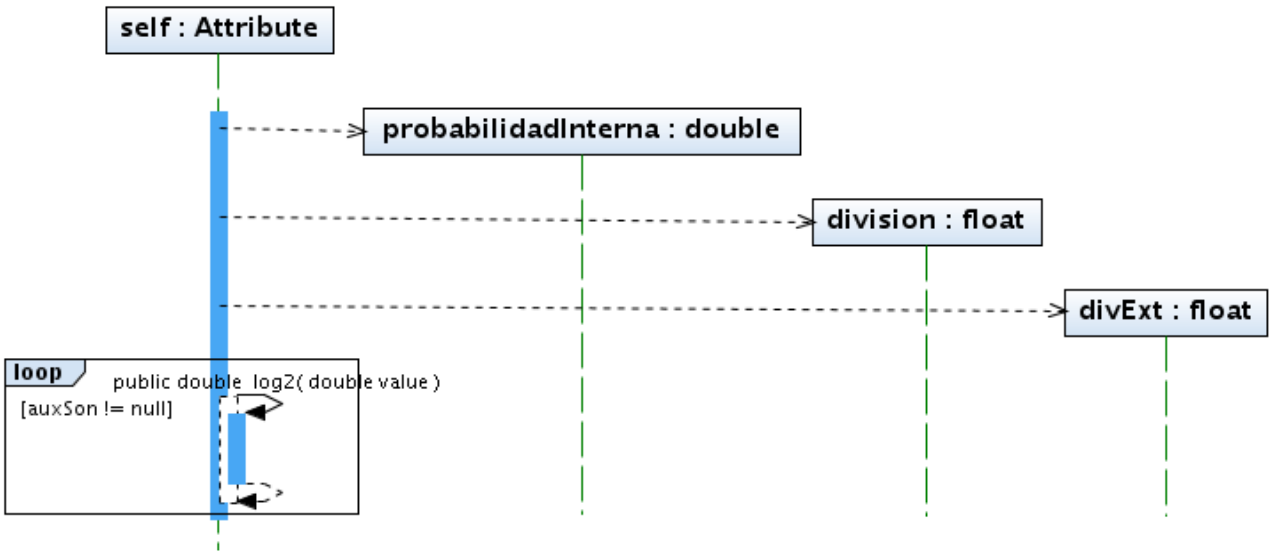

Figura 7.50: setEntropia

Clase Node

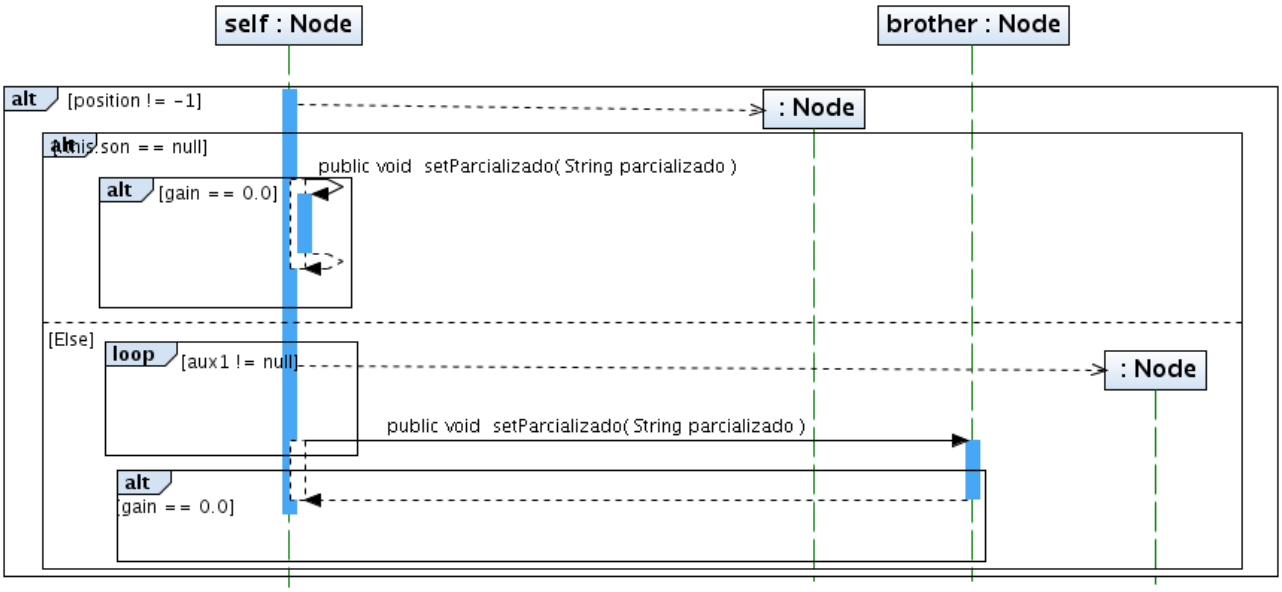

Figura 7.51: addSon

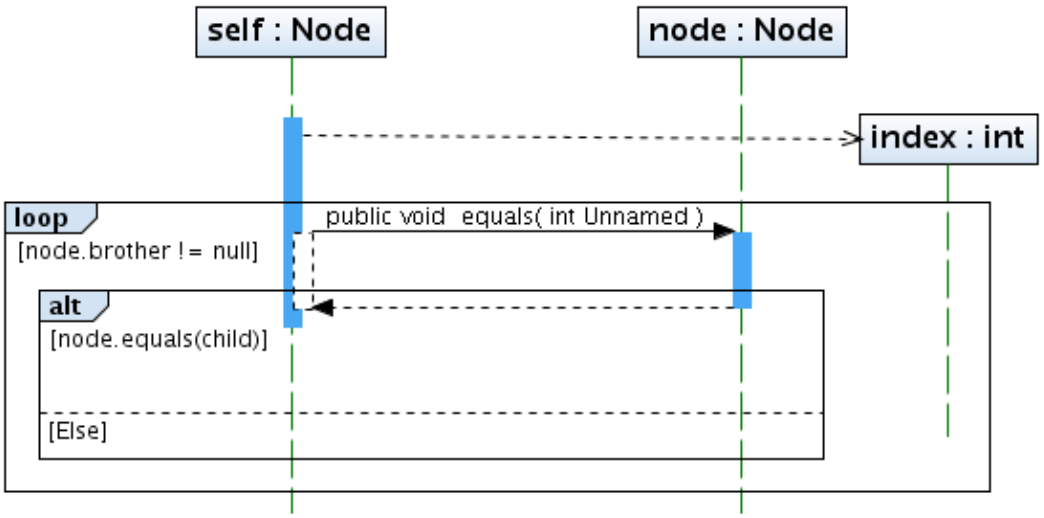

Figura 7.52: getIndexOfChild

7.2. Diseño

7.2.1. Diagramas de Colaboración

Clase Apriori

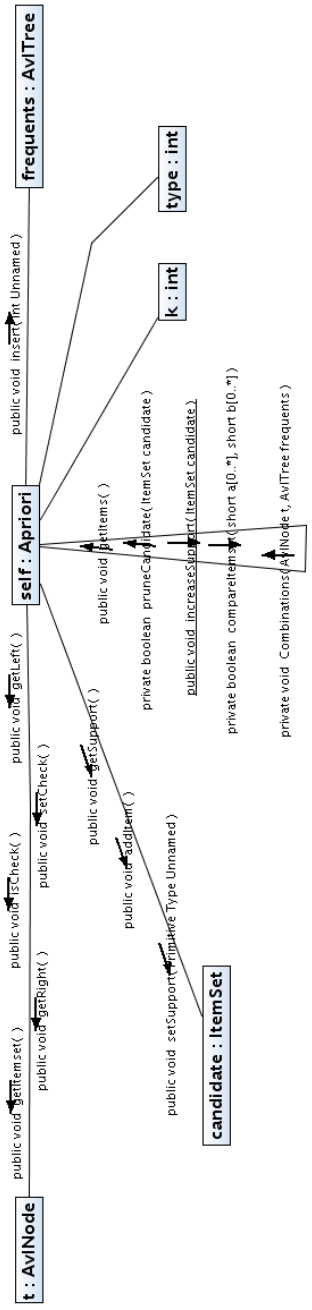

Figura 7.53: Combinations

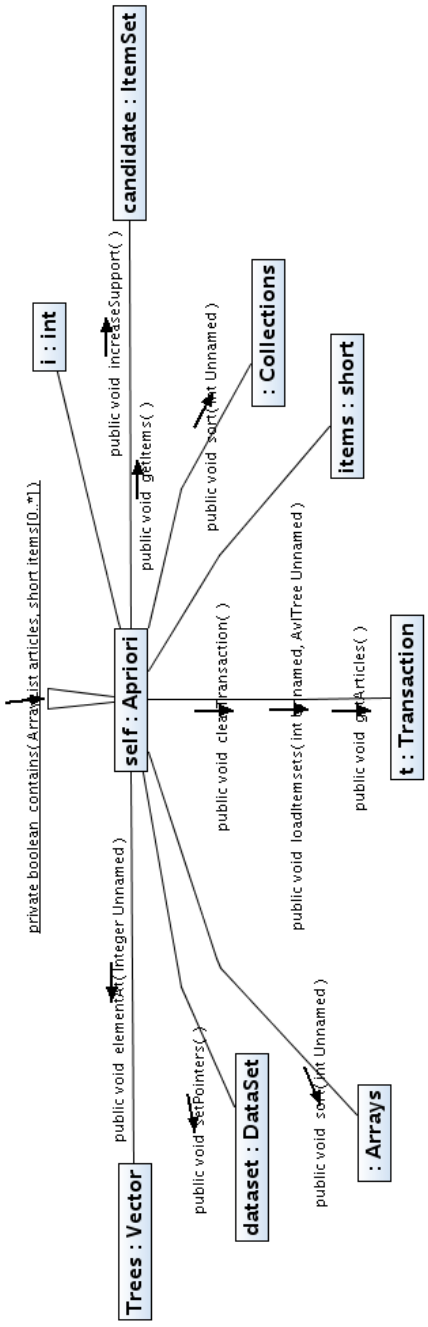

Figura 7.54: IncreaseSuport

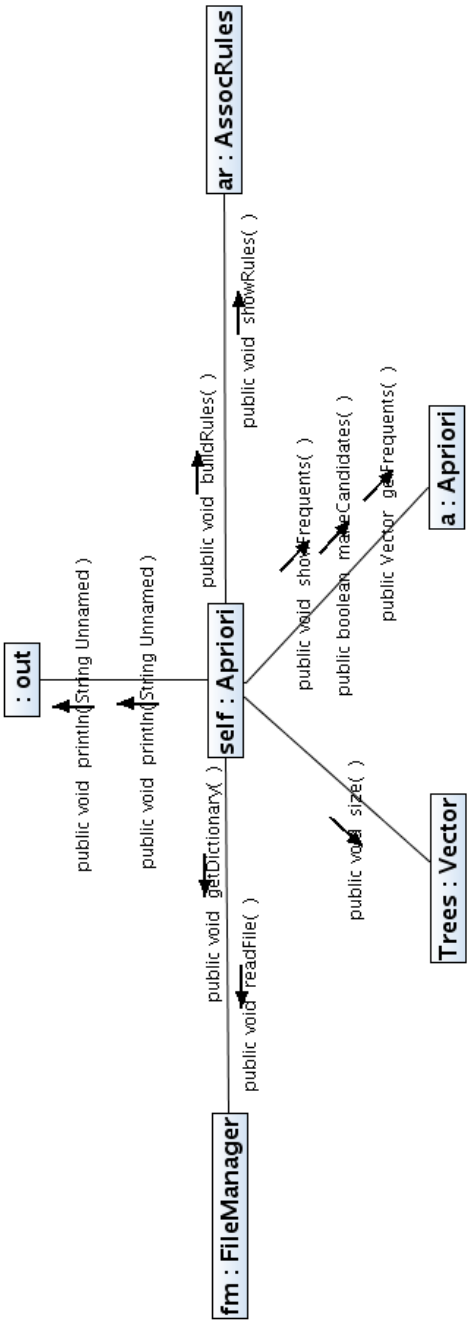

Figura 7.55: main

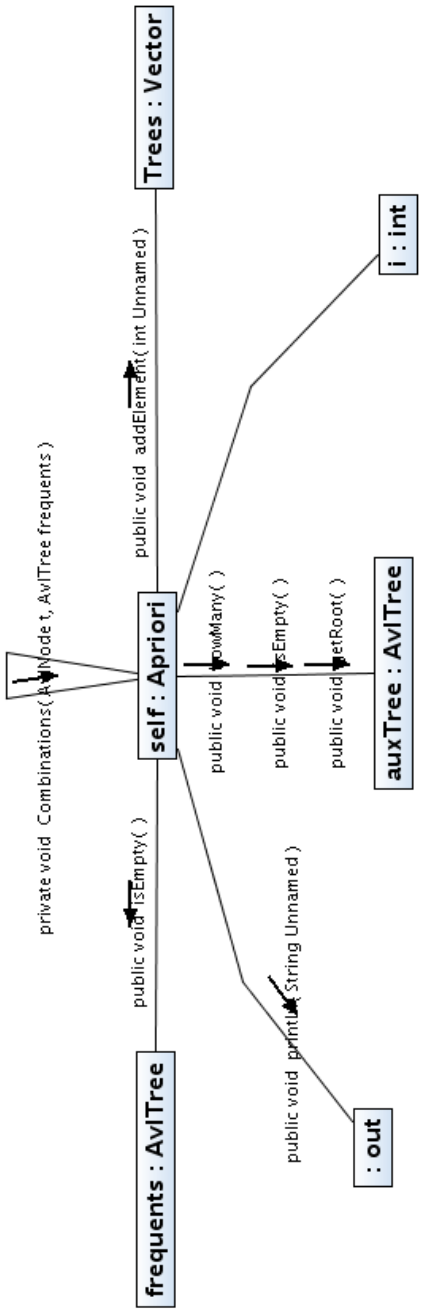

Figura 7.56: makeCandidates

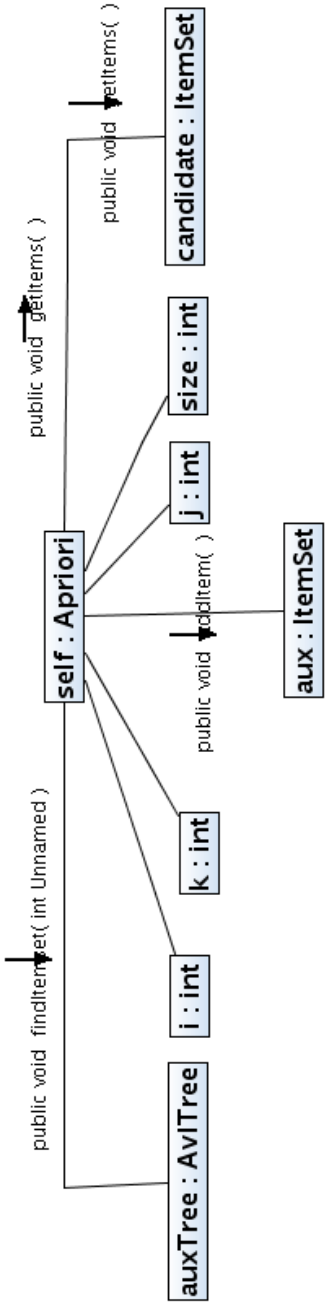

Figura 7.57: pruneCandidate

Clase EuipAsso

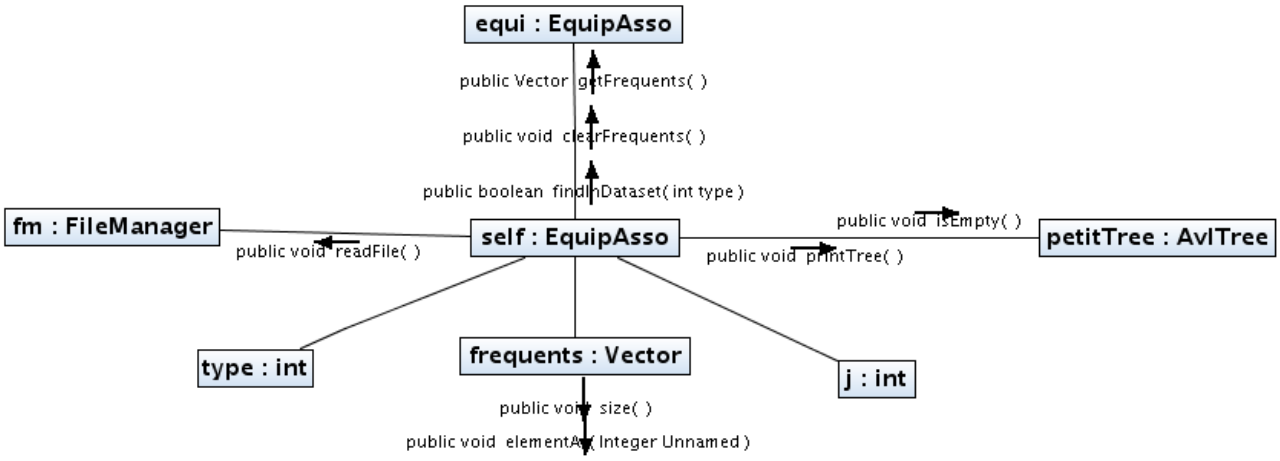

Figura 7.58: main

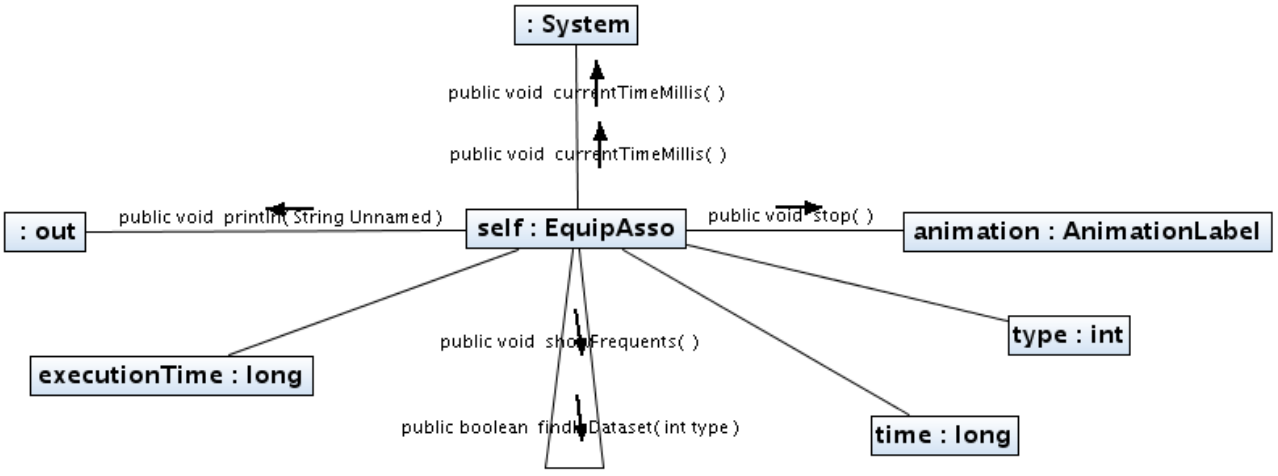

Figura 7.59: run

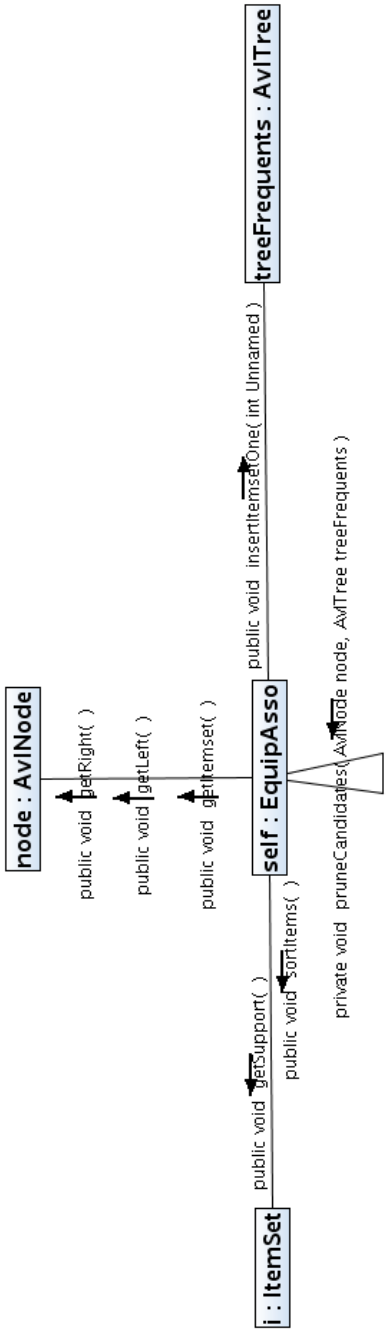

Figura 7.60: pruneCandidates

Clase Combinations

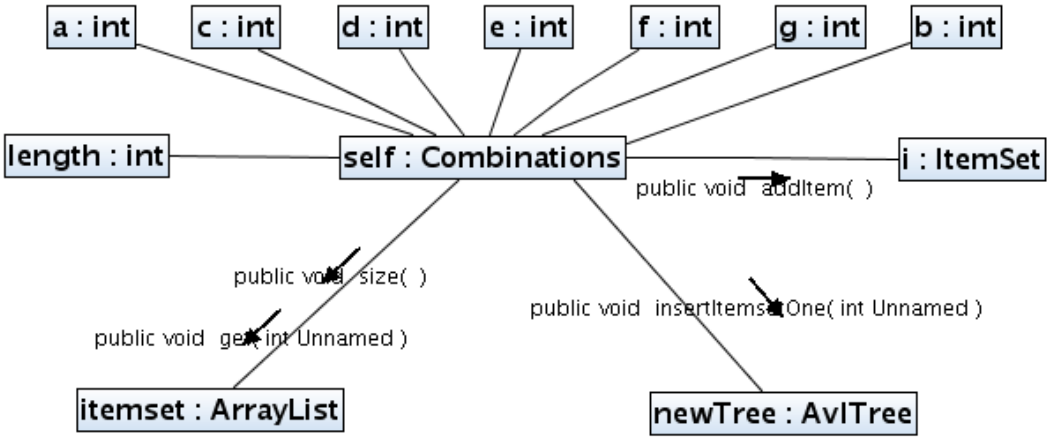

Figura 7.61: combine7

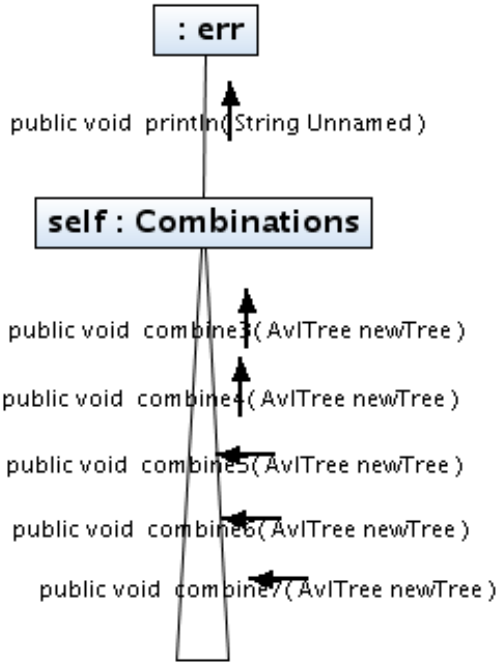

Figura 7.62: letsCombine.png

Clase BaseConditionals

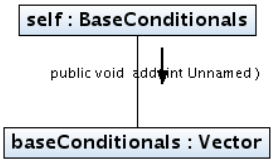

Figura 7.63: addBaseConditionals

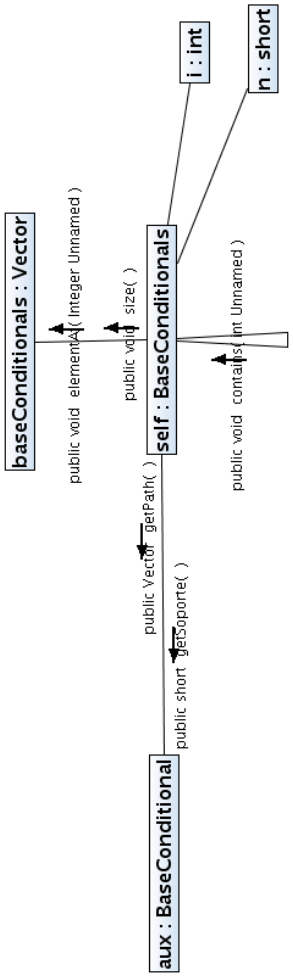

Figura 7.64: findItem

Clase Combinations

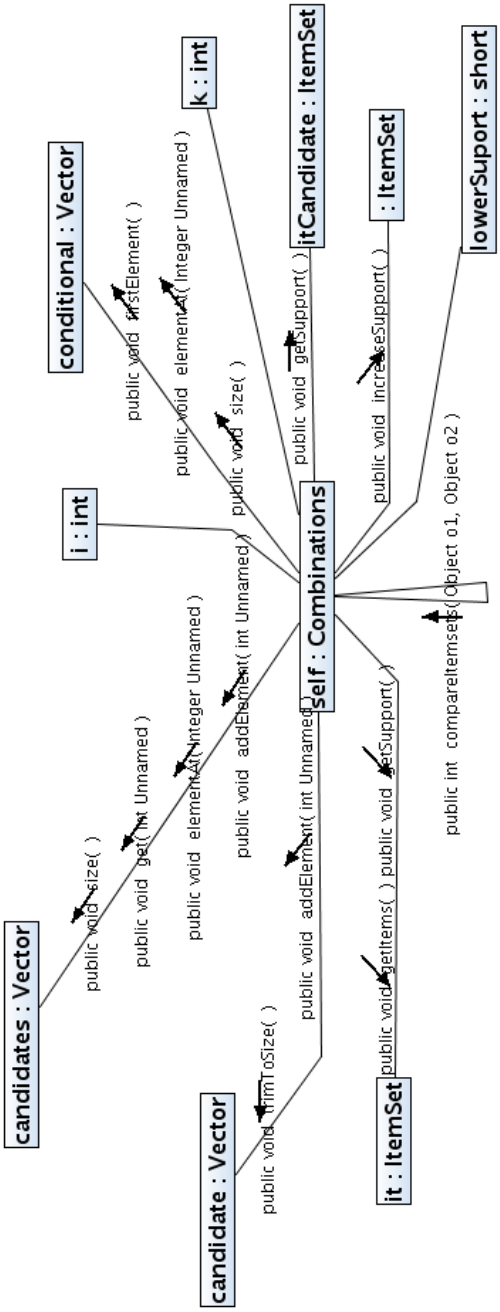

Figura 7.65: addCandidates

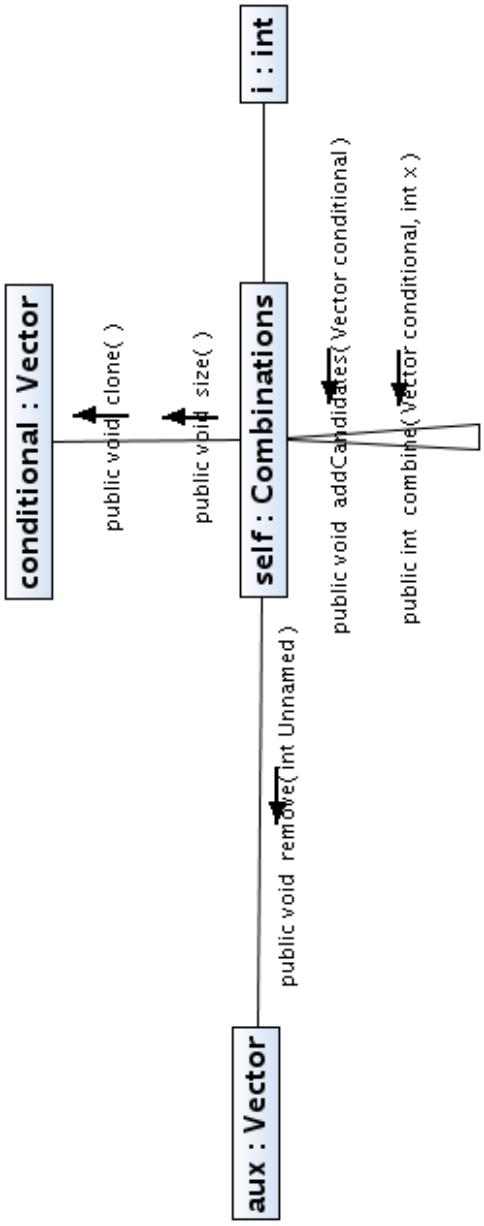

Figura 7.66: Combine

Clase FPGrowth

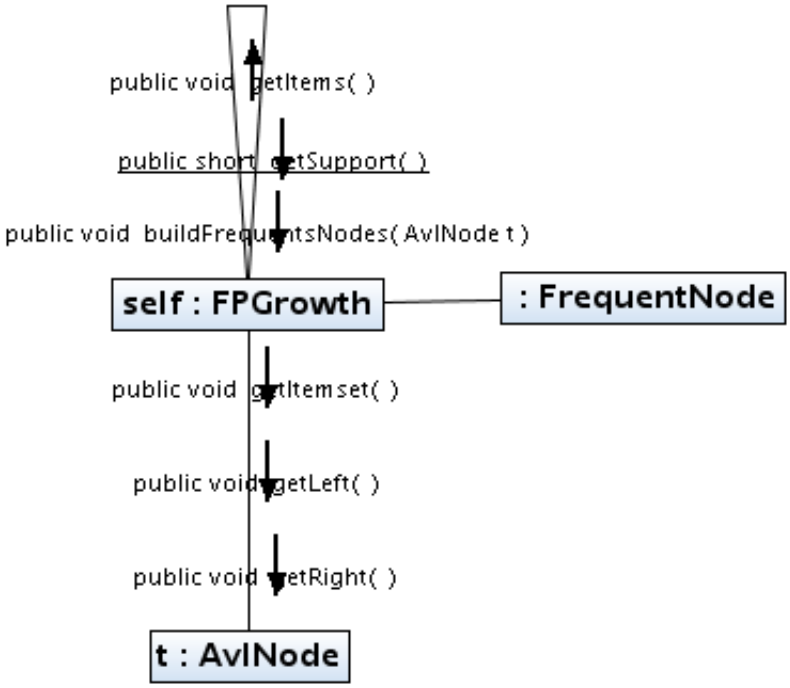

Figura 7.67: builtFrecuenceNode

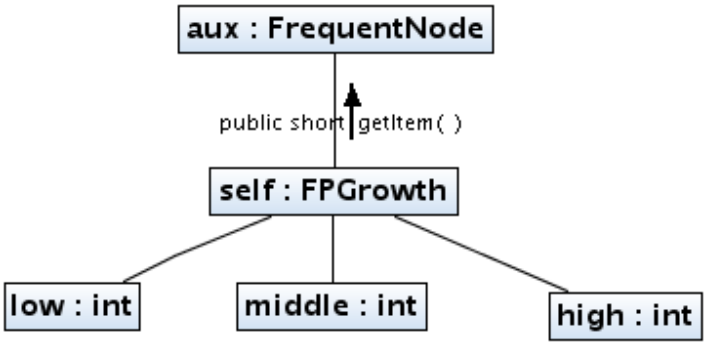

Figura 7.68: FrequentNode

Clase AvlTree

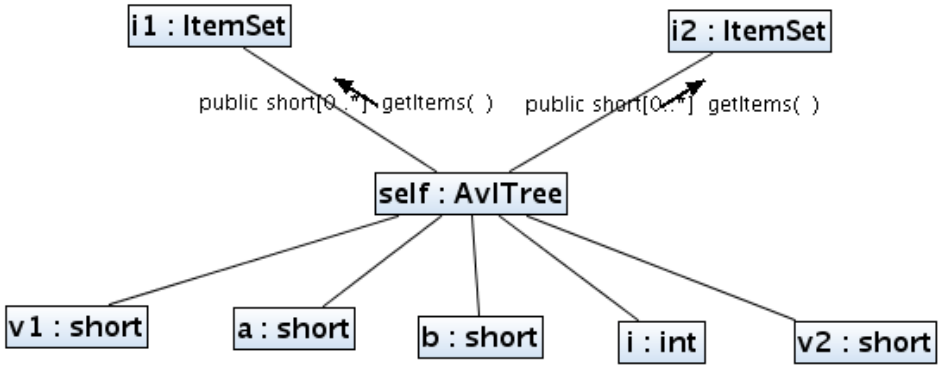

Figura 7.69: compareItems

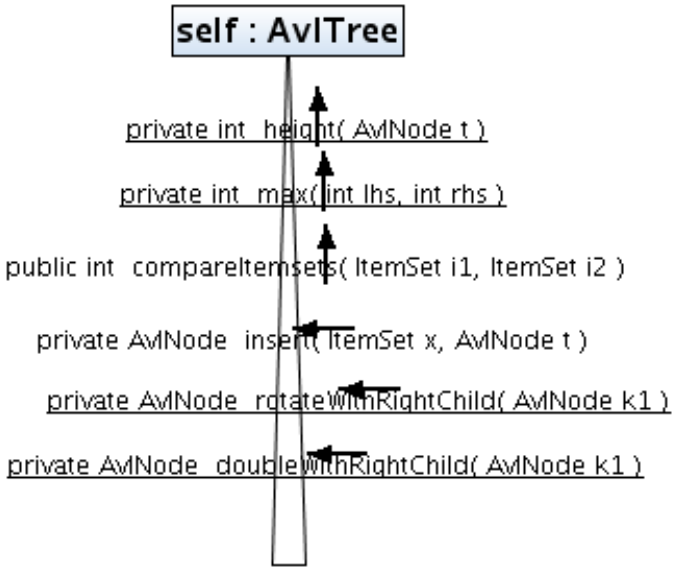

Figura 7.70: insert

7.2.2. Diagramas de Clase

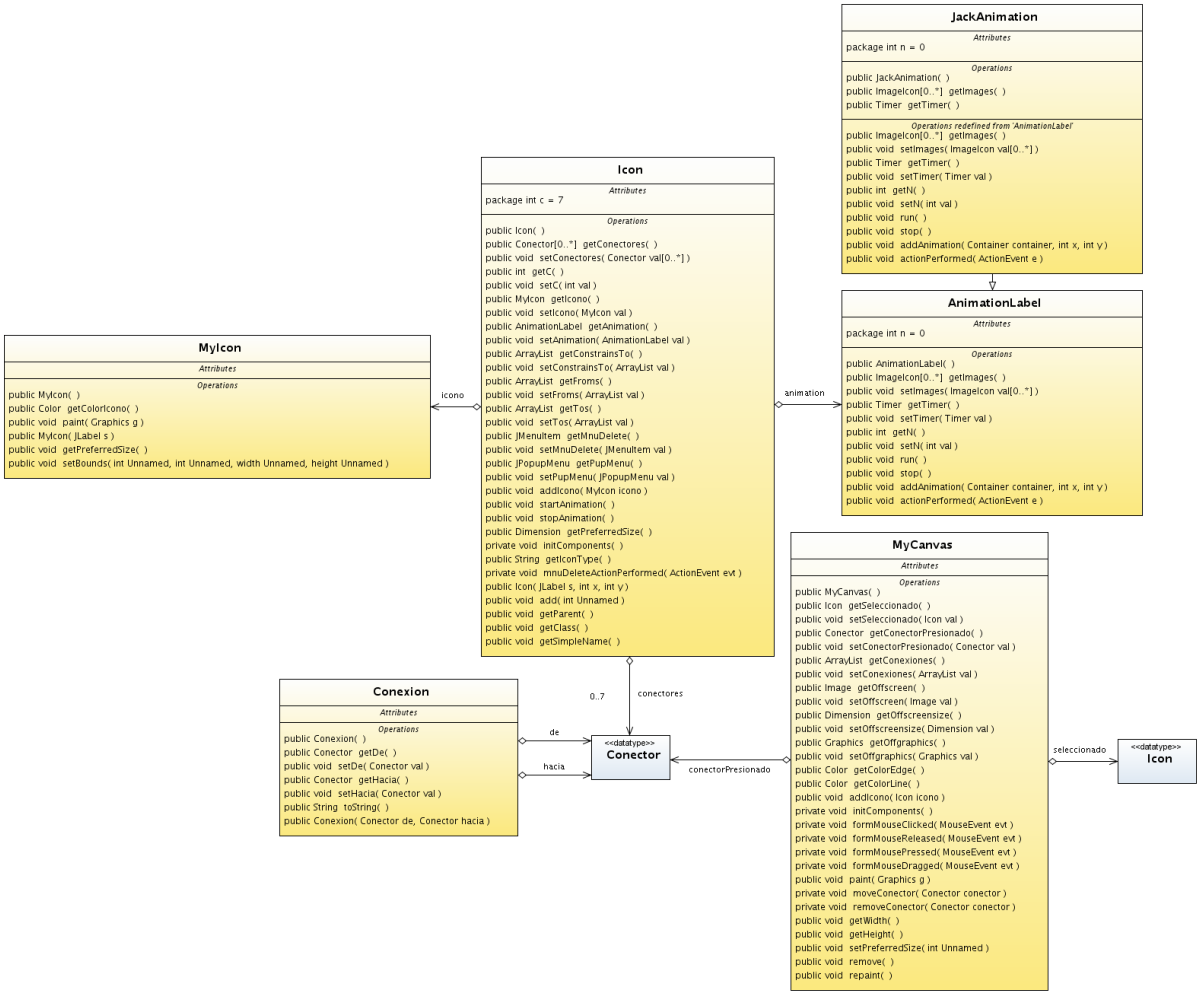

Figura 7.71: Paquete KnowledgeFlow

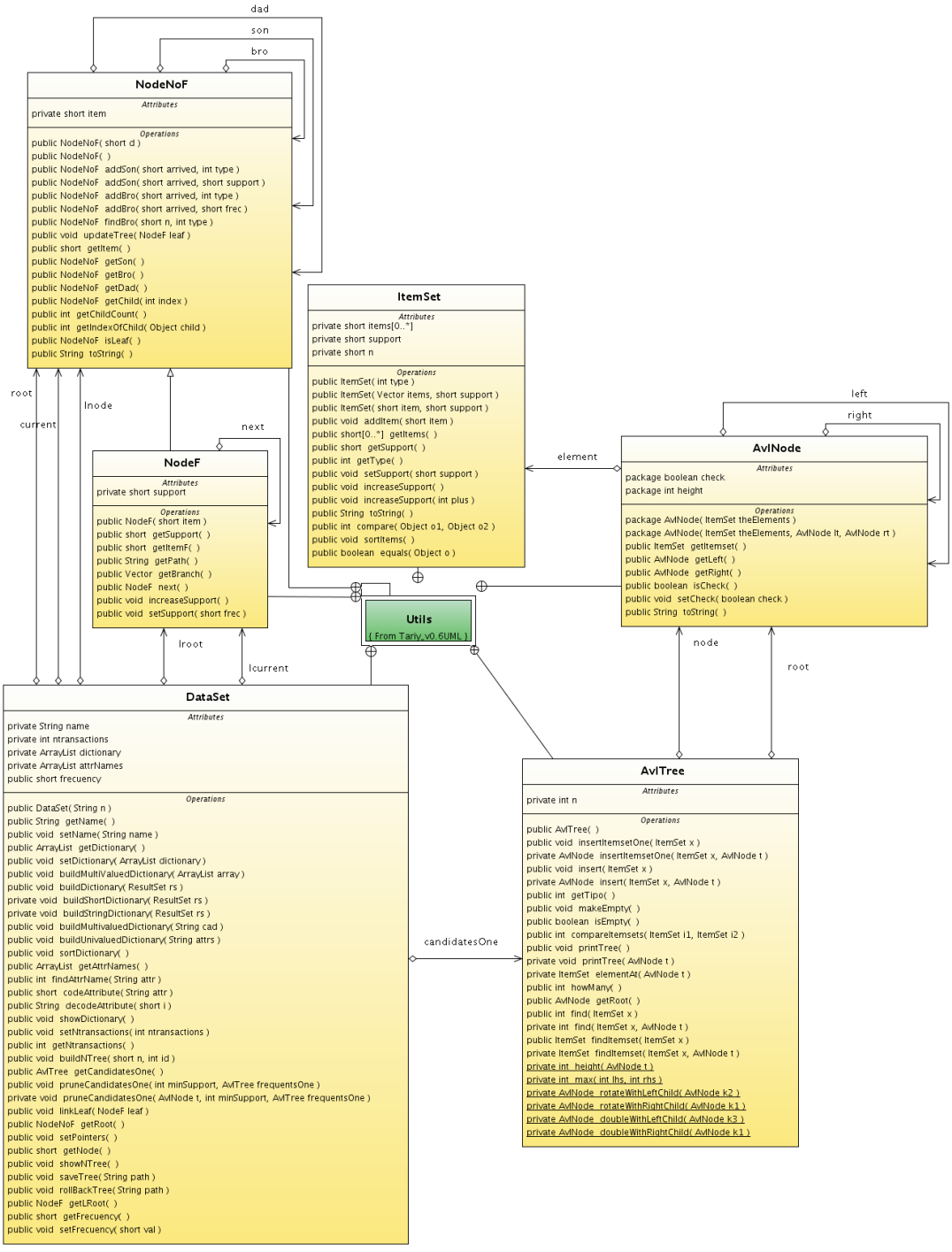

Figura 7.72: Paquete Utils

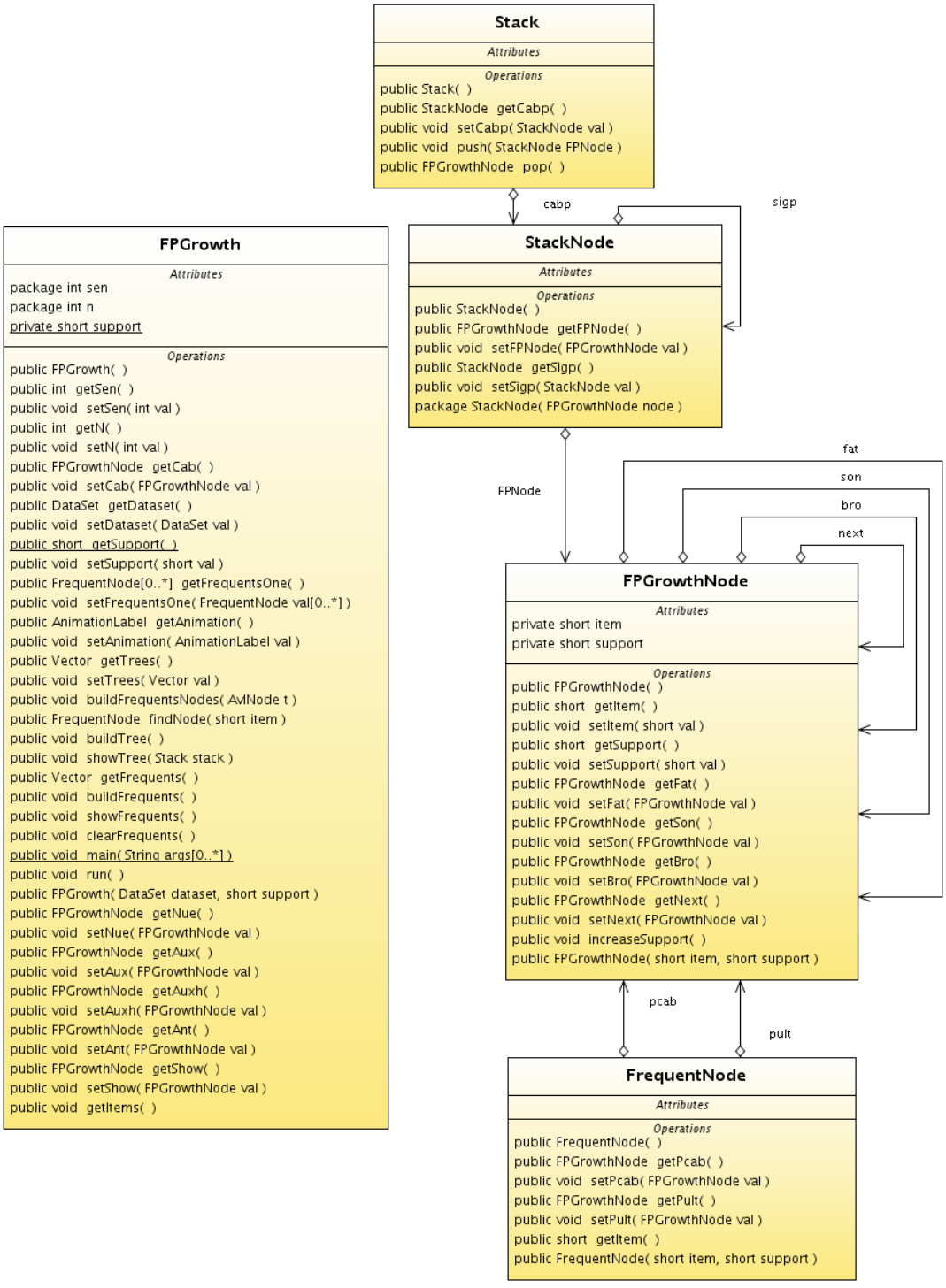

Figura 7.73: Paquete FPGrowth

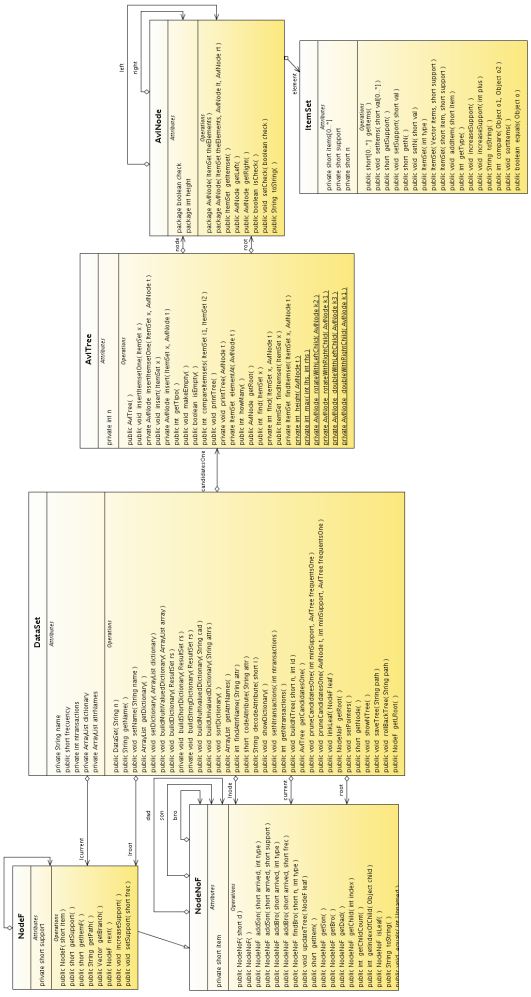

Figura 7.74: Paquete EquipAsso

Figura 7.75: Paquete Mate

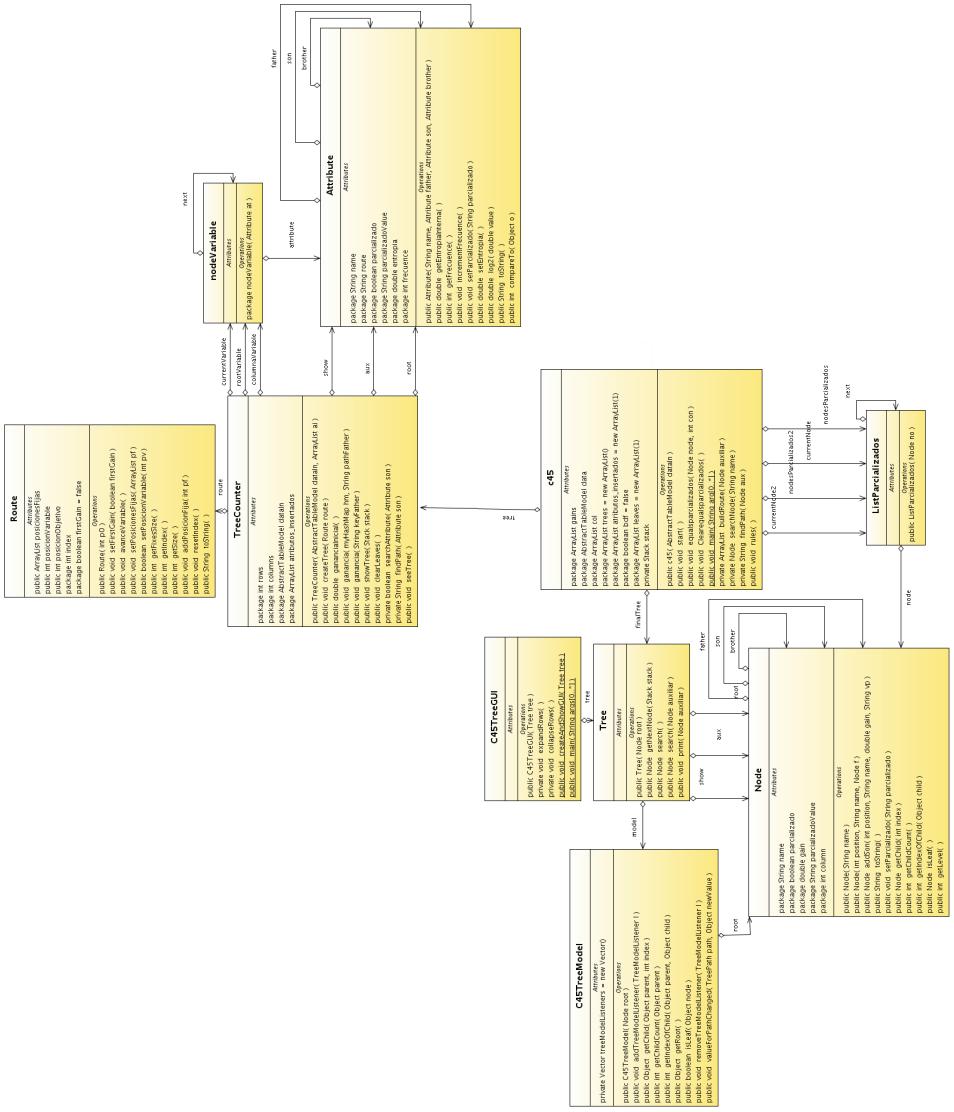

Figura 7.76: Paquete C45

7.2.3. Diagramas de Paquetes

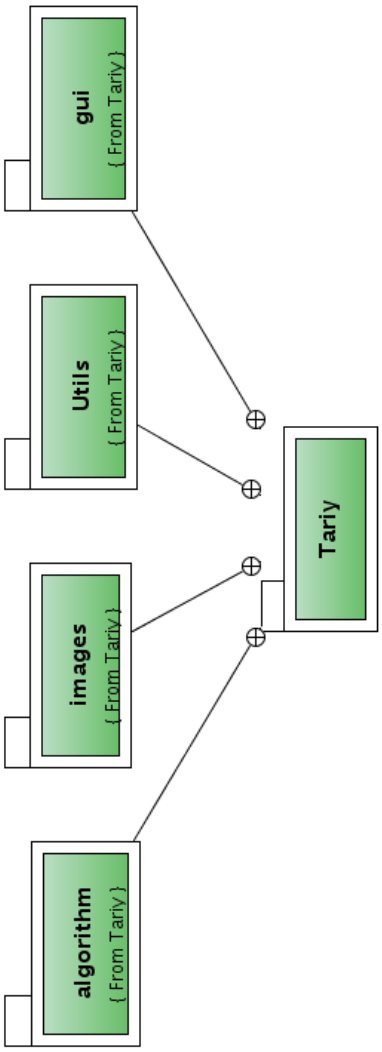

Figura 7.77: Paquete Principal

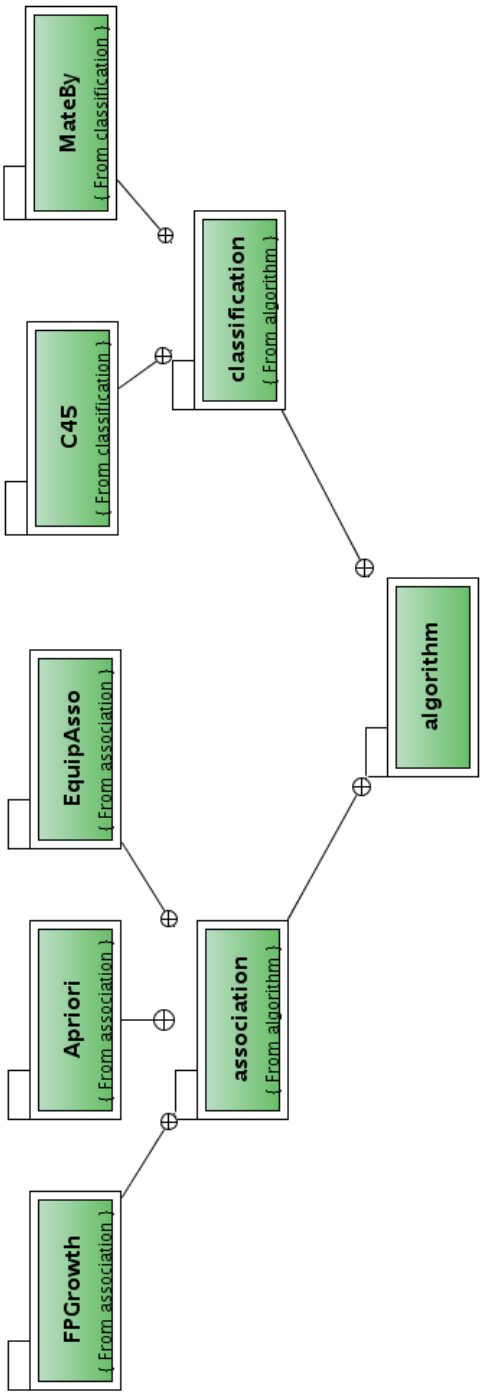

Figura 7.78: Paquete Algoritmos

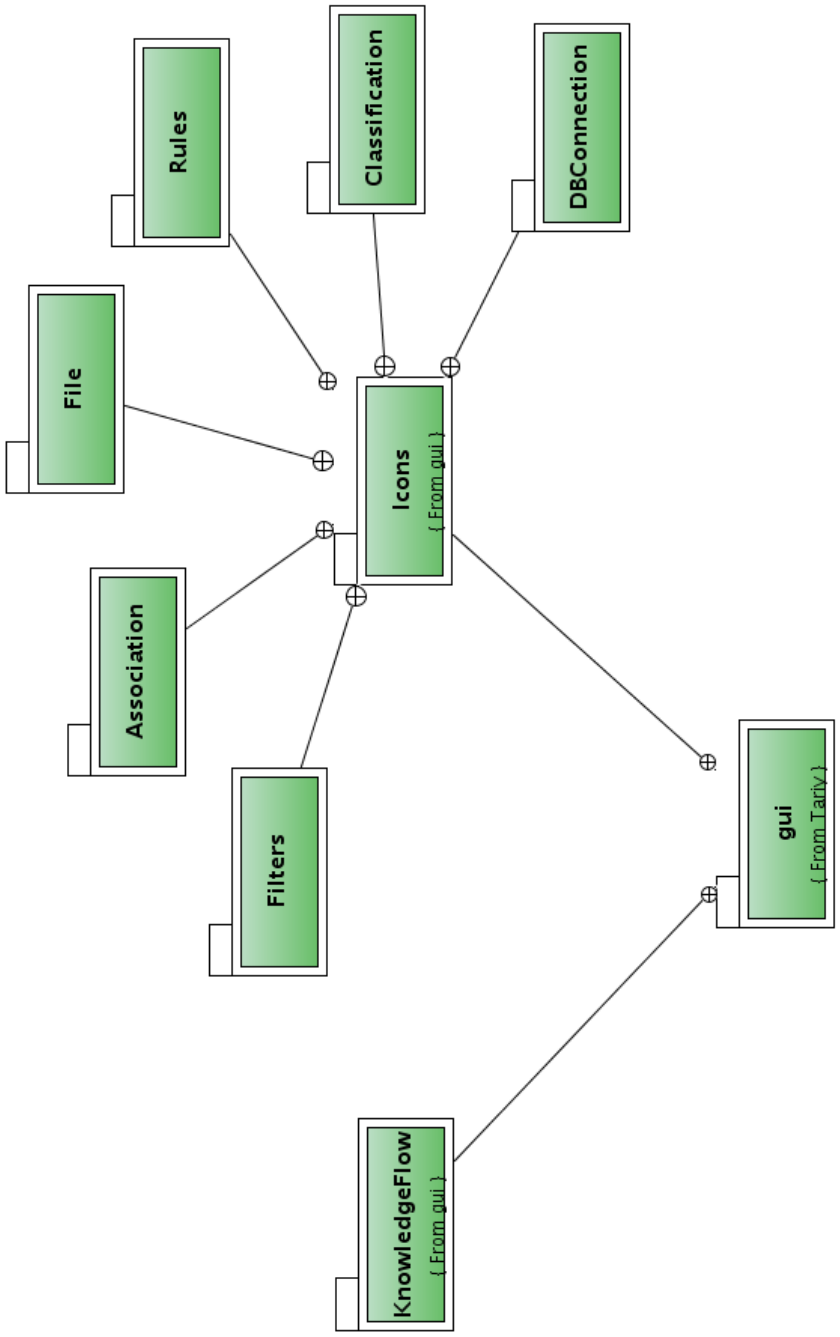

Figura 7.79: Paquete Interfaz Gráfica

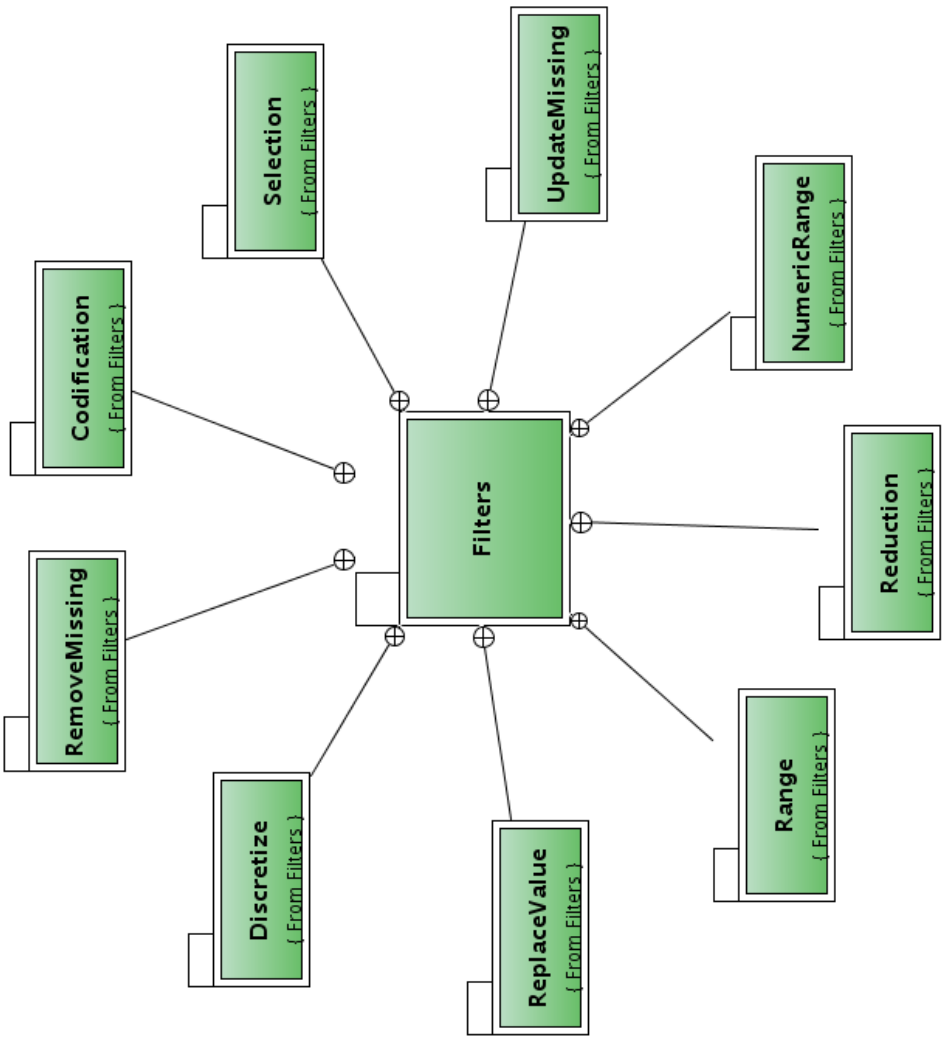

Figura 7.80: Paquete GUI Filtros

7.3. Implementación

7.3.1. Arquitectura de TariyKDD

Para el desarrollo de TariyKDD se utilizaron computadores con procesador AMD 64 bits, disco duro Serial ATA, útil al tomar los datos desde un repositorio y al momento de realizar pruebas de rendimiento de los algoritmos, ya que su velocidad de transferencia es de 150 MB/sg; además la RAM que se utilizo fué superior a los 512 MB, ya que la Minería de Datos requiere grandes cantidades de memoria por el tamaño de los conjuntos de datos.

El sistema operativo sobre el cual se trabajo durante la implementación de TariyKDD es Fedora Core en su versiones 3 y 5. El lenguaje de programación en el que esta elaborado TariyKDD es Java 5.0, actualización 06.

Dentro del proceso de Descubrimiento de Conocimiento, TariyKDD comprende las etapas de Selección, Preprocesamiento, Minería de Datos y Visualización de Resultados. De esta forma la implementación de la herramienta se hizo a través de los siguientes módulos de software cuya estructura se muestra en la figura 7.81.

Módulo de Conexión

El Módulo de Conexión permite al usuario acceder a los conjuntos de datos a través de un Archivo Plano o una Base de Datos.

La opción Archivo Plano le permite al usuario seleccionar un conjunto de datos que se encuentra en disco, en un archivo de acceso aleatorio, el formato para el archivo debe ser ARFF[15], debido a que este es uno de los más conocidos y tiene una estructura que lo hace fácil de comprender por ser estandar (debido a su estructura de etiquetas).

En cuanto a la Conexión a Bases de Datos, TariyKDD puede conectarse con PostgreSQL a través de su manejador JDBC tipo 4 [25]. Este driver es el más eficiente ya que traduce de forma directa las peticiones del API Java al protocolo nativo del Sistema Gestor, con la ventaja de que resulta sencilla la migración a otro diferente, lo único que habría que hacer sería descargar el driver del fabricante adecuado.

Almacenamiento de datos en memoria

Una vez se ha hecho la conexión al conjunto de datos, ya sea a través de Archivo Plano o mediante Bases de Datos, el siguiente paso que se hace en TariyKDD

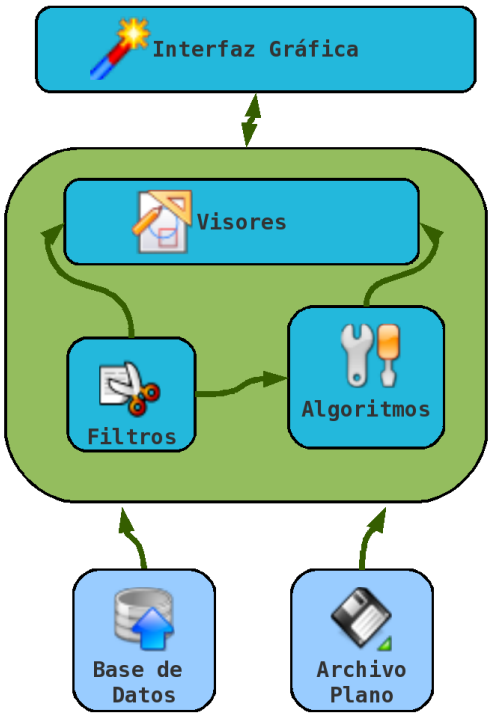

Figura 7.81: Árquitectura TariyKDD

es almacenar los datos en memoria principal, en una estructura especial que administra de manera optima el tamaño de los datos. La cual se describe en este módulo.

Una de las principales dificultades dentro del proceso de descubrimiento de conocimiento es el uso adecuado de los recursos del sistema y en especial de la memoria principal si tenemos en cuenta que se pretende trabajar con amplios volúmenes de datos. Ya sea cargando un conjunto de datos desde un archivo plano o directamente desde una conexión a un SGBD se espera organizar estos datos de una manera compacta con el objetivo de almacenar esta información en memoria principal evitando repetidas llamadas a disco lo que significa un aumento en los tiempos de ejecución de la herramienta.

Formatos tradicionales para el almacenamiento de transacciones, como el formato ARFF, trabajan con cabeceras donde se registran los diferentes campos o atributos del conjunto de datos seguidos de las transacciones como tal, separadas una de

otra por cambios de línea donde cada atributo esta separado a su vez por comas. En conjuntos de datos discretizados cuyos atributos pueden tomar un rango determinado de valores, dentro de la parte en donde se almacenan los datos es común encontrar segmentos de transacciones que coinciden o incluso transacciones completas que se repiten.

Es posible aprovechar estas coincidencias dentro de una estructura de datos como un árbol N-Ario donde cada rama represente una posible transacción y donde las bifurcaciones dentro de esa rama representen segmentos compartidos con otras transacciones o inclusive transacciones que estén contenidas dentro de esa misma rama.

Para explicar de mejor manera esta propuesta consideremos el conjunto de datos representado en la siguiente tabla, hay que tener en cuenta que los items del conjunto de datos original son codificados para mejorar la administración de la memoria.

T	**A**	**B**	**C**	**D**	**E**
1	1	1	3	4	5
2	1	1	3	4	6
3	1	2	3	6	7
4	2	3	4	1	3
5	1	2	3	6	7
6	2	3	4	6	5

Se puede ver que los cuatro primeros campos de las transacciones 1 y 2 son iguales por lo que pueden compartir nodos dentro del árbol N-Ario.

Como se puede ver en la figura 7.82, los valores entre paréntesis en las hojas del árbol representan el número de repeticiones de la transacción, por ejemplo la número 3 es la misma transacción 5, por tanto en el árbol N-Ario estos dos registros serán almacenados en una sola rama, con la precaución de contar su soporte.

Entre menos número de valores tenga un atributo y entre más transacciones formen el conjunto de datos, existe mayor posibilidad de encontrar coincidencias y aprovechar segmentos de transacciones ya almacenadas para guardar las nuevas que se repitan.

Un análisis del presente formato para compresión de datos con respecto al formato ARFF se muestra en los anexos, en el cuadro 9.8, donde se registra el tamaño en

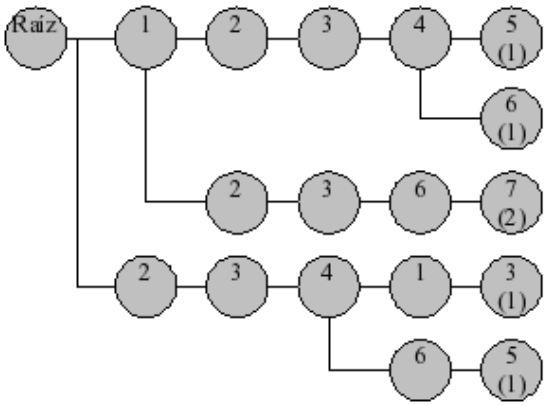

Figura 7.82: Árbol N-Ario

disco de cada formato al almacenar conjuntos de datos con diferente número de transacciones y atributos.

Módulo Filtros

El módulo filtros o data cleaning, se encarga de hacer un refinamiento de los datos en dos etapas, por un lado hace un proceso de limpieza sobre datos corruptos, vacios, ruidosos, inconsistentes, duplicados, alterados etc, por otro lado hace una seleccion de estos datos para escojer aquellos que brinden informacion de calidad, aplicando muestreos, discretizaciones, etc. De esta forma obtenemos datos depurados según el objetivo del analista, para que posteriormente se pueda aplicar el nucleo KDD o de Minería de Datos sobre datos coherentes, limpios y consistentes.

A este módulo, pertenecen los filtros de Remove Missing, Update Missing, Selection, Range, Reduction, Codification, Replace Value, Numeric Range y Discretize, los cuales se alimentan y presentan sus resultados a través de un TableModel que es el medio por el cual comunican sus flujos de datos.

Módulo algoritmos

Dentro de este módulo se encuentran dos tipos de algoritmos, los de asociación y los de clasificación:

1. Asociación: Los algoritmos de asociación implementados en TariyKDD son EquipAsso, FPGrowth y Apriori. Los tres algoritmos utilizan un vector de árboles AVL balanceados para almacenar los itemsets frecuentes. Cada posición del vector almacena un tipo de itemsets frecuentes en un árbol AVL. De

esta forma la posición 0 del vector almacena un árbol AVL que contiene los itemsets frecuentes tipo 1. En este módulo de TariyKDD, la mayor ventaja que proporcionan los árboles AVL es la rapidez con la que se realizan las busquedas a la hora de determinar si un itemset frecuente ya existe.

Así mismo los tres algoritmos usan el árbol N-Ario que se describio anteriormente para almacenar el conjunto de datos que se vaya a minar. Los datos comprimidos en esta estructura son tomados como entradas por los tres algoritmos, a partir de estos se aplican las técnicas de Minería de Datos correspondientes y el producto final son las reglas.

2. Clasificación: Dentro de este módulo en TariyKDD se implementaron los algoritmos Mate y C4.5. Para los algoritmos de clasificación igual que en los de asociación la estructura que se utilizo para su implementación fué un árbol N-Ario. En C4.5 y Mate para almacenar los datos en primera instancia. Ya que ha medida que se desarrollan los algoritmos este árbol va cambiando, así, al final el árbol tiene las reglas de clasificación.

   La estructura utilizada para mostrar las reglas de clasificación es un árbol N-Ario que implementa los métodos de la interfaz de Java, TreeModel , la cual define un modelo de datos adecuado para poder visualizarlos en un JTree o un control de Java que despliega las reglas jerarquicamente de acuerdo a la estructura en la que estan almacenadas, es decir el árbol N-Ario.

Módulo Visores

Este módulo implementa las clases necesarias para mostrar de forma gráfica las reglas que se obtienen después de haber realizado Minería de Datos.

Si el usuario ha utilizado algoritmos de asociación obtiene como resultado reglas, que puede observar a través de una JTable, la cual es utilizada para desplegar y editar tablas de dos dimensiones. Para desplegar estas reglas, primero, se tiene un array con los datos que se van a mostrar, a partir de estos se construye un módelo propio de tabla implementando los métodos de la interfaz Java, TableModel. Después simplemente el módelo se pasa al constructor de JTable para que esta clase se encarge de desplegar las reglas.

Pero si durante el proceso de Minería de Datos se utilizaron algoritmos de clasificación, las reglas son desplegadas en un árbol N-Ario que dentro de TariyKDD se ha llamado Árbol de Resultados. La construcción de este se la realiza como se

dijo anteriormente implementando los métodos de la interfaz TreeModel, modelo que luego es usado por un JTree para visualizar las reglas de manera jerárquica.

Módulo GUI

El módulo de GUI es usado en los 4 módulos anteriores y es el encargado de brindrar una interfaz gráfica amigable al usuario. Para su desarrollo se utilizaron las funcionalidades del proyecto Matisse, proyecto encargado de proveer facilidades para la construccion de entornos gráficos a los proyectos desarrollados con NetBeans y que da un soporte a las aplicaciones que usan Swing, el cual es un conjunto de clases y componentes usados en interfaces gráficas desde botones hasta tablas y estructuras de árboles.

7.3.2. Descripción de clases

Paquete Utils

Clase DataSet En esta estructura los algoritmos de asociación almacenan los datos o items de forma comprimida, ocupando menos espacio en memoria. La estructura utilizada por DataSet es un árbol N-Ario que almacena los datos en cada nodo como tipo short. Lo particular de esta estructura es el aprovechamiento de la memoria principal, ya que en una sola rama almacena items de diferentes transacciones, controlando individualmente su número de apariciones.

Clase FileManager Esta clase gestiona todo lo relacionado con flujos de archivos, como por ejemplo crear un archivo plano, construir el diccionario de datos a partir de un archivo de acceso aleatorio, entre otras funciones.

Clase BaseDatos Esta clase gestiona todo lo relacionado con el manejo de las Bases de Datos, como la conexión, y la selección, de atributos.

Clase NodeNoF Esta clase representa un nodo básico del DataSet, este nodo no tiene soporte.

Clase NodeF Esta clase extiende a la clase NodeF y agrega el soporte a cada nodo del DataSet.

Clase AvlTree Los itemsets frecuentes generados por los algoritmos de asociación son almacenados en un árbol AVL balanceado, cuya estructura se encuentra en esta clase.

Clase AvlNode Es en si, un nodo del árbol AVL que almacena los itemsets frecuentes. Tiene un campo de tipo ItemSet en donde se guarda el dato que va en el nodo y tiene los punteros derecho e izquierdo a los demás nodos del árbol.

Clase ItemSet La clase ItemSet almacena un conjunto de items o itemsets en un vector así como su respectivo soporte.

Clase Transaction Esta clase gestiona todas las operaciones que deben hacerse sobre las transacciones. Como por ejemplo cargar las transacciones para los diferentes algoritmos y así como también realiza los diferentes ordenamientos de las transacciones, por item y por soporte.

Paquete Apriori

Clase Apriori Esta clase implementa todos los métodos necesarios para ejecutar el algoritmo Apriori. Los parametros necesarios para comenzar el algoritmo son: un soporte de tipo short y un dataset (estructura de tipo árbol N-Ario en la cual los datos son comprimidos) y sobre el cual se realizan tantos recorridos como itemsets frecuentes existan.

Paquete EquipAsso

Clase EquipAsso Para ejecutar el algoritmo EquipAsso los parametros necesarios son: un soporte de tipo short y un dataset (estructura de tipo árbol N-Ario en la cual los datos son comprimidos). Basicamente para obtener los itemsets frecuentes, lo primero que se debe hacer es recorrer el árbol N-Ario tomar cada una de sus transacciones, realizar todas sus combinaciones y ver cual de ellas pasa soporte y clasifica como itemset frecuente.

Clase Combinations Recibe como parametros el tipo y el itemset a combinar. El tipo es un número que indica hasta que profundidad se desea combinar el itemset en cuestion.

Paquete FPGrowth

Clase FPGrowth El algoritmo FPGrowth tiene su propio árbol N-Ario para almacenar los datos. Recorre el árbol y toma cada una de sus ramas, a partir de estas construye los Patrones Condicionales Base, luego los Patrones Condicionales y a partir de estos determina cuales son los itemsets frecuentes, los cuales se almacenan en un árbol AVL balanceado.

Clase FPGrowthNode Clase que tiene la estructura del árbol N-Ario del algoritmo FPGrowth. Es decir tiene los punteros necesarios para armar un árbol N-Ario, tiene un puntero al hijo, al padre y al hermano.

Clase BaseConditional Clase que almacena los Patrones Condicionales Base a partir del árbol N-Ario de la clase FPGrowth.

Clase BaseConditonals Solo los nodos que pasan el soporte minímo se consideran frecuentes, estos, tienen un puntero a cada uno de sus Patrones Condicionales Base, a partir de los cuales se obtienen los itemsets frecuentes.

Paquete MateBy

Clase MateBy Así como los demás algoritmos, MateBy utiliza el dataset o estructura de tipo árbol N-Ario para comprimir los datos que se van a minar. A partir de estos datos se realizan combinaciones y se calcula su entropía y su ganancia. Las combinaciones con la mayor ganancia se almacenan en un árbol de reglas, conformando así los resultados de MateBy.

Clase Entro Agrupa los nodos del árbol de acuerdo a su padre o rama y determina quienes tienen la mayor ganancia, de acuerdo a esto se construye el árbol de reglas.

7.3.3. Casos de uso reales

Ingreso a la aplicación

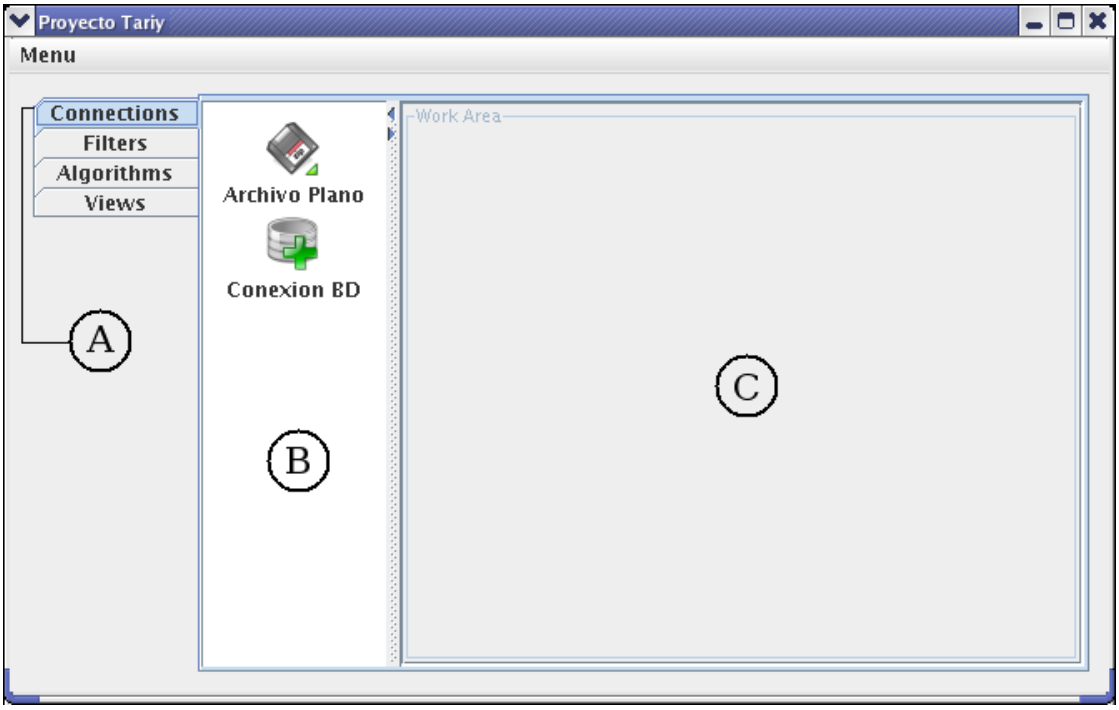

Figura 7.83: Ingreso a la aplicación

ACCIÓN DEL ACTOR	RESPUESTA DEL SISTEMA
1. El usuario ejecuta la aplicación	2. La interfaz gráica de la aplicación aparece como se muestra en la figura. A: Área de pestañas a través de las cuales se puede acceder a los difernetes módulos de la aplicación. Ej, el móulo por defecto es ‘Connections‘. B: Área en la que aparecen las opciones de cada módulo. Ej, las opciones del módulo ’Connections’ son ’Archivo Plano’ y ’Conexión DB’. C: Área de trabajo sobre la que se arman los proyectos de Minería de Datos

Módulo filtros

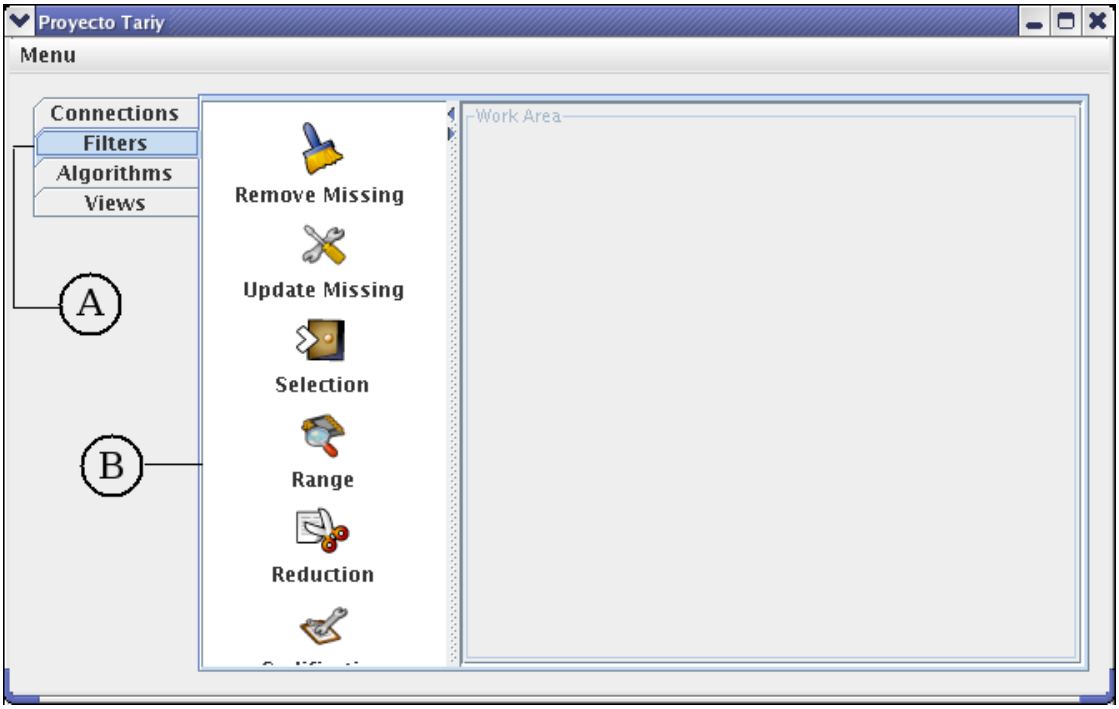

Figura 7.84: Módulo filtros

ACCIÓN DEL ACTOR	RESPUESTA DEL SISTEMA
1. El usuario hace click en la pestaña A 'Filtros'	2. Aparecen B Las opciones del módulo 'Filtros'

Módulo algoritmos

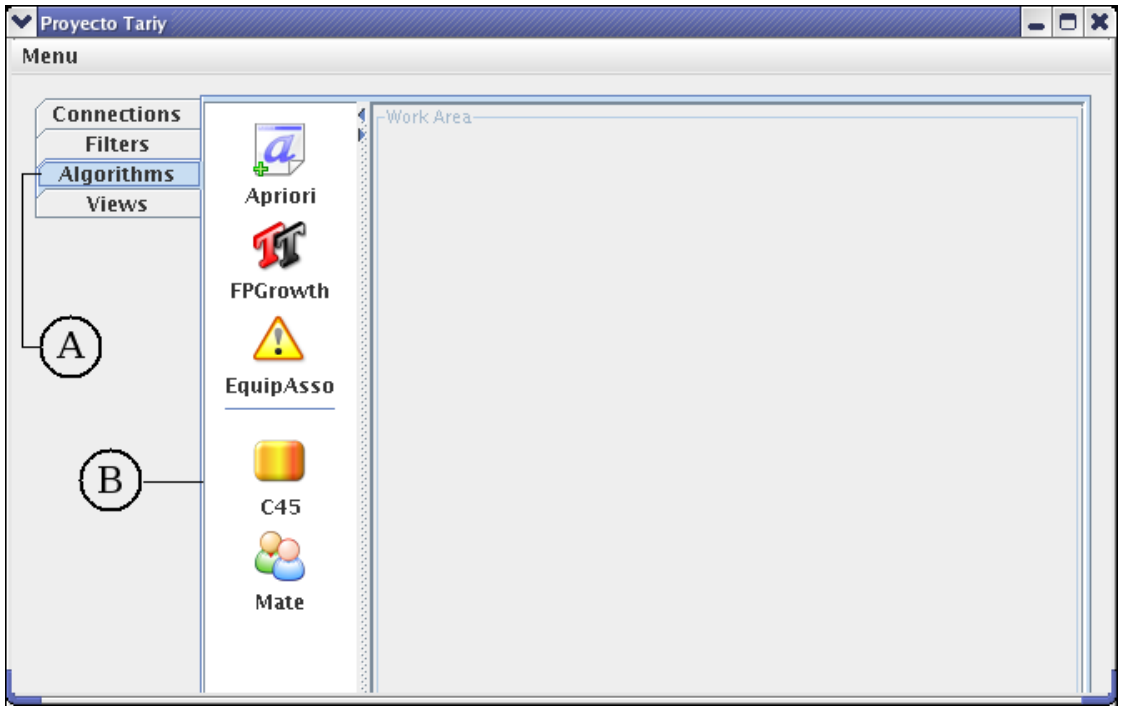

Figura 7.85: Módulo algoritmos

ACCIÓN DEL ACTOR	RESPUESTA DEL SISTEMA
1. El usuario hace click en la pestaña A 'Algoritmos'	2. Aparecen B Las opciones del módulo 'Algoritmos'

Módulo visualización

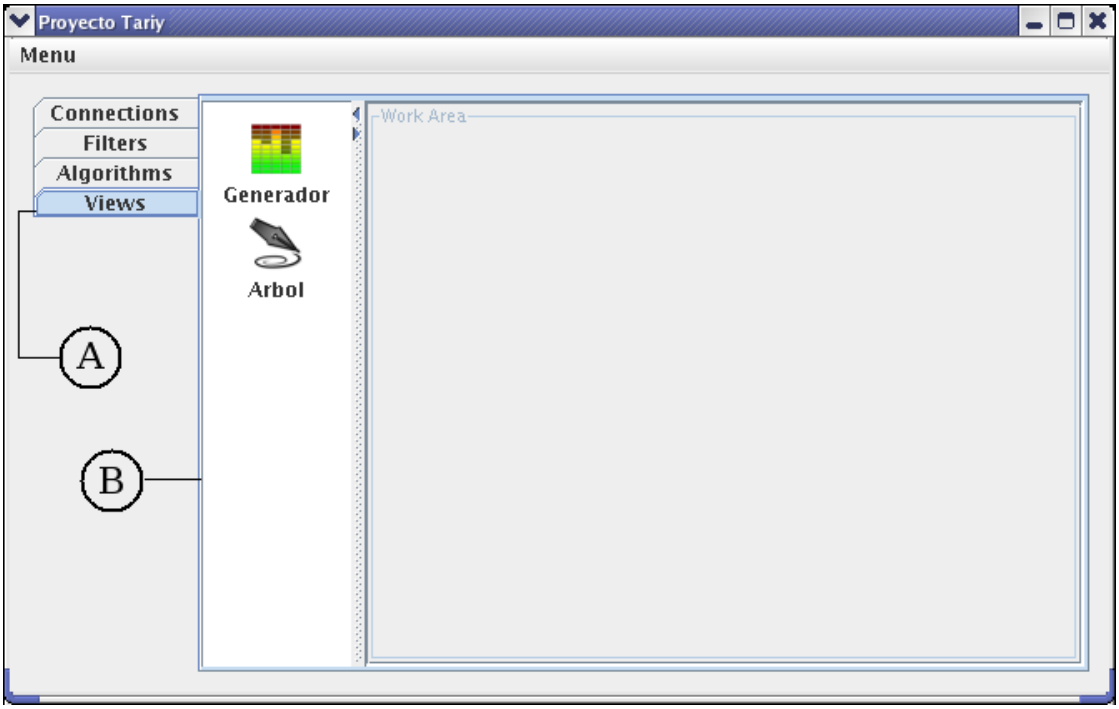

Figura 7.86: Módulo visualización

ACCIï¿½ DEL ACTOR	RESPUESTA DEL SISTEMA
1. El usuario hace click en la pestaña A 'Visualización'	2. Aparecen B Las opciones del módulo 'Visualización'

Conexión a un archivo plano

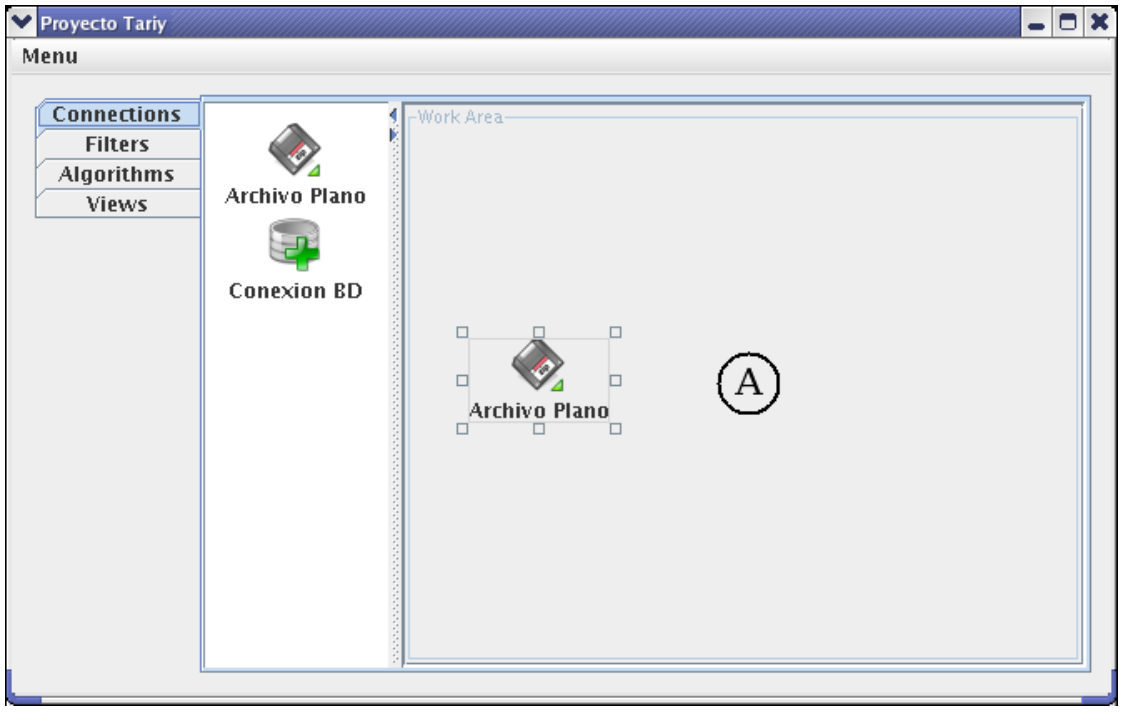

Figura 7.87: Conexión a un archivo plano

ACCIÓN DEL ACTOR	RESPUESTA DEL SISTEMA
1. El usuario hace click sobre el ícono 'Archivo de Texto'	2. El ícono 'Archivo de Texto' aparece sobre A: área de trabajo.

Conexión a una base de datos

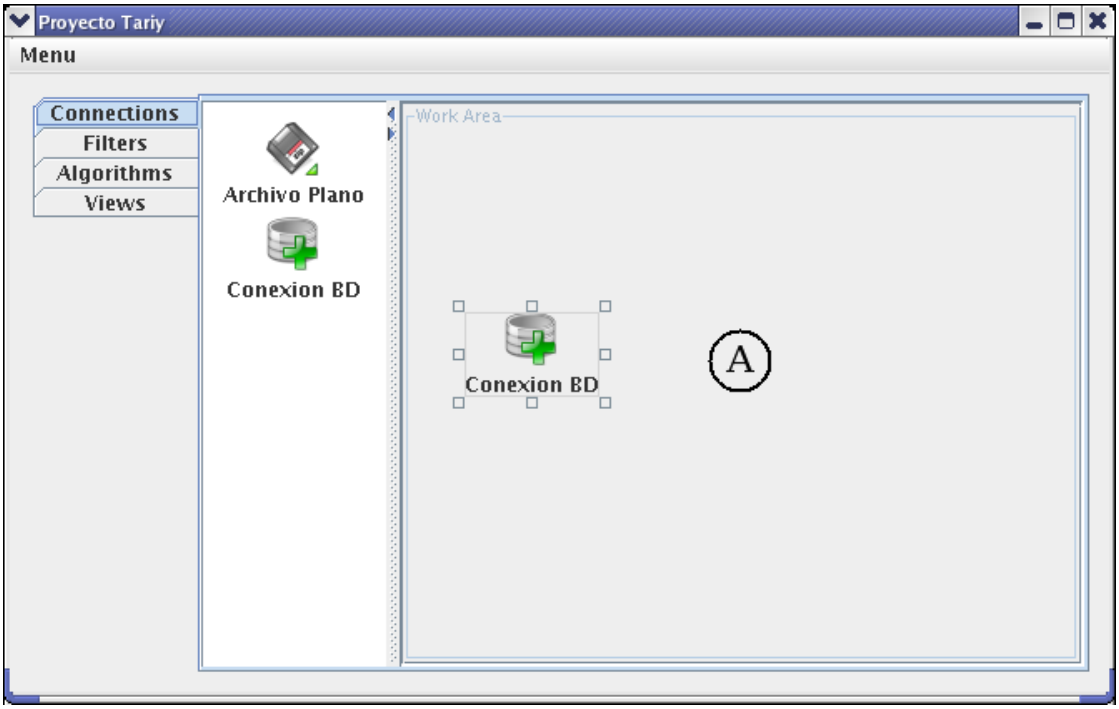

Figura 7.88: Conexión a una base de datos

ACCIÓN DEL ACTOR	RESPUESTA DEL SISTEMA
1. El usuario hace click sobre el ícono 'Conexión BD'	2. El ícono 'Conexión BD' aparece sobre A: área de trabajo.

Menú emergente conexión BD

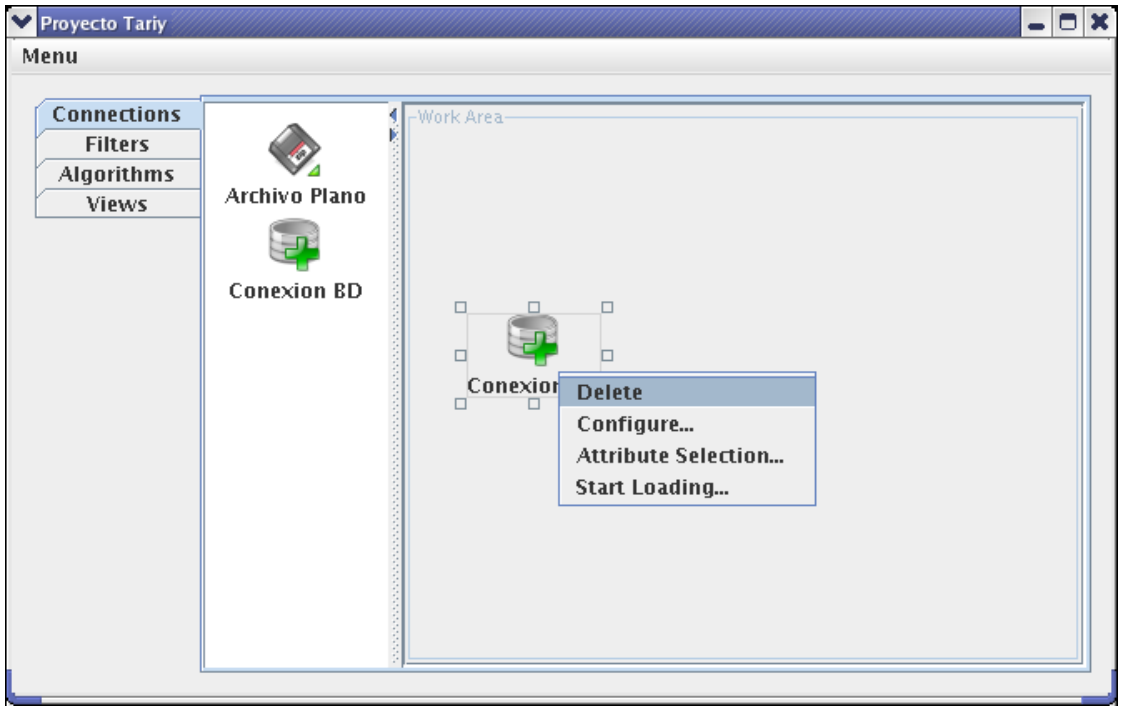

Figura 7.89: Menú emergente conexión BD

ACCIï¿½ DEL ACTOR	RESPUESTA DEL SISTEMA
1. El usuario hace click derecho sobre el ícono 'Conexión BD'	2. Se depliega A: menu del ícono 'Conexión BD'. Las opciones son: 'Delete': usada para eliminar el ícono del área de trabajo. 'Configure': usada para configurar la conexión a una base de datos. 'Selección de atributos': usada para seleccionar de forma gráfica los datos que sería usados más adelante. 'Cargar': ejecuta el query que se generá en la selección de atributos

Configuración conexión BD

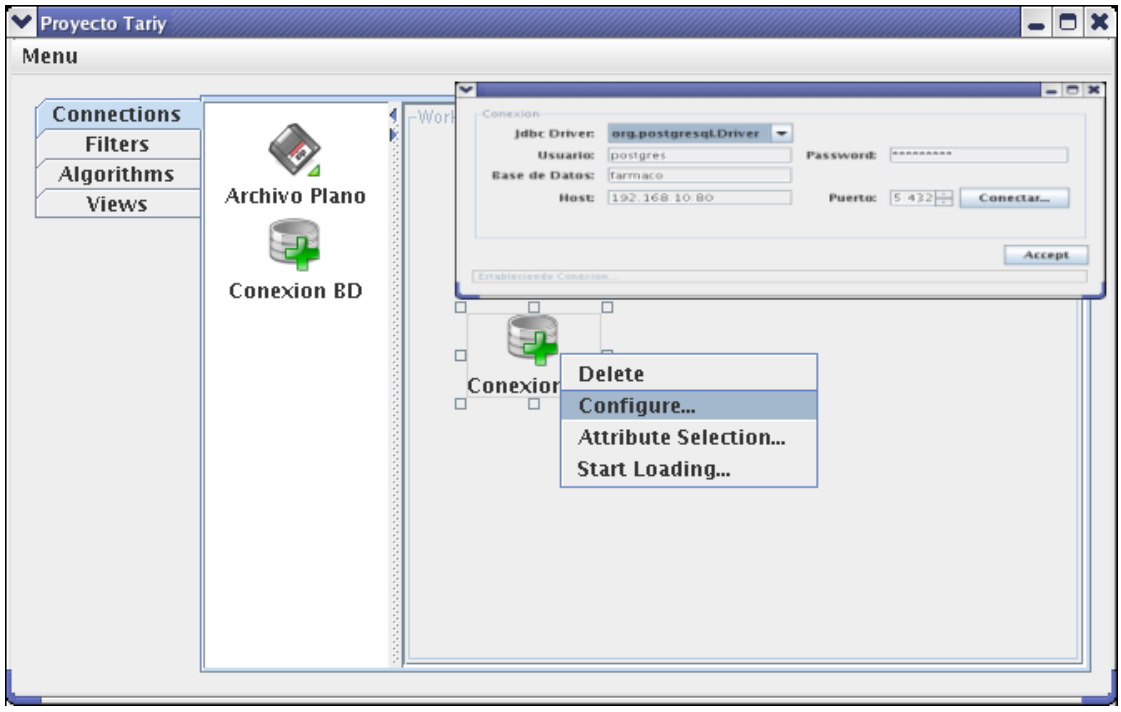

Figura 7.90: Configuración conexión BD

ACCIÓN DEL ACTOR	RESPUESTA DEL SISTEMA
1. El usuario hace click derecho sobre el ícono 'Conexión BD' y selecciona la opción 'Configure'	2. Emerge una ventana de configuración de conexión a bases de datos

Ventana de conexión BD

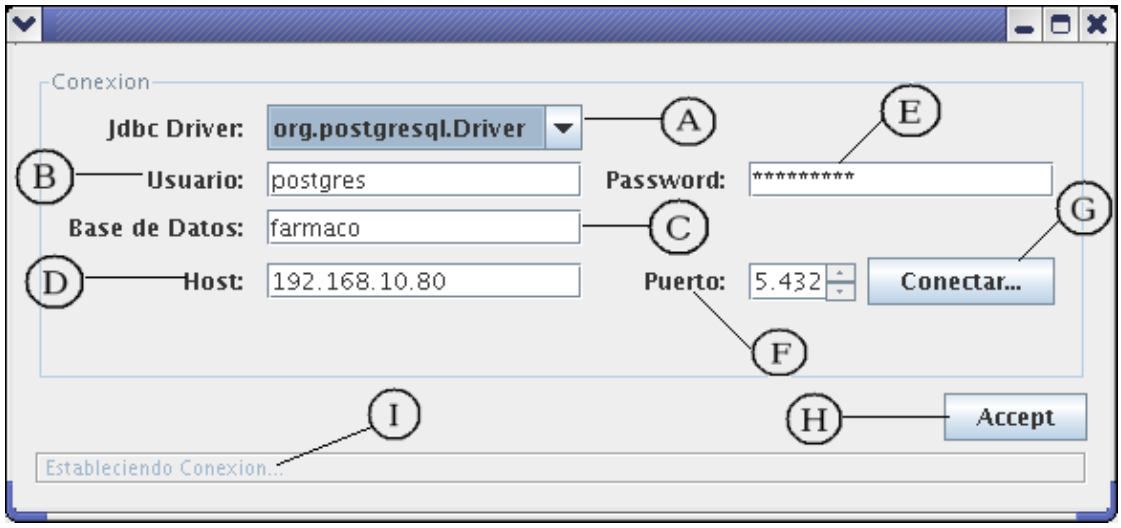

Figura 7.91: Ventana de conexión BD

ACCIÓN DEL ACTOR	RESPUESTA DEL SISTEMA
1. El usuario desea configurar la conexión a una base de datos'	2. Las opciones de la ventana de configuración de conexión a bases de datos tiene los siguientes campos: A: Lista de controladores ODBC para varios tipos de bases de datos. B: Nombre del usuario de la base de datos. C: Nombre de la base de datos. D: Nombre del servidor. E: 'Password': clave de acceso a la base de datos. F: nmero del puerto utilizado para la comunicación con la base de datos. G: botón de conexión. H: botón para aceptar la conexión hecha. I: mensaje que indica el estado de la conexión.

Selección de atributos

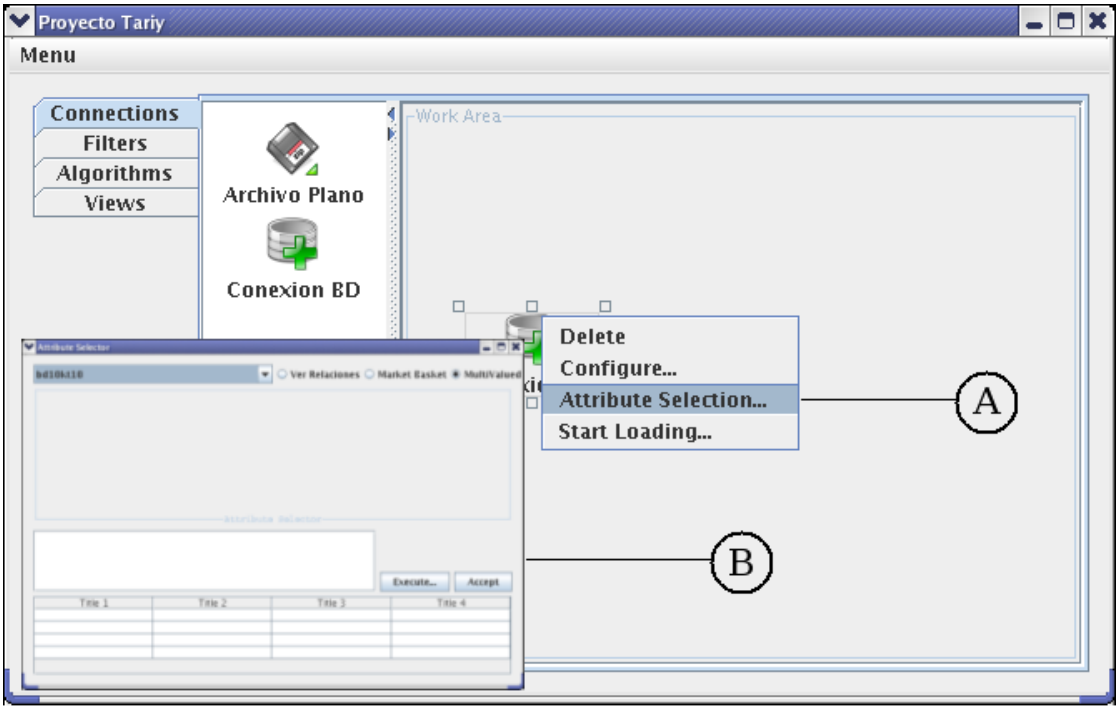

Figura 7.92: Selección de atributos

ACCIÓN DEL ACTOR	RESPUESTA DEL SISTEMA
1. El usuario hace click derecho sobre el ícono 'Conexión BD' para hacer la selección de atributos	2. Aparece el menú emergente del ícono.y se ejecuta la ventana de selección de atributos A.
3. El usuario hace click en la opción B: 'Selección de Atributos'	4. Aparece la ventana de selección de atributos B.

Ventana selección de atributos

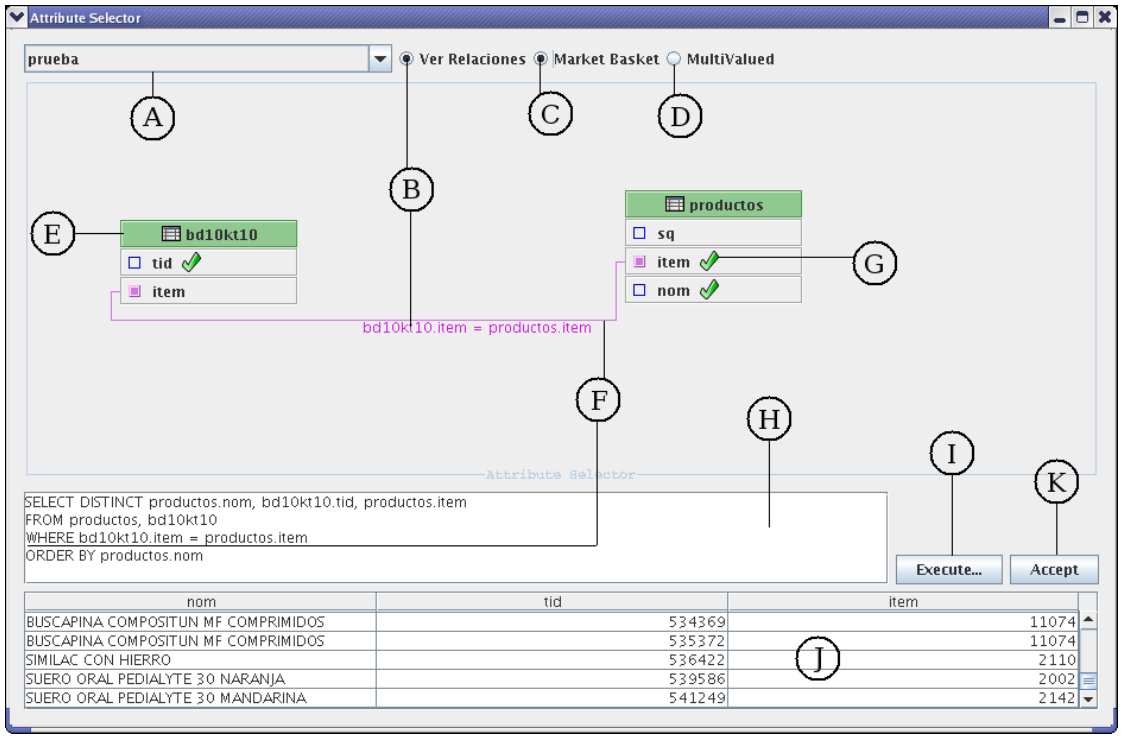

Figura 7.93: Ventana selección de atributos

ACCIÓN DEL ACTOR	RESPUESTA DEL SISTEMA
1. El usuario desea hacer la selección de atributos	2. Aparece la ventana de selección de atributos. A: lista desplegable de las tablas de la base de datos a la que se ha conectado. Al seleccionar una de ellas su representación gráfica aparecera en el espacio de trabajo E. B: opción que permite ver las relaciones establecidas a través de la línea de conexión de atributos entre las tablas. C: esta opción es útil cuando se trabajan problemas de canasta de mercado. D: opción para trabajar tablas multivaluadas. F:línea que permite realizar las relaciones entre atributos de dos tablas. El resultado de la relación establecida se refleja en el query. G:Si se hace click sobre uno de los atributos aparece un ícono de verificación que indica los campos que serán mostrados al ejecutar el query. H: espacio en el que se crea el query. Es posible editarlo manualmente. I: botón de ejecución del query. J: tabla en la que se muestra el resultado de la ejecución del query. K: botón para aceptar las operaciones realizadas.

Filtro Remove Missing

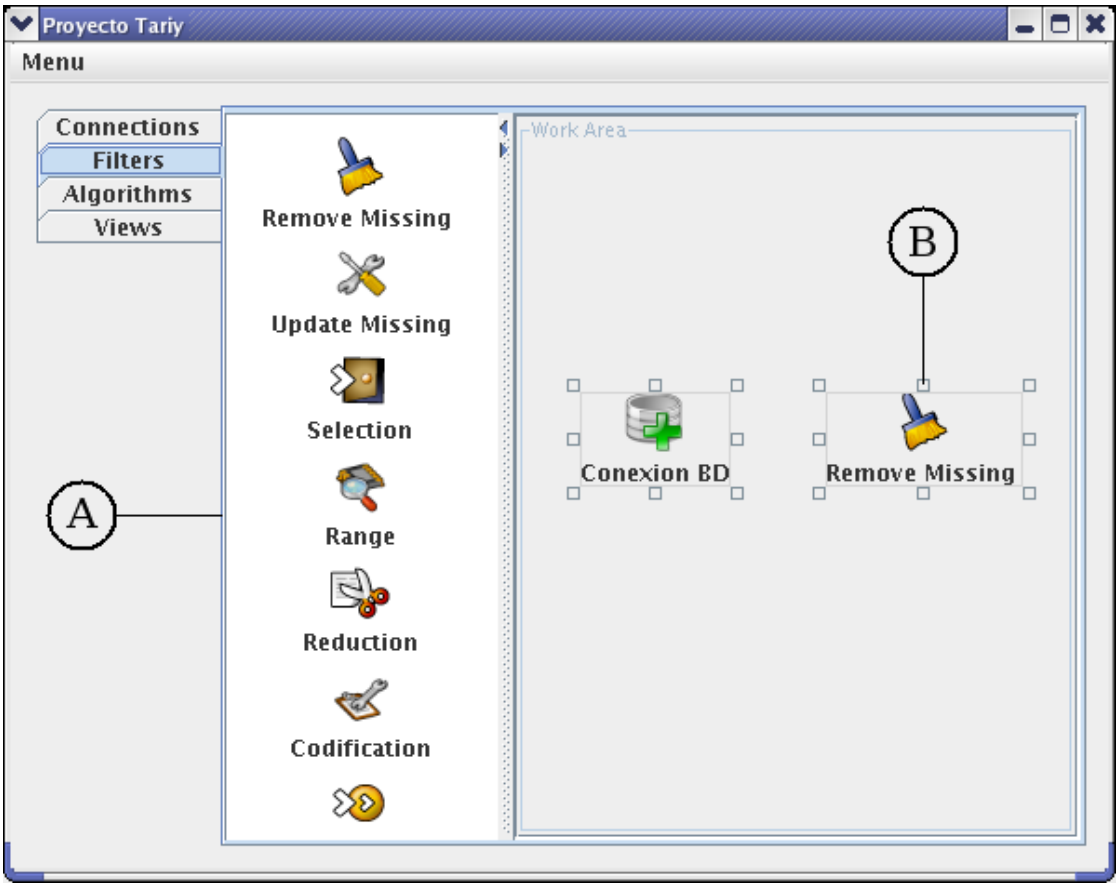

Figura 7.94: Filtro Remove Missing

ACCIÓN DEL ACTOR	RESPUESTA DEL SISTEMA
1. El usuario hace click sobre uno de los íconos del mï¿½ulo A: 'Filtros'.	2. En el área de opciones del módulo aparecen los 9 íconos correspondientes a los filtros
3. El usuario hace click sobre uno de los íconos correspondientes a los filtros.	4. El ícono correspondiente aparace en el área de trabajo B.

Conexión filtros a BD

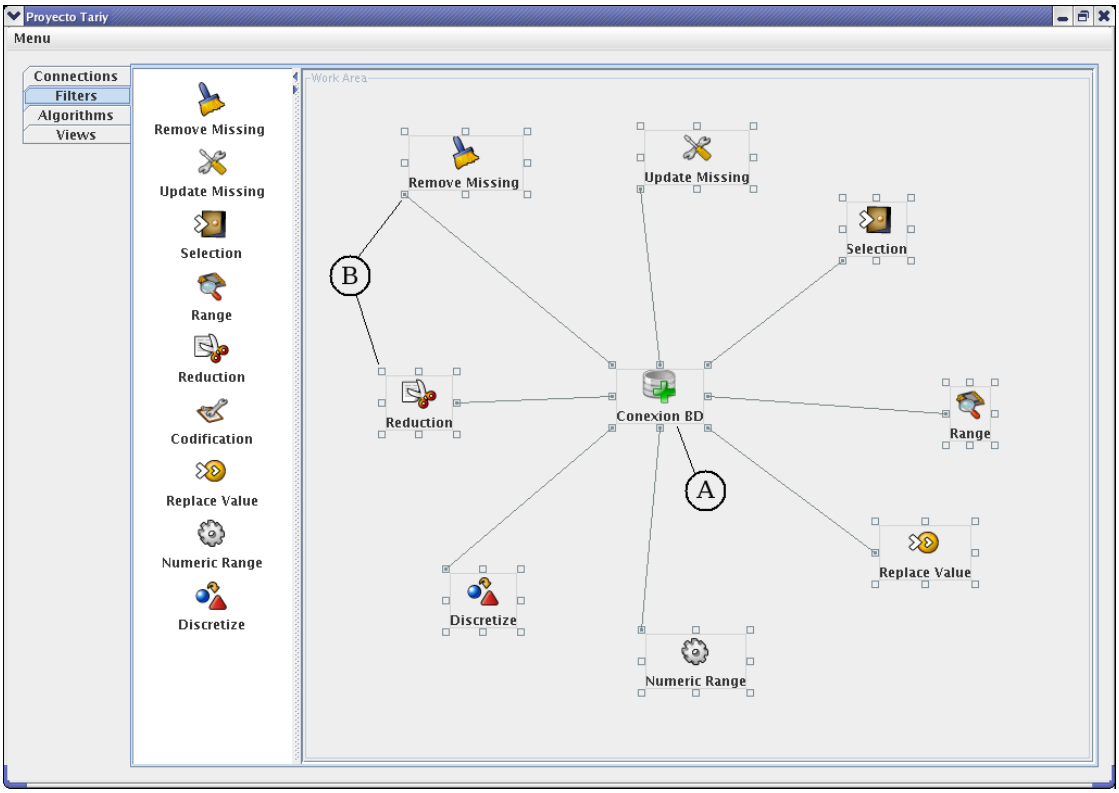

Figura 7.95: Conexión filtros a BD

ACCIÓN DEL ACTOR	RESPUESTA DEL SISTEMA
1. El usuario conecta una base de datos a alguno o varios de de los filtros A.	2. Los íconos pueden ser conectados por medio de una línea B.

Menú emergente de filtros

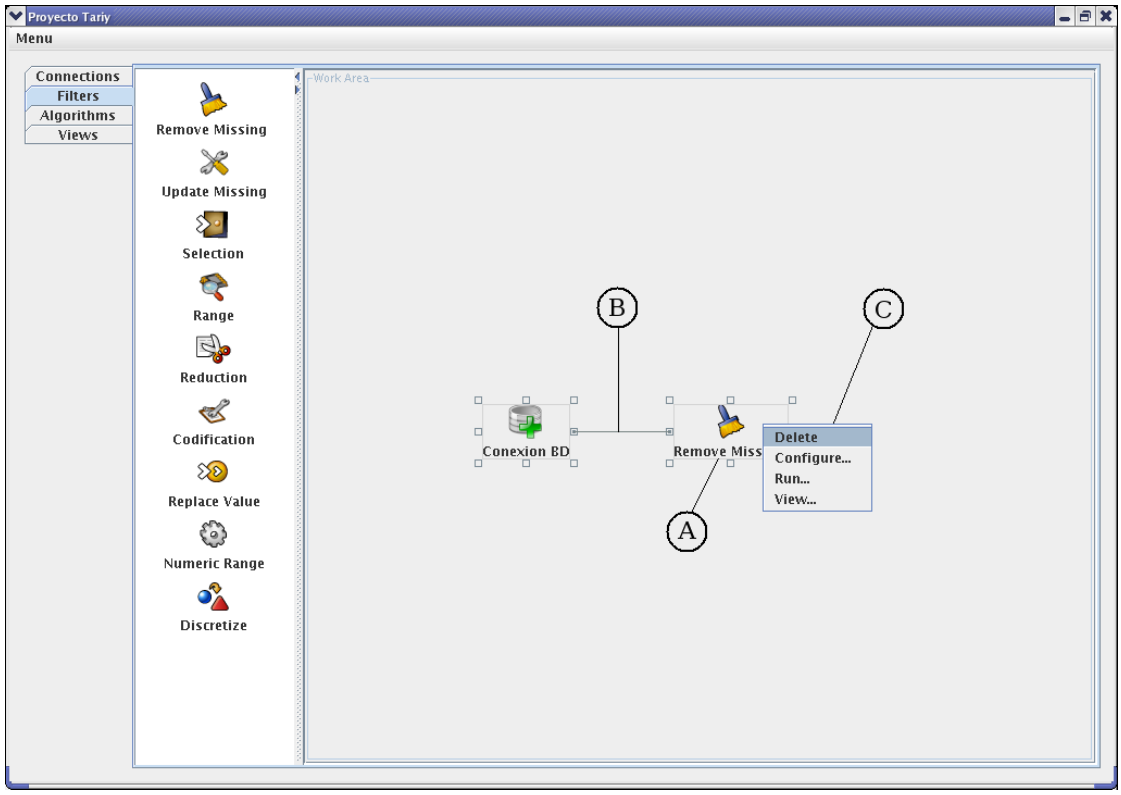

Figura 7.96: Menú emergente de filtros

ACCIÓN DEL ACTOR	RESPUESTA DEL SISTEMA
1. El usuario hace click sobre el ícono 'Remove Missing'.	2. El ícono aparece en el área de trabajo A.
3. El usuario conecta el filtro a la base de datos.	4. Aparece un hilo que conecta los íconos B.
5. EL usuario hace click derecho sobre el filtro.	6. Aparece el menu emergente del ícono C. La opción Delete, borra el filtro del área de trabajo. Este filtro no tiene ventana de configuración. La opción 'Run' ejecuta la aplicación del filtro. La opción 'View' muestra la ventana de vizualización de datos que sería descrita en el siguiente caso de uso

Visualización de datos filtrados

Figura 7.97: Visualización de datos filtrados

Figura 7.98: caso nueve

ACCIÓN DEL ACTOR	RESPUESTA DEL SISTEMA
1. El usuario hace click sobre la opción 'View' del menu desplegable filtro en el área de trabajo .	2. Aparece la ventana de vizualización de datos filtrados y no filtrados. Los campos son, A: Variables o nombres de los campos de la tabla. B: Datos de entrada que son los datos que llegaron al filtro inicialmente. C: Datos filtrados que son el resultado de haber aplicado el filtro. D: nmero de registros eliminados al aplicar el filtro. E: Nmero de registros después de aplicar el filtro. En la figura 16 se ve la grilla sobre la que se muetran los datos en el caso 'Datos de entrada'

Configuración filtro Update Missing

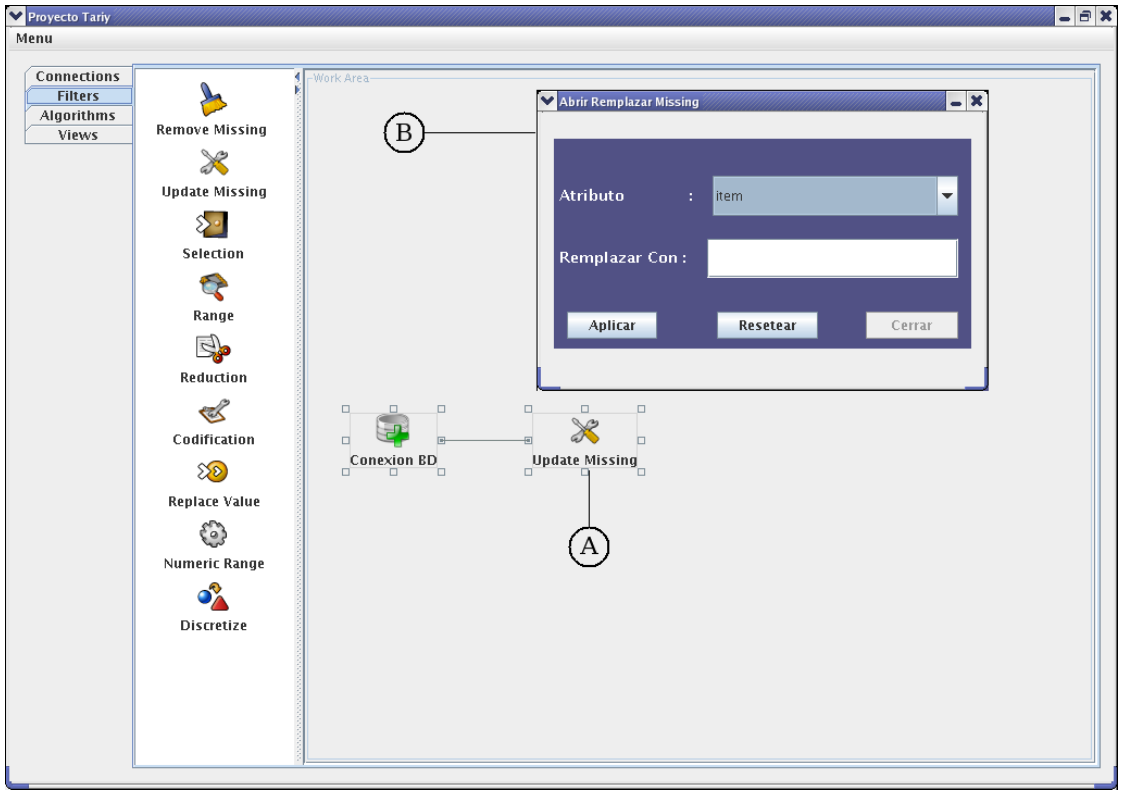

Figura 7.99: Configuración filtro Update Missing

ACCIÓN DEL ACTOR	RESPUESTA DEL SISTEMA
1. El usuario hace click derecho sobre el filtro A y elige la opción 'Configuración'	2. Se muestra B la ventana de configuración correspondiente al filtro 'Update Missing'. Los campos son: Atributo, en el cual se escribe el nombre del atributo a buscar en el conjunto de datos. Reemplazar con, aqui se escribe el nuevo valor del atributo

Configuración filtro Selection

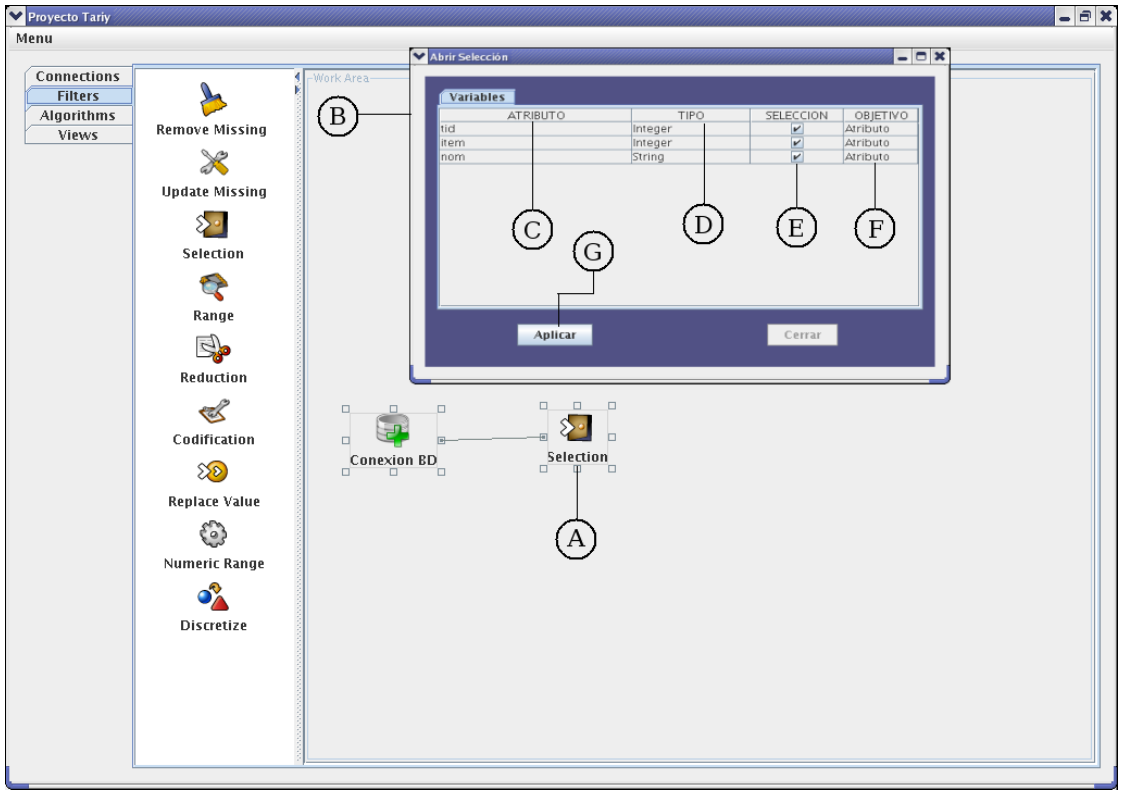

Figura 7.100: Configuración filtro Selection

ACCIÓN DEL ACTOR	RESPUESTA DEL SISTEMA
1. El usuario hace click derecho sobre el filtro A y elige la opción 'Configuración'	2. Se muestra B la ventana de configuración correspondiente al filtro 'Selection'. Los campos son: C: Atributo, en esta grilla se muestran los nombres de los atributos seleccionados. D: Tipo, muestra el tipo de datos de los atributos. E: cajas de verificación para escoger los atributos a utilizar. F: es posible escoger un atributo clase haciendo click sobre estos campos. Esto es útil en experimentos de clasificación. G: el botón 'Aplicar' debe ser precionado para que el filtro sea aplicado.

Configuración filtro Range

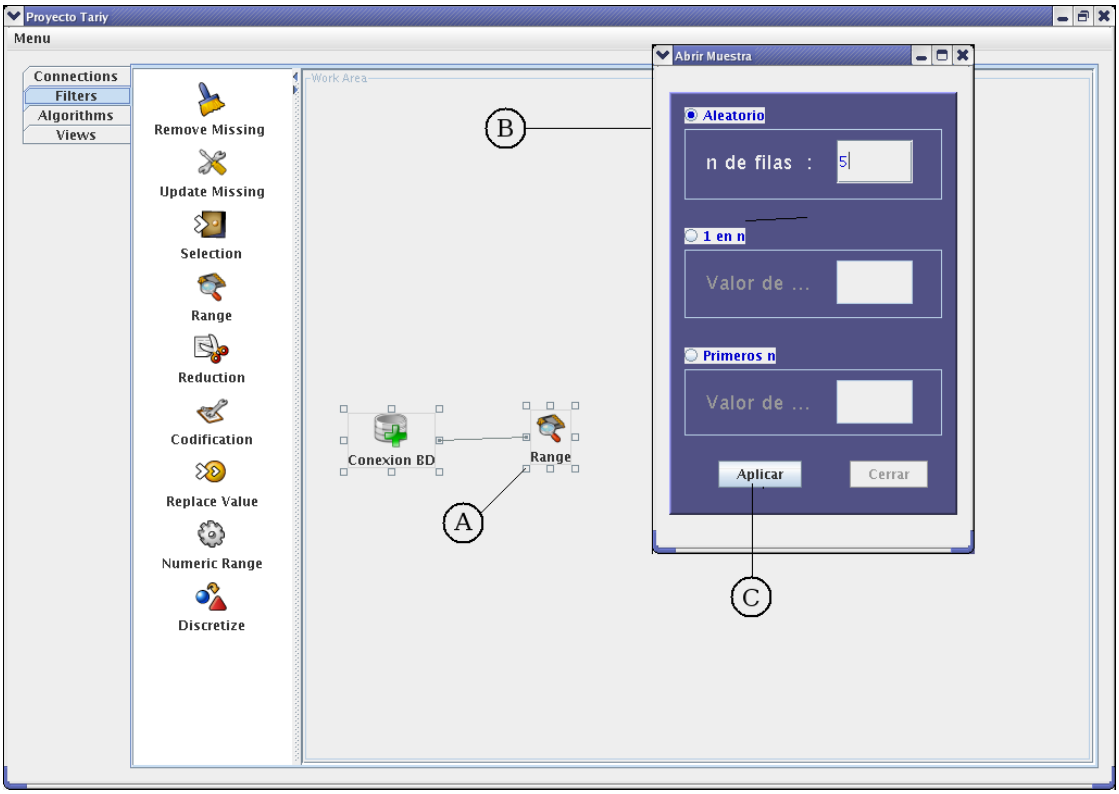

Figura 7.101: Configuración filtro Range

ACCIÓN DEL ACTOR	RESPUESTA DEL SISTEMA
1. El usuario hace click derecho sobre el filtro A y elige la opción 'Configuración'	2. Se muestra B la ventana de configuración correspondiente al filtro 'Range'. Los campos son:**Aleatorio**, en donde se escribe el número **n** de filas que se desea sean escogidas aleatoriamente. **1 en n**, donde **n** es el periodo utilizado para seleccionar los datos a utilizar. **Primeros n**, donde **n**es el nmero campos a incluir en la selección a partir del primero.

Configuración filtro Reduction

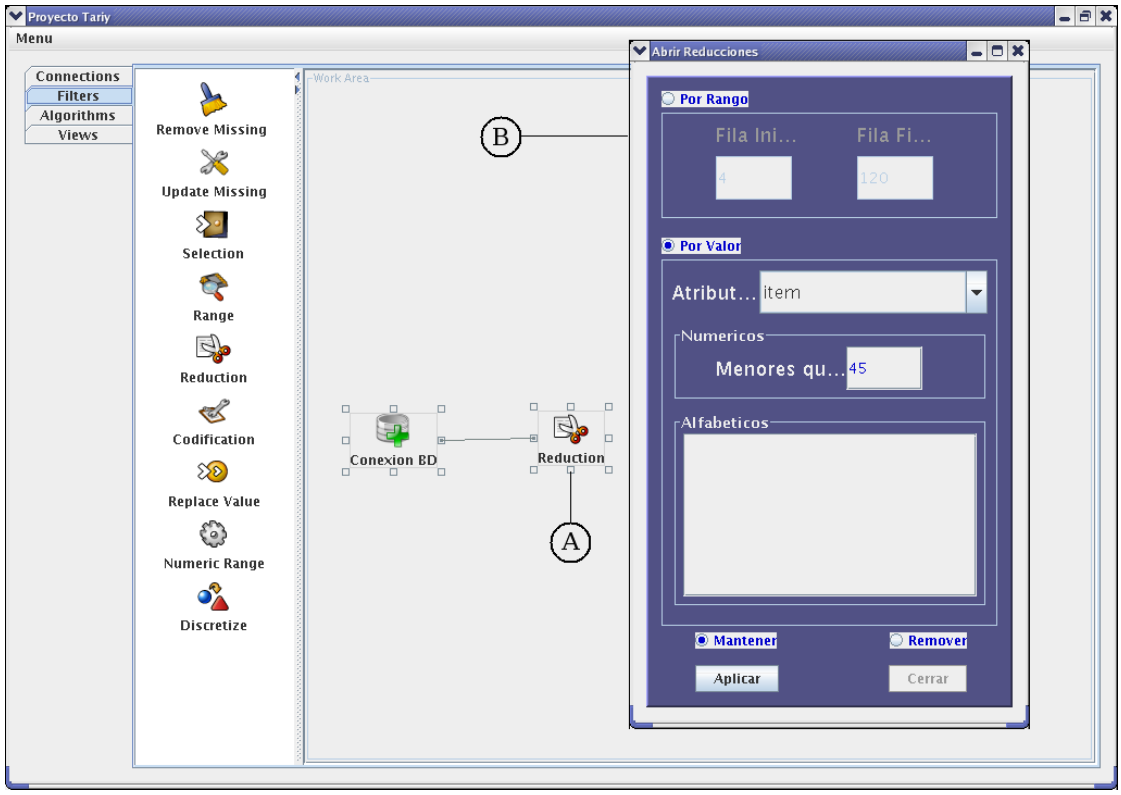

Figura 7.102: Configuración filtro Reduction

ACCIÓN DEL ACTOR	RESPUESTA DEL SISTEMA
1. El usuario hace click derecho sobre el filtro A y elige la opción 'Configuración'	2. Se muestra la ventana B de configuración correspondiente al filtro 'Reduction'. Los campos son: **Por rango**, los campos son 'Fila ini cial' donde se escribe la fila a partir de la cual inicia el rango y 'Fila final' que es el límite superior del rango. **Por Valor:** Se elige el nombre del atributo y luego en caso de que los valores a quitar sean numéricos en el campo 'Menores que' se especifica el nmero a partir del cual se hace la reducción. Si el atributo es alfabético se escribe su valor en el área de texto y en las casillas de selección se especifica si ese valor se desea 'Mantener' o 'Remover'. **Aplicar**: ejecuta el filtro.

Configuración filtro Codification

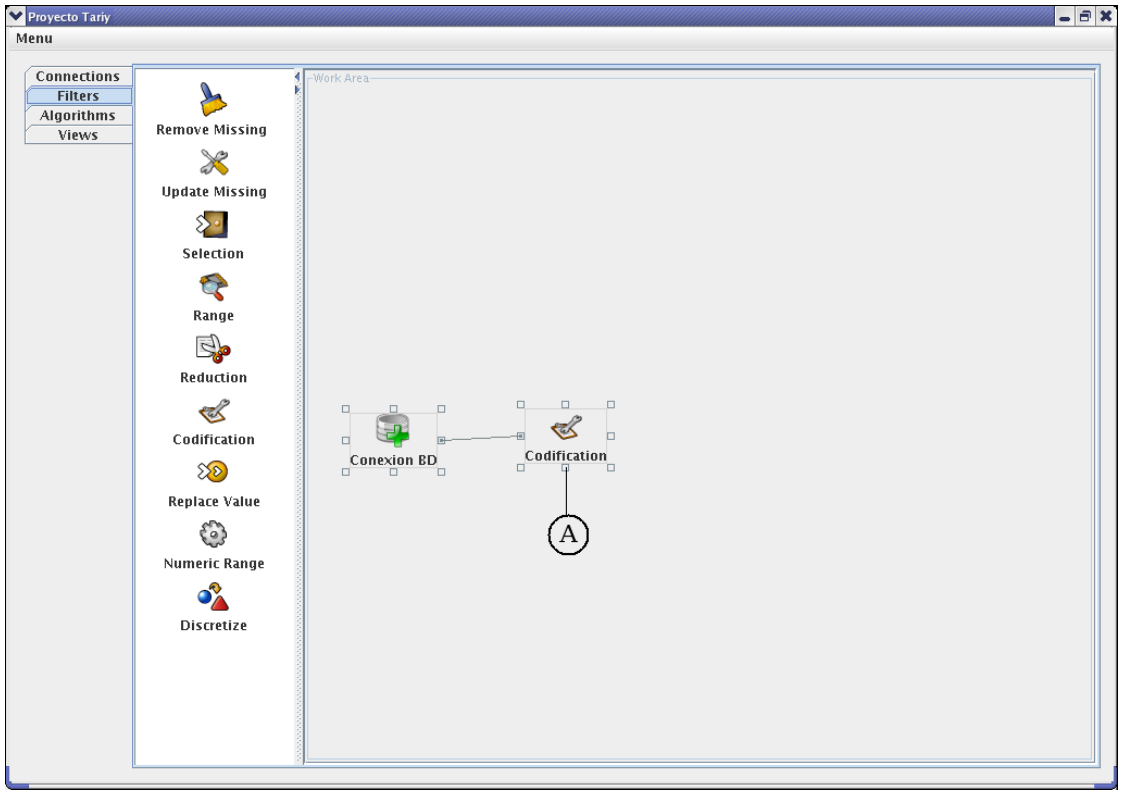

Figura 7.103: Configuración filtro Codification

ACCIÓN DEL ACTOR	RESPUESTA DEL SISTEMA
1. El usuario hace click derecho sobre el filtro A y elige la opción 'Configuración'	2. Se muestra la ventana de configuración correspondiente al filtro 'Codifcation'. Este filtro no tiene ventana de configuración. Se aplica para asignar un número a valores alfabéticos

Configuración filtro Replace Value

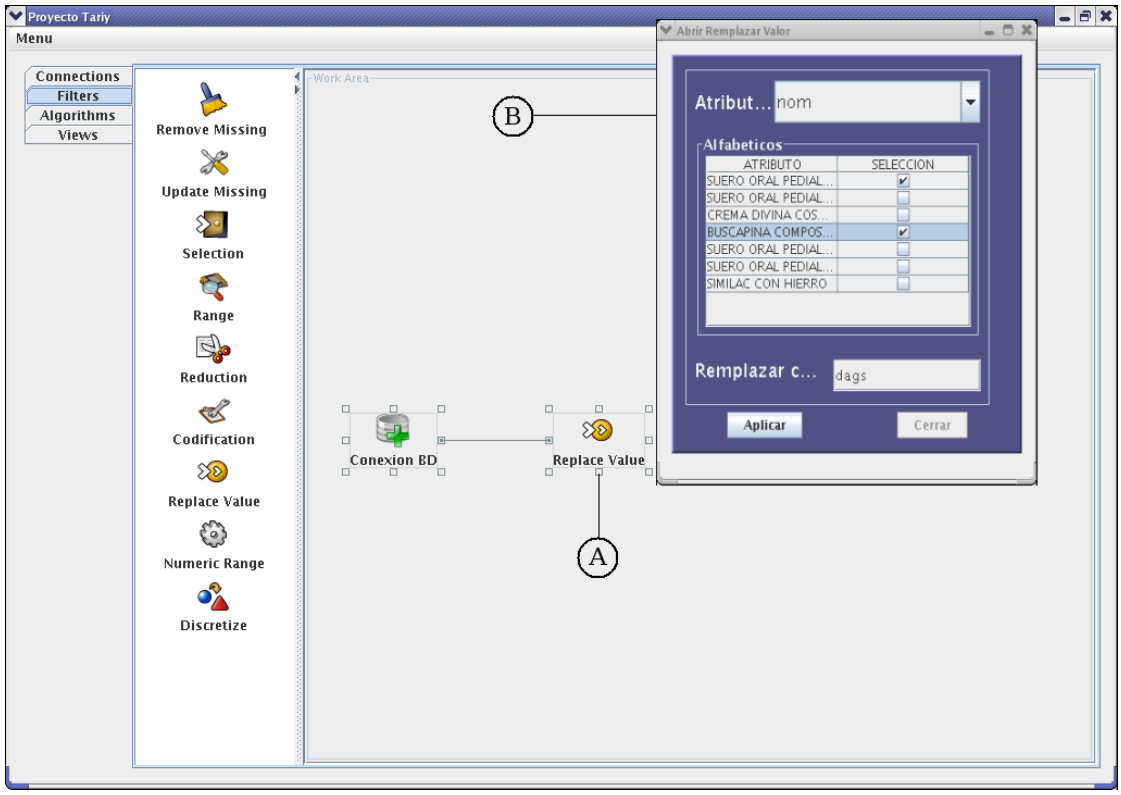

Figura 7.104: Configuración filtro Replace Value

ACCIÓN DEL ACTOR	RESPUESTA DEL SISTEMA
1. El usuario hace click derecho sobre el filtro A y elige la opción 'Configuración'	2. Se muestra la ventana de configuración correspondiente al filtro 'Replace Value'. Los campos son: Atributo, en el cual se elige el nombre del atributo a buscar en el conjunto de datos. Reemplazar con, aqui se escribe el nuevo valor del atributo. **Aplicar**: ejecuta el filtro.

Configuración filtro Numeric Range

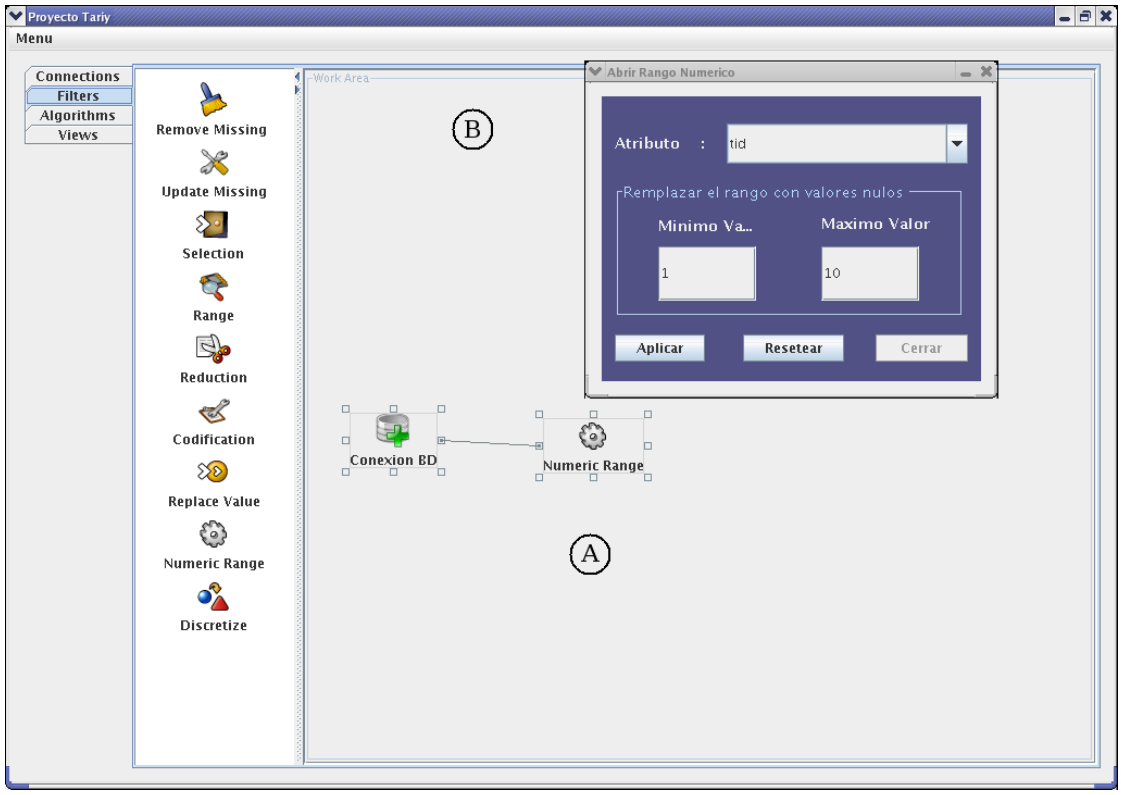

Figura 7.105: Configuración filtro Numeric Range

ACCIÓN DEL ACTOR	RESPUESTA DEL SISTEMA
1. El usuario hace click derecho sobre el filtro A y elige la opción 'Configuración'	2. Se muestra la ventana B de configuración correspondiente al filtro 'Numeric Range'. Los campos son: **Atributo**, en el cual se escribe el nombre del atributo a discretizar de tipo numérico. **Reemplazar rango con valores nulos**: aqui es posible especificar un rango de datos que seran convertidos a núlos. **Mímimo valor**: límite inferiror del rango. **Mínimo valor**: límite superior del rango. **Aplicar**: ejecuta el filtro. **Resetear**: deja los campos en blanco

Configuración filtro Discretize

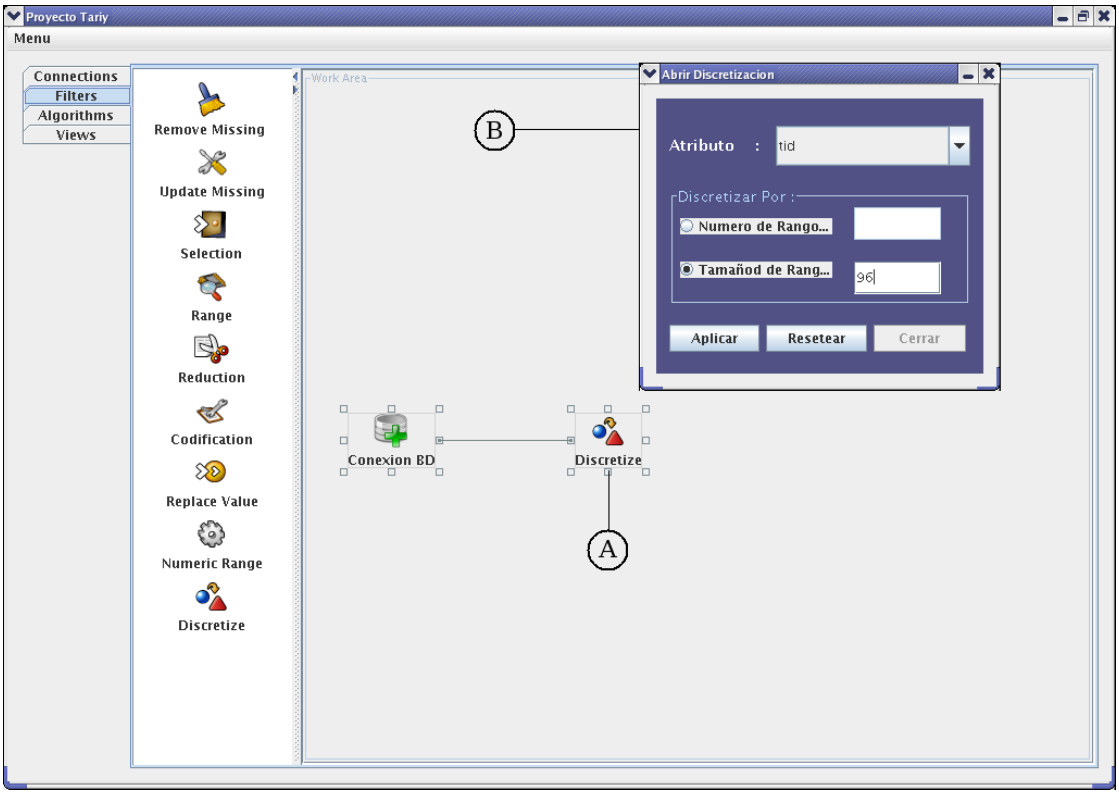

Figura 7.106: Configuración filtro Discretize

ACCIÓN DEL ACTOR	RESPUESTA DEL SISTEMA
1. El usuario hace click derecho sobre el filtro A y elige la opción 'Configuración'	2. Se muestra la ventana B de configuración correspondiente al filtro 'Discretize'. Los campos son: **Atributo**, en el cual se escribe el nombre del atributo a discretizar. **Discretizar por**: 'Número de rango': se puede establecer el número de rangos a crear. 'Tamaño del rango': se especifica el tamaño del rango **Aplicar**: ejecuta el filtro. **Resetear**: deja los campos en blanco

Algoritmos

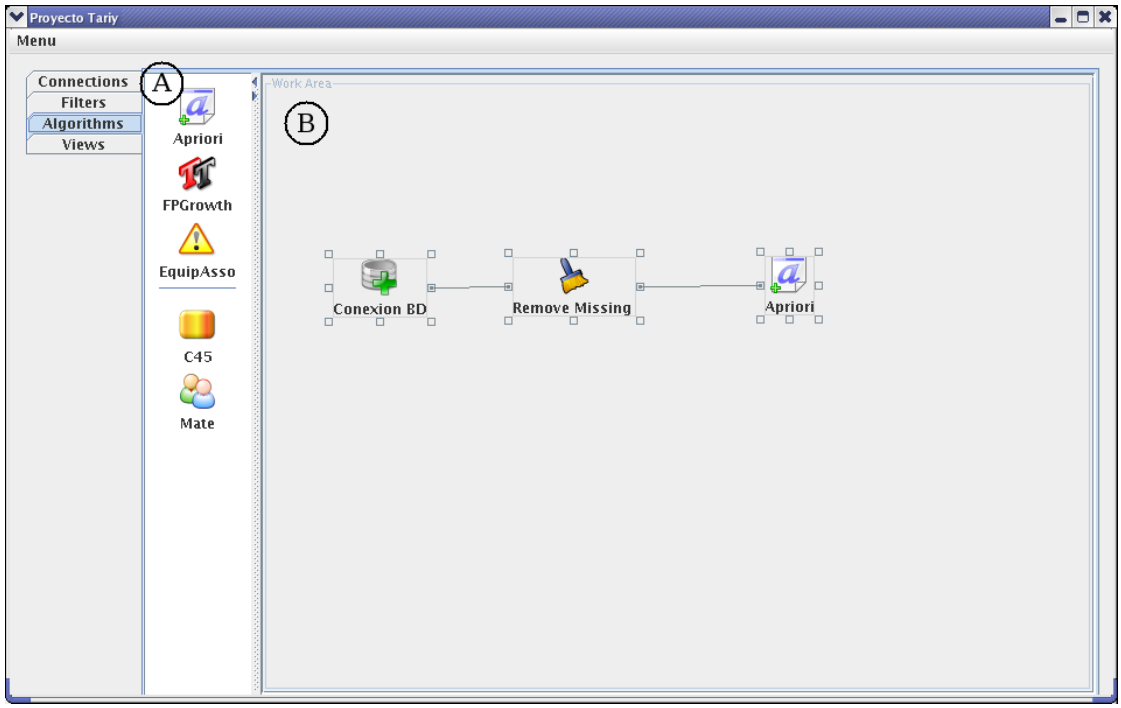

Figura 7.107: Algoritmo Apriori

ACCIÓN DEL ACTOR	RESPUESTA DEL SISTEMA
1. Si el usuario quiere minar los datos con Apriori y presiona sobre A (Área de opciones), en el icono respectivo.	2. En B (Área de trabajo) aparece el icono del algoritmo Apriori.

Opción Delete

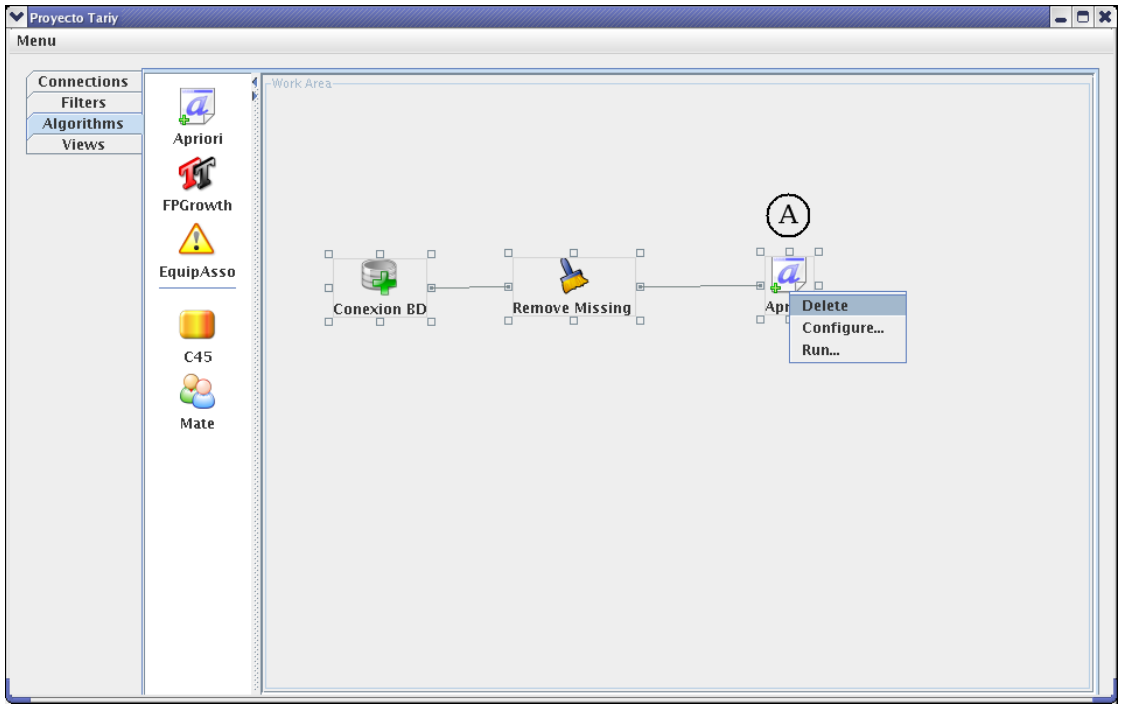

Figura 7.108: Opción Delete

ACCIÓN DEL ACTOR	RESPUESTA DEL SISTEMA
1. El usuario hace click derecho sobre A: el icono del algoritmo (cualquiera que este sea, Apriori, EquipAsso, FPGrowth, MateBy o C4.5) y elige la opción delete del menú de configuración.	2. El icono del algoritmo es borrado del área de trabajo.

Opción Configure

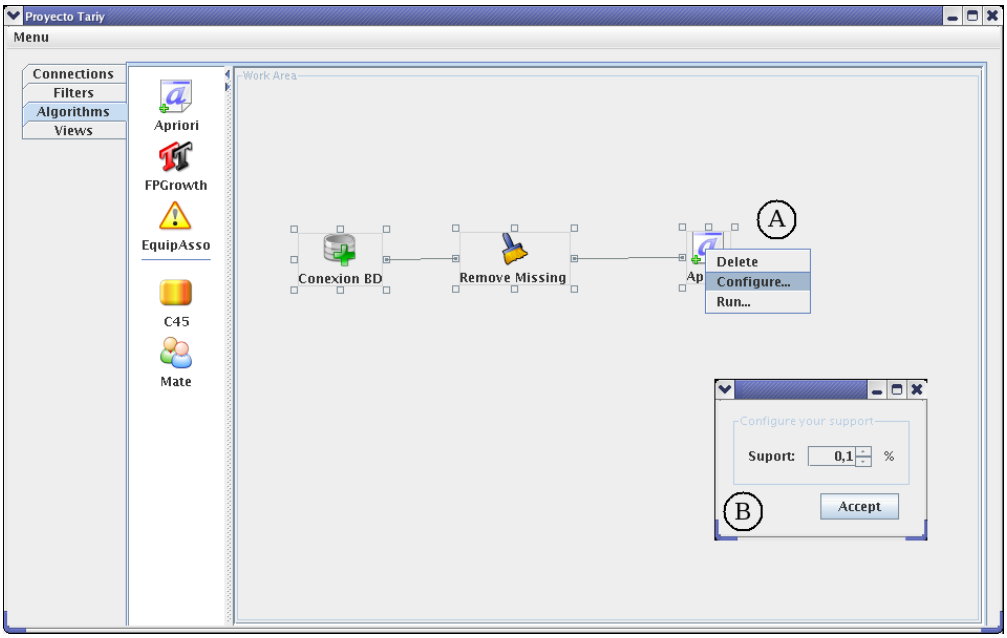

Figura 7.109: Opción Configure

ACCIÓN DEL ACTOR	RESPUESTA DEL SISTEMA
1. El usuario hace click derecho sobre A: el icono del algoritmo (cualquiera que este sea, Apriori, EquipAsso, FPGrowth, MateBy o C4.5) y elige configurar sus parametros.	2. Sobre el área de trabajo aparece una ventana B, para que el usuario configure el soporte del algoritmo.

Opción Run

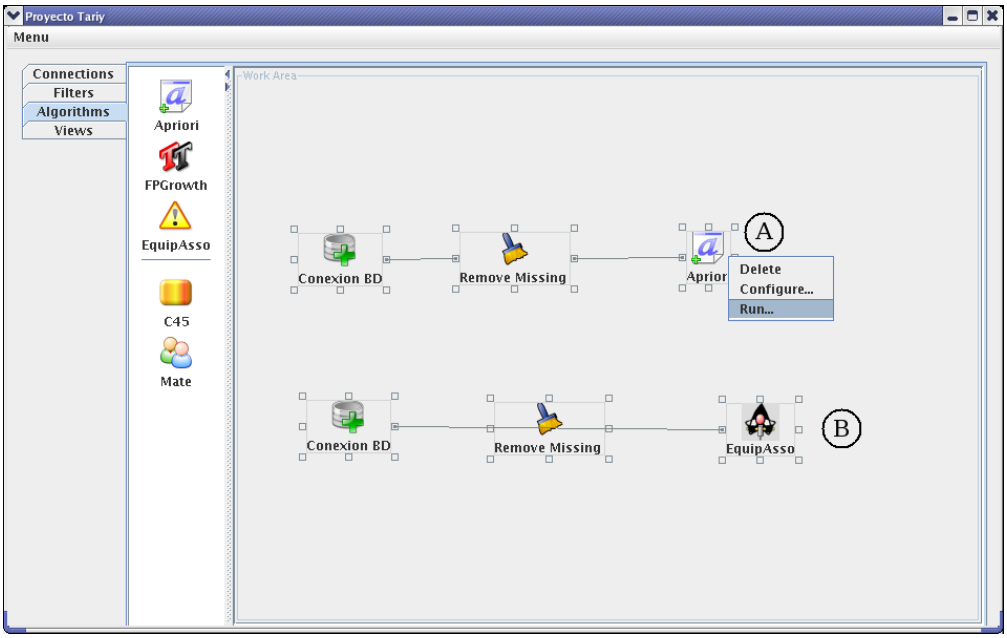

Figura 7.110: Opción Run

ACCIÓN DEL ACTOR	RESPUESTA DEL SISTEMA
1. El usuario hace click derecho sobre A: el icono del algoritmo (cualquiera que este sea, Apriori, EquipAsso, FPGrowth, MateBy o C4.5) y elige la opción run.	2. El icono del algoritmo cambia por una animación, así como se muestra en B.

Algoritmo FPGrowth

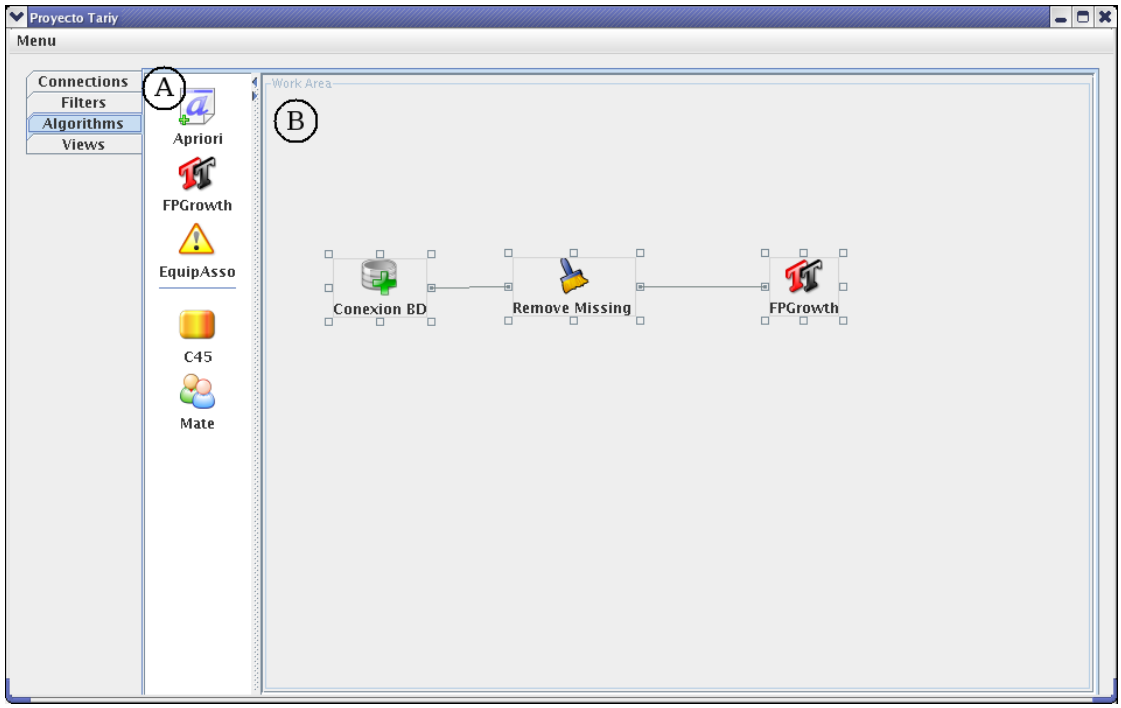

Figura 7.111: Algoritmo FPGrowth

ACCIÓN DEL ACTOR	RESPUESTA DEL SISTEMA
1. Si el usuario quiere minar los datos con FPGrowth y presiona sobre A (Área de opciones), en el icono respectivo.	2. En B (Área de trabajo) aparece el icono del algoritmo FPGrowth.

Algoritmo EquipAsso

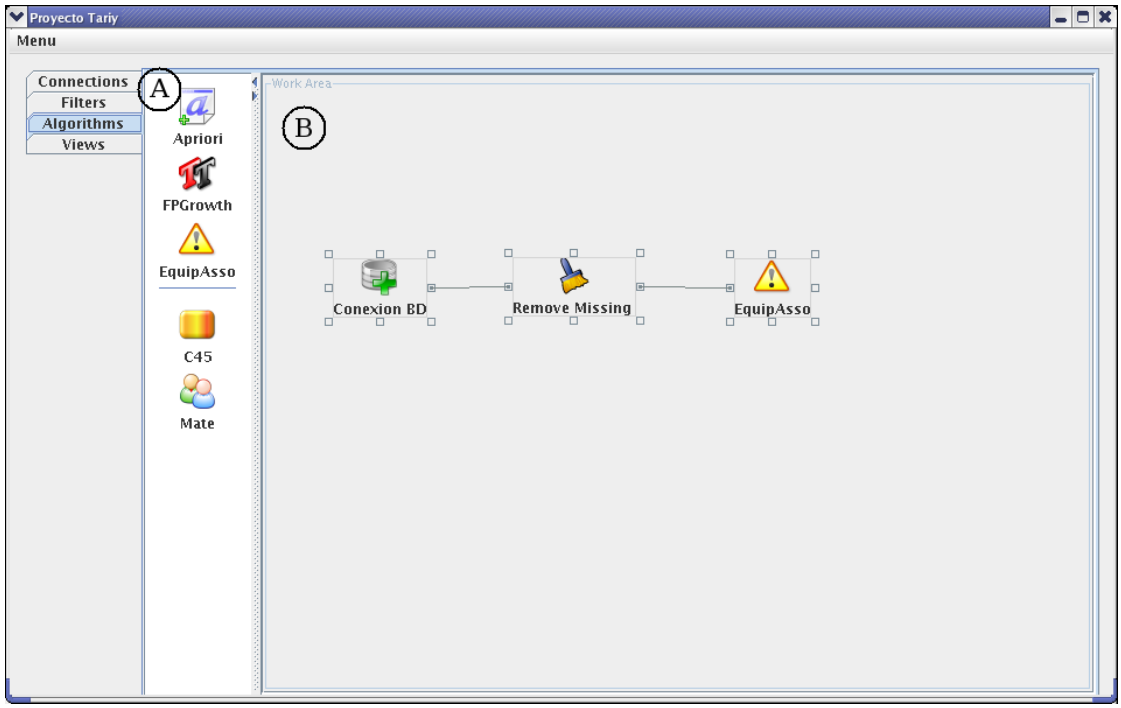

Figura 7.112: Algoritmo EquipAsso

ACCIÓN DEL ACTOR	RESPUESTA DEL SISTEMA
1. Si el usuario quiere minar los datos con EquipAsso y presiona sobre A (Área de opciones), en el icono respectivo.	2. En B (Área de trabajo) aparece el icono del algoritmo EquipAsso.

Algoritmo C4.5

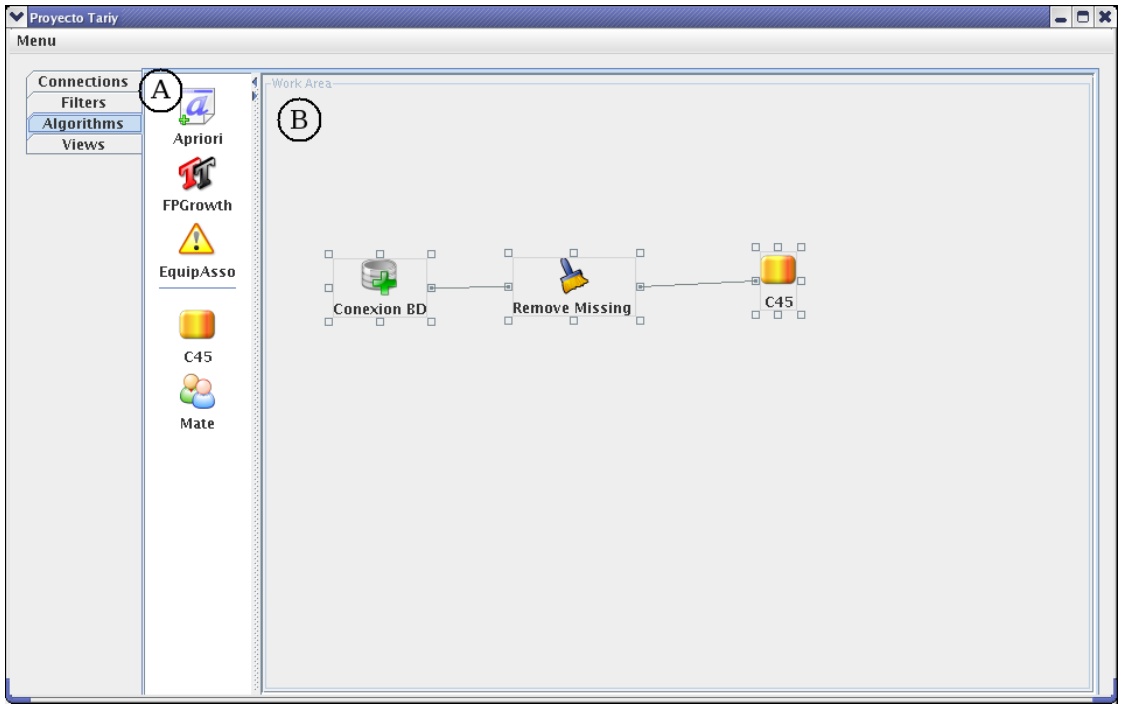

Figura 7.113: Algoritmo C4.5

ACCIÓN DEL ACTOR	RESPUESTA DEL SISTEMA
1. Si el usuario quiere minar los datos con C4.5 y presiona sobre A (Área de opciones), en el icono respectivo.	2. En B (Área de trabajo) aparece el icono del algoritmo C4.5.

Algoritmo Mate

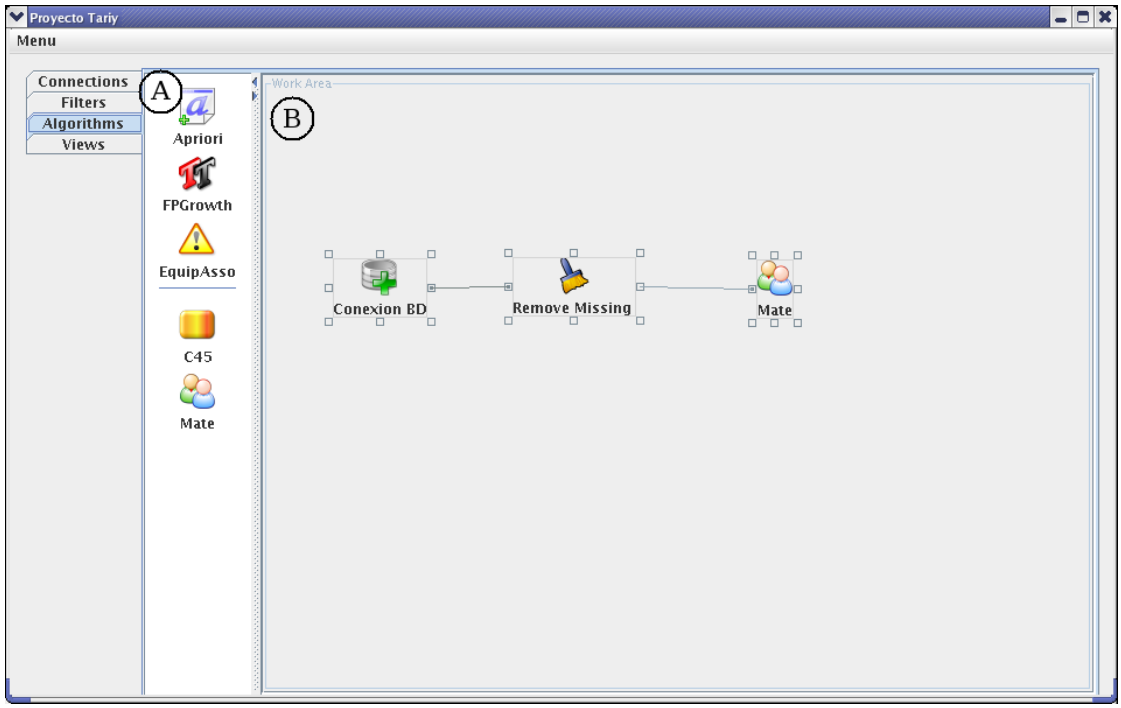

Figura 7.114: Algoritmo Mate

ACCIÓN DEL ACTOR	RESPUESTA DEL SISTEMA
1. Si el usuario quiere minar los datos con Mate y presiona sobre A (Área de opciones), en el icono respectivo.	2. En B (Área de trabajo) aparece el icono del algoritmo Mate.

Diagrama de Visualización

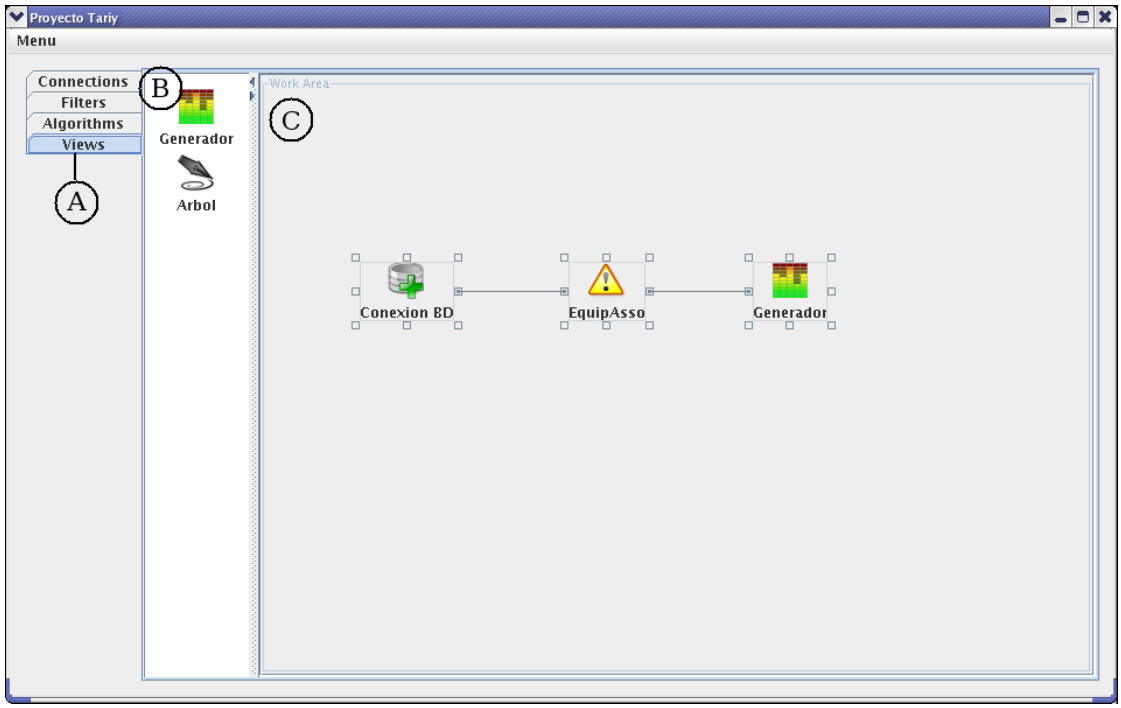

Figura 7.115: Diagrama de Visualización

ACCIÓN DEL ACTOR	RESPUESTA DEL SISTEMA
1. Cuando el usuario ha construido una secuencia de Minería de Datos, con cualquiera de los algoritmos, en A se encuentra en la sección de vistas y en B (Área de opciones) ha hecho click en el icono generador.	2. Entonces en C (Área de trabajo) aparece el icono del generador, a través del cual el usuario puede acceder a las opciones de este módulo.

Opción Delete

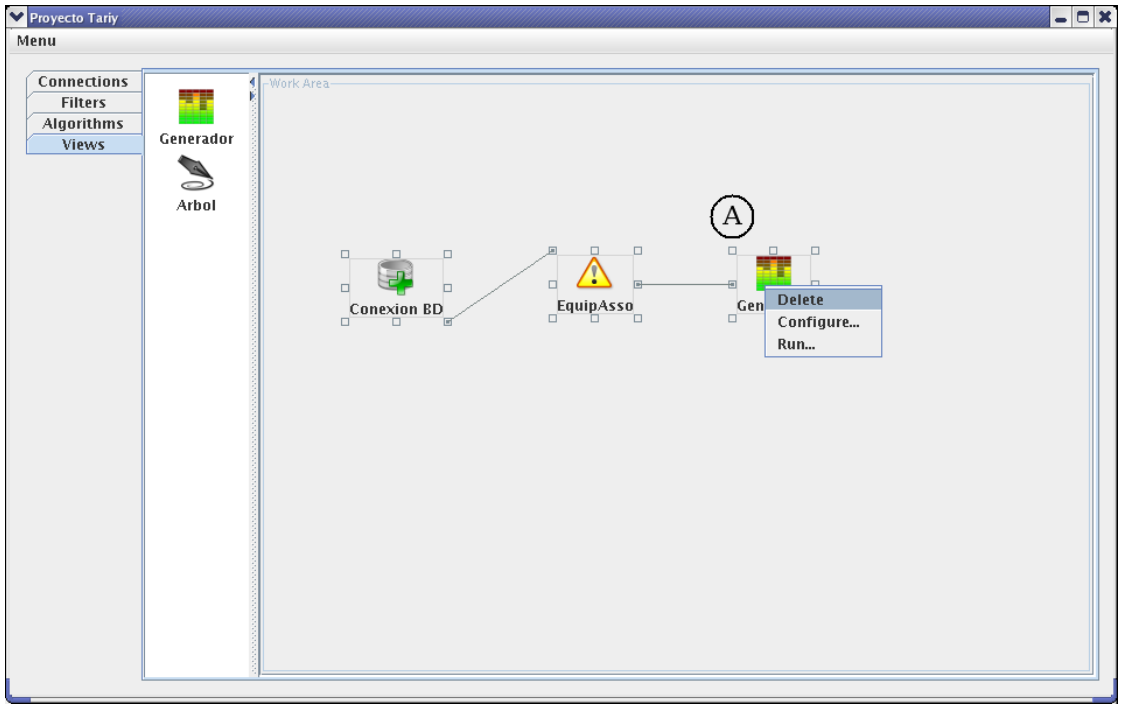

Figura 7.116: Opción Delete

ACCIÓN DEL ACTOR	RESPUESTA DEL SISTEMA
1. El usuario hace click derecho sobre A: el icono generador y elige la opción Delete.	2. El icono desaparece del área de trabajo, esperando un nuevo icono en la secuencia de Minería de Datos.

Opción Configure

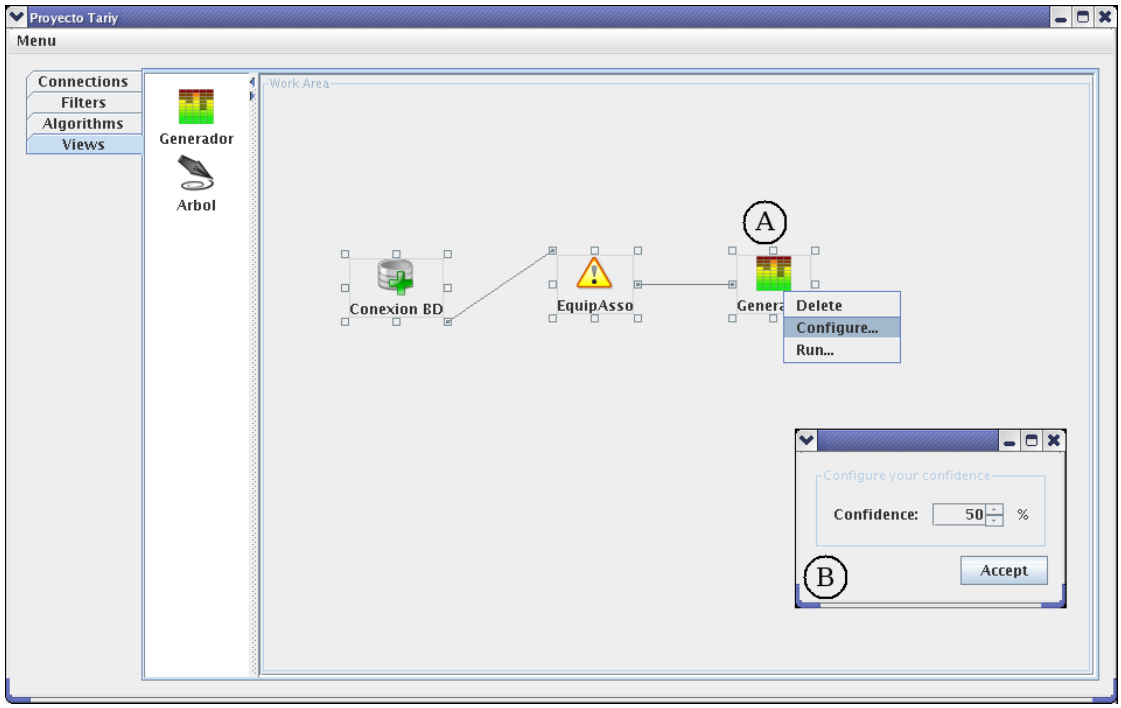

Figura 7.117: Opción Configure

ACCIÓN DEL ACTOR	RESPUESTA DEL SISTEMA
1. El usuario hace click sobre A: el icono del generador y elige configurar sus parametros.	2. Sobre el área de trabajo aparece una ventana B, para que el usuario configure la confianza con la cual se van a filtrar las reglas de asociación.

Opción Run

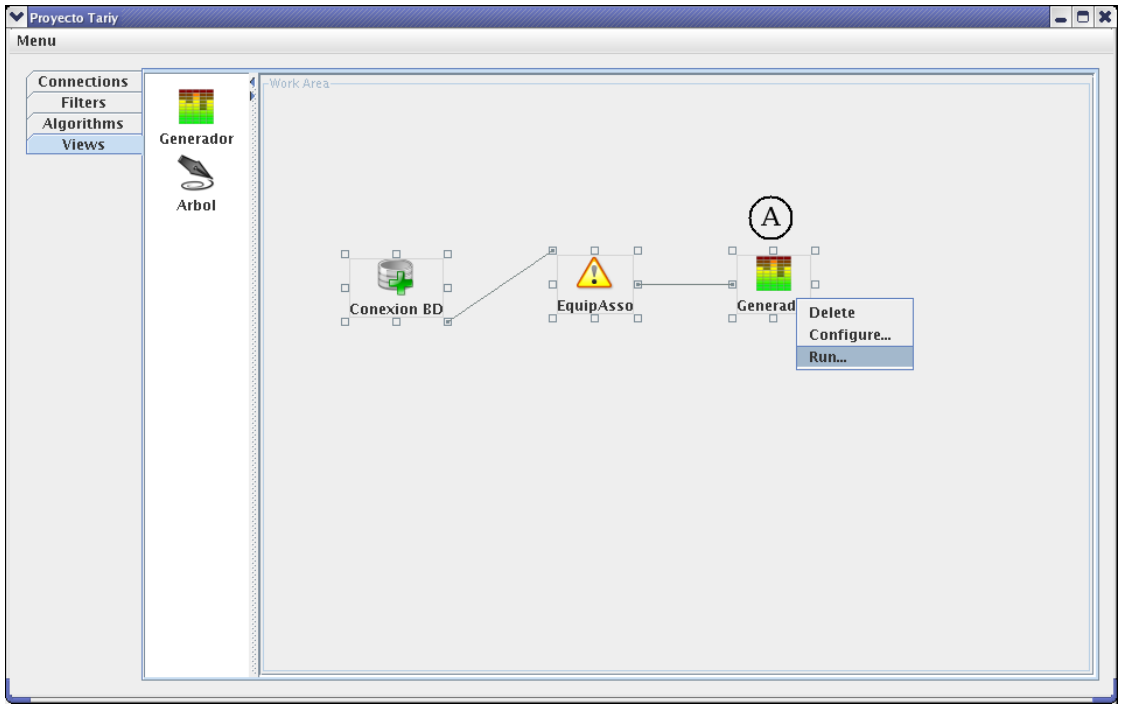

Figura 7.118: Opción Run

ACCIÓN DEL ACTOR	RESPUESTA DEL SISTEMA
1. El usuario hace click derecho sobre A: el icono generador y elige la opción Run.	2. En el área de trabajo aparece una ventana B, con las reglas obtenidas a partir de los algoritmos de Minería de Datos (La cual se explica en la siguiente figura).

Visor de Reglas

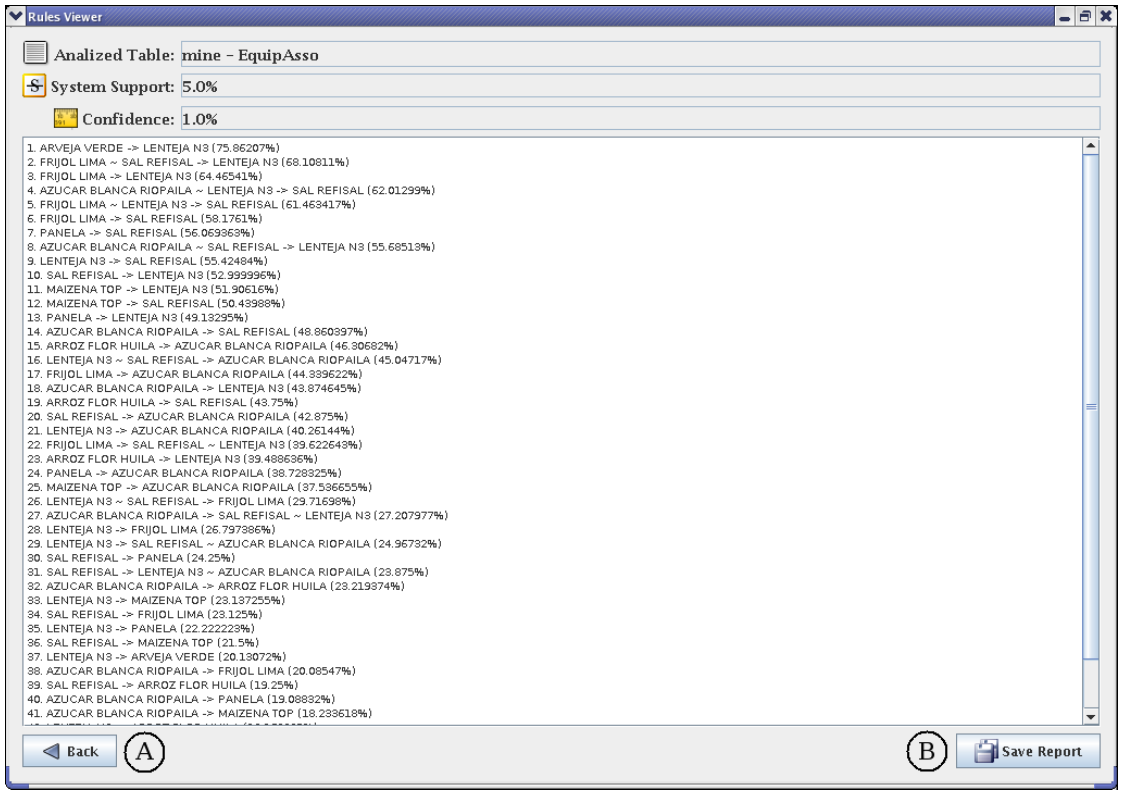

Figura 7.119: Visor de Reglas

ACCIÓN DEL ACTOR	RESPUESTA DEL SISTEMA
1. El usuario tiene la opción de hacer click en A, o en B.	2. Al hacer click en A, la ventana de reglas desaparece y si hace click en B el usuario tiene la opción de guardar el reporte de las reglas de asociación.

Capítulo 8

Conclusiones

En este proyecto se diseñó e implementó una herramienta débilmente acoplada con el SGBD PostgreSQL que da soporte a las etapas de conexión, preprocesamiento, minería y visualización del proceso KDD. Igualmente se incluyeron en el estudio nuevos algoritmos de asociación y clasificación propuestos por Timaran [51].

Para el desarrollo del proyecto se hizo un análisis de varias herramientas de software libre que abordan tareas similares a las que se pretendía en este trabajo. Se identificó las limitaciones y virtudes de estas aplicaciones y se diseñó una metodología para el desarrollo de una herramienta que cubriera las falencias encontradas.

Teniendo en cuenta la intención de liberar la herramienta se establecieron patrones de diseño que hicieran posible el acoplamiento de nuevas funcionalidades a cada uno de los módulos que lo componen, facilitando así la inclusión futura de nuevas características y el mejoramiento continuo de la aplicación.

La construcción de TariyKDD comprendió el desarrollo de cuatro módulos que cubrieron, el proceso de conexión a datos, tanto a archivos planos como a bases de datos relacionales, la etapa de preprocesamiento, donde se implementaron 9 filtros para la selección, transformación y preparación de los datos, el proceso de minería, que comprendió tareas de asociación y clasificación, implementando 5 algoritmos, Apriori, FPGrowth y EquipAsso para asociación y C4.5 y MateBy para clasificación y el proceso de visualización de resultados, utilizando tablas y árboles para generar reportes de los resultados y reglas obtenidas. Estos desarrollos fueron logrados usando en su totalidad herramientas de código abierto y software libre.

Se desarrolló un modelo de datos que facilitó la aplicación de algoritmos de asociación sobre bases de datos enmarcadas en el concepto de canasta de mercado

donde la longitud de cada transacción es variable.

Se realizaron pruebas para evaluar la validez de los algoritmos implementados. Para el plan de pruebas de la tarea de asociación, se utilizaron conjuntos de datos reales de transacciones de un supermercado de la Caja de Compensación familiar de Nariño. Para Clasificación, se trabajó con conjuntos de datos especializados para este tipo de algoritmos disponibles en [16].

Analizando las pruebas obtenidas para Asociación, en esta arquitectura débilmente acoplada con PostgreSQL, el rendimiento es muy significativo obteniendo muy buenos tiempos de respuesta al aplicar el algoritmo EquipAsso.

Fruto de este estudio se publicó y sustentó un artículo internacional en el marco del Congreso Latinoamericano de Estudios Informaticos - CLEI 2006 realizado en la ciudad de Santiago de Chile.

Se cuenta con una versión de TariyKDD con la capacidad de extraer reglas asociación y clasificación bajo una arquitectura débilmente acoplada con el SGBD PostgreSQL desarrollada bajo los lineamientos del software libre.

Una vez que se han descrito los resultados más relevantes que se han obtenido durante la realizacion de este proyecto, se sugiere una serie de recomendaciones como punto de partida para futuros trabajos:

1. Realizar mayores pruebas de rendimiento de esta arquitectura e implementar otras primitivas que Timaran propone para tareas de Asociación y Clasificación.

2. Implementar otras tares y algoritmos de minería de datos, asi como nuevos filtros e interfaces de visualización que permitan el mejoramiento continuo de TariyKDD.

3. Implementar nuevas interfaces gráficas que permitan la visualización de información de una manera más amigable para el usuario.

4. Probar el módulo de clasificación con bases de datos reales.

5. Implementar una funcionalidad que permita aplicar el modelo de clasificación construido y clasificar datos cuya clase se desconoce.

6. Liberar y compartir una versión de TariyKDD con la capacidad de descubrir conocimiento en bases de datos.

Finalmente este trabajo nos permitio aplicar los conocimientos adquiridos en el programa de Ingeniería de Sistemas y en especial los de la electiva de bases de datos, asi como nuestro trabajo y aprendizaje dentro del Grupo de Investigacion GRiAS Línea KDD.

Capítulo 9

ANEXOS

9.1. Pruebas y resultados

9.1.1. Rendimiento algoritmos de asociación

El conjunto de datos utilizados en las pruebas pertenecen a las transacciones de uno de los supermercados más importantes del departamento de Nariño (Colombia) durante un periodo determinado. El conjunto de datos contiene 10.757 diferentes productos. Los conjuntos de datos minados con la herramienta TariyKDD se muestran en el cuadro 9.1.

Para cada conjunto de datos se realizó preprocesamiento y transformación de datos con el fin de eliminar los productos repetidos en cada transacción y posteriormente se cargaron las tablas objeto de un modelo simple (i.e. una tabla con esquema Tid, Item) a la estructura de datos DataSet descrito en el capítulo 7 en la sección de implementación.

Cuadro 9.1: Conjuntos de Datos

Nomenclatura	**Numero de Registros**	**Numero de Transaccions**	**Promedio items por transaccion**
BD85KT7	555.123	85.692	7
BD40KT5	194.337	40.256	5
BD10KT10	97.824	10.731	10

Se evaluó el rendimiento de los algoritmos Apriori, FP-Growth y Equipasso, comparando los tiempos de respuesta para diferentes soportes mínimos. Los resultados de la evaluación del tiempo de ejecución de estos algoritmos, aplicados a los conjuntos de datos BD85KT7, BD40KT5 y BD10KT10, se pueden observar en las figuras 9.1, 9.2 y 9.3 respectivamente.

En general, observando el comportamiento de los algoritmos FP-Growth y Equipasso con los diferentes conjuntos de datos, se puede decir que su rendimiento es similar, contrario al tiempo de ejecución de Apriori, que se ve afectado significativamente a medida que se disminuye el soporte.

Cuadro 9.2: Tiempos de ejecución tabla BD85KT7

BD85KT7			
Soporte (%)	Tiempo (ms)		
	Apriori	FPGrowth	EquipAsso
4.15	750	166	85
4.75	362	162	82
5.35	365	164	83
5.95	365	162	83
6.55	120	159	80

Cuadro 9.3: Tiempos de ejecución tabla BD40KT5

BD40KT5			
Soporte (%)	Tiempo (ms)		
	Apriori	FPGrowth	EquipAsso
1.90	268	66	29
2.00	265	64	29
2.10	132	63	27
2.20	45	61	27
2.30	44	61	27

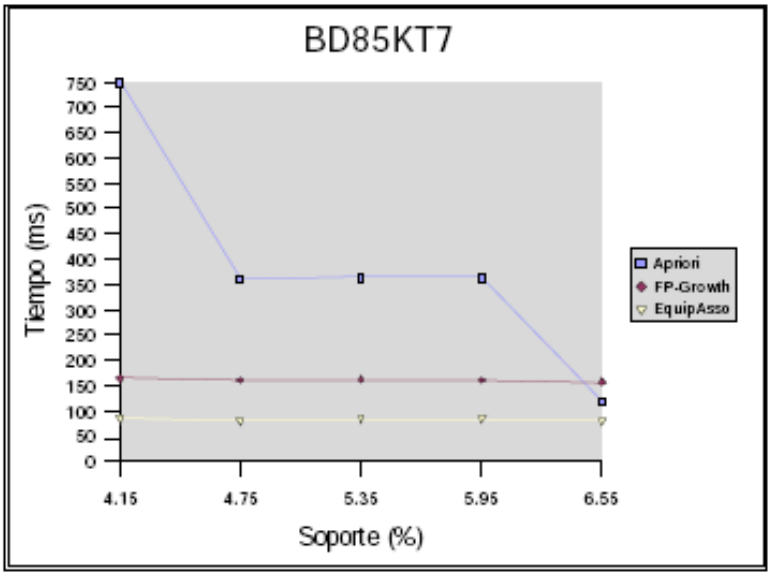

Figura 9.1: Rendimiento BD85KT7

Cuadro 9.4: Tiempos de ejecución tabla BD10KT10

BD10KT10			
Soporte (%)	Tiempo (ms)		
	Apriori	FPGrowth	EquipAsso
3.00	525	28	15
4.00	182	26	14
5.00	105	25	13
6.00	53	24	13
7.00	52	24	13

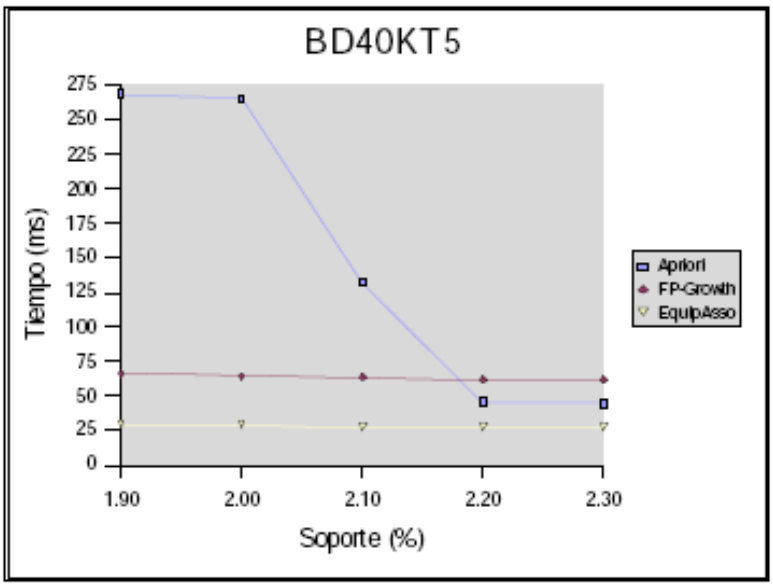

Figura 9.2: Rendimiento BD40KT5

Analizando el tiempo de ejecución de únicamente los algoritmos FP-Growth y EquipAsso (figuras 9.4, 9.5 y 9.6) para los conjuntos de datos BD85KT7, BD40KT5 y BD10KT10. Con soportes más bajos, el comportamiento de estos algoritmos sigue siendo similar.

Cuadro 9.5: Tiempos de ejecución tabla BD85KT7

BD85KT7			
Soporte (%)	Tiempo (ms)		
	Apriori	FPGrowth	EquipAsso
1.00	-	5130	1225
1.50	-	730	270
2.00	-	212	205
2.50	-	202	202
3.00	-	185	187

Cuadro 9.6: Tiempos de ejecución tabla BD40KT5

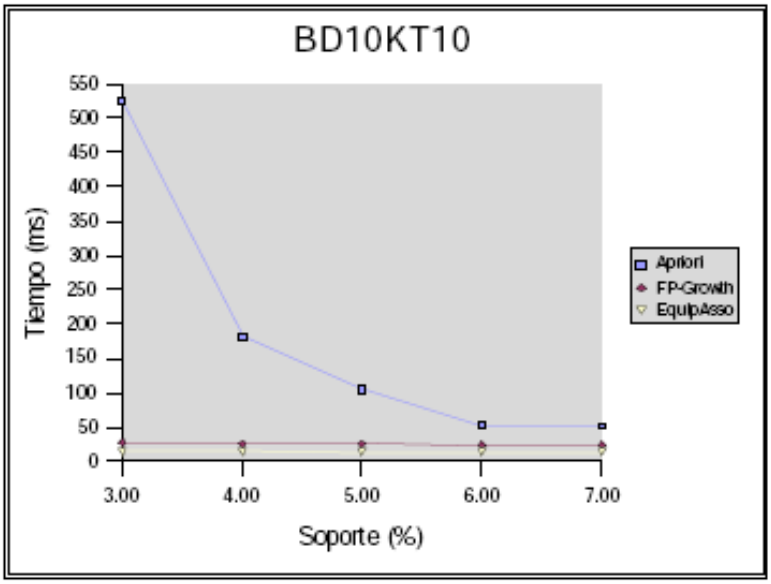

Figura 9.3: Rendimiento BD10KT10

BD40KT5			
Soporte (%)	Tiempo (ms)		
	Apriori	FPGrowth	EquipAsso
0.10	-	965	741
0.20	-	425	290
0.30	-	240	168
0.40	-	156	121
0.50	-	124	105

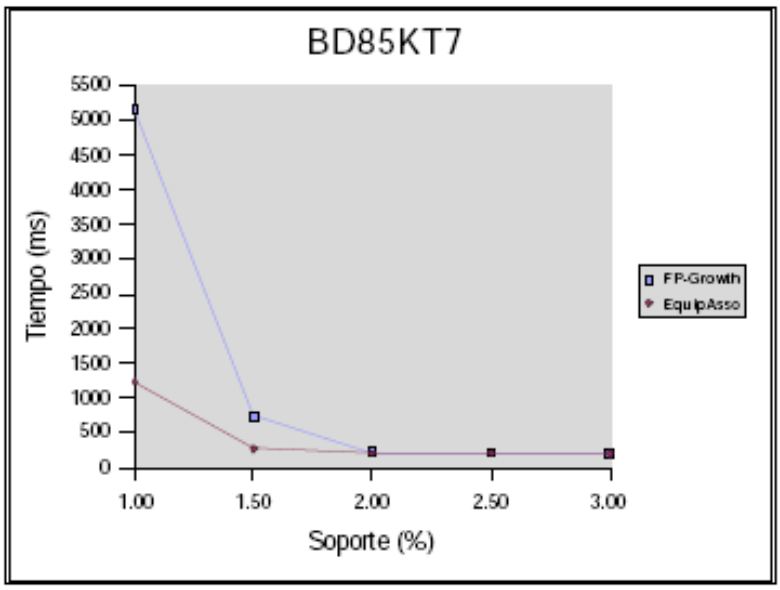

Figura 9.4: Rendimiento BD85KT7

Cuadro 9.7: Tiempos de ejecución tabla BD10KT10

BD10KT10			
Soporte (%)	Tiempo (ms)		
	Apriori	FPGrowth	EquipAsso
0.50	-	257	181
0.75	-	133	93
1.00	-	78	60
1.25	-	61	50
1.50	-	46	42

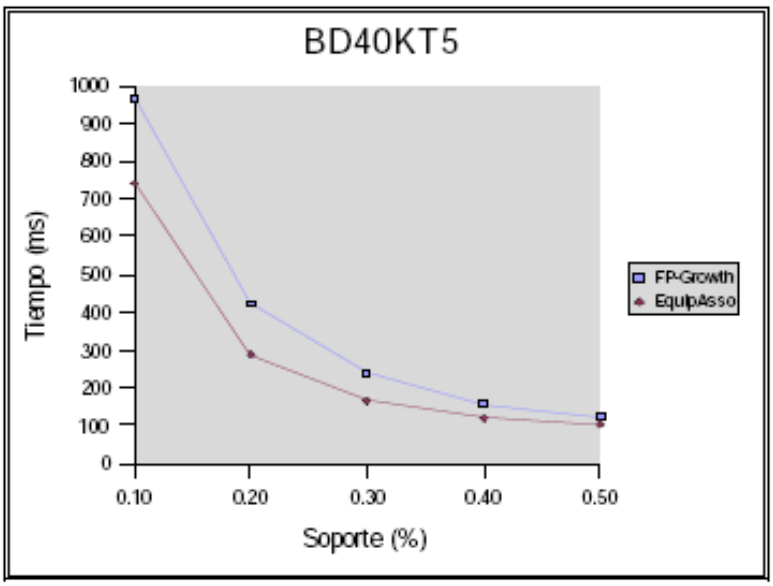

Figura 9.5: Rendimiento BD40KT5

9.1.2. Rendimiento formato de comprensión Tariy - Formato ARFF

Un análisis del formato para compresión de datos descrito en el Capítulo 7 del presente trabajo, en la sección Arquitectura Tariy, con respecto al formato ARFF de la herramienta de Minería de Datos WEKA se muestra en el siguiente cuadro, donde se registra el tamaño en disco de cada formato al almacenar conjuntos de datos con diferente número de transacciones y atributos.

Cuadro 9.8: Análisis formatos de almacenamiento

Archivo ARFF	**Núm. Instancias**	**Núm. Atributos**	**Tam. ARFF (KB)**	**Tam. Tariy (KB)**
mushroom	8124	23	726.30	133.81
titanic	2201	4	64.60	0.17
tictactoe	958	10	26.50	14.00
soybean	683	36	194.10	55.46
vote	435	17	32.20	8.87
contact-lenses	24	5	1.10	0.23
weather.nominal	14	5	0.57	0.16

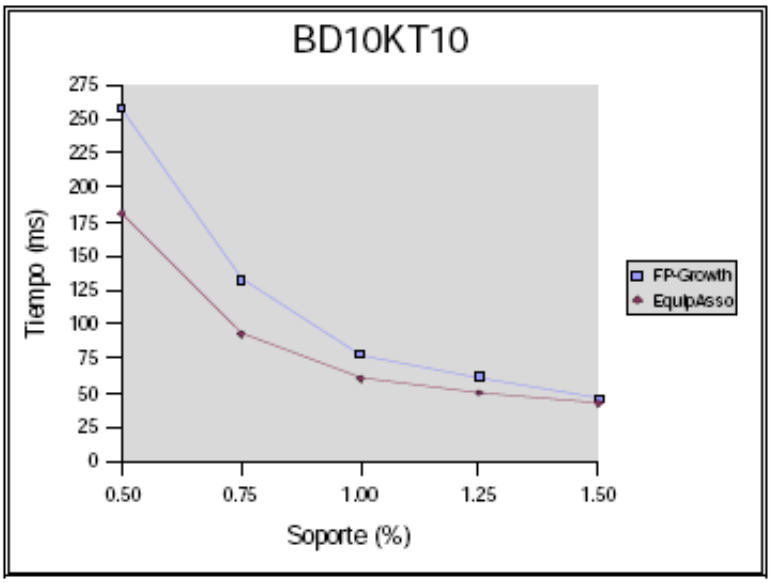

Figura 9.6: Rendimiento BD10KT10

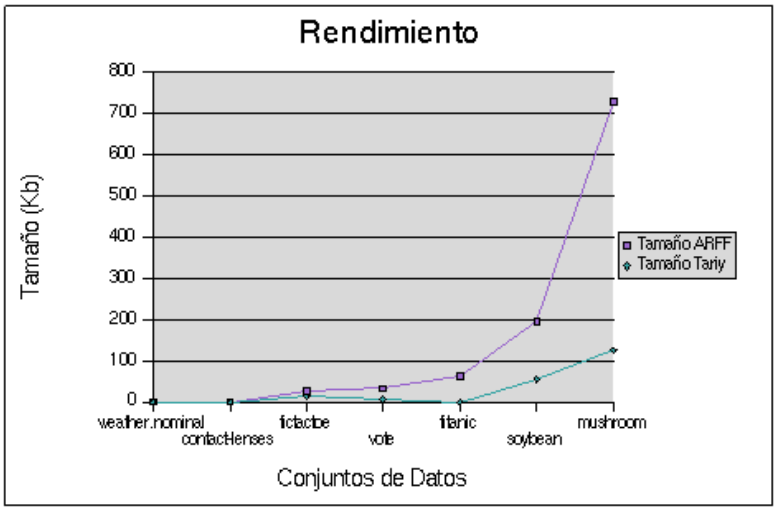

Figura 9.7: Rendimiento formatos de almacenamiento

Bibliografía

[1] Université Lumière Lyon 2. Eric equipe de recherche en ingénierie des conaissances. http://chirouble.univ-lyon2.fr, 2006.

[2] NASA Ñational Aeronautics and Space Administration. Tropical cyclone windspeed indicator. http://pm-esip.nsstc.nasa.gov/cyclone/.

[3] R. Agrawal, T. Imielinski, and A. Swami. *Mining Association Rules between Sets of Items in Large Databases*. ACM SIGMOD, 1993.

[4] R. Agrawal, M. Mehta, J. Shafer, R. Srikant, A. Arning, and T. Bollinger. The quest data mining system. In *2Âº Conference KDD y Data Mining*, Portland, Oregon, 1996.

[5] R. Agrawal and K. Shim. Developing tightly-coupled data mining applications on a relational database system. In *The Second International Conference on Knowledge Discovery and Data Mining*, Portland, Oregon, 1996.

[6] R. Agrawal and R. Srikant. Fast algorithms for mining association rules. In *VLDB Conference*, Santiago, Chile, 1994.

[7] R. Brachman and T. Anand. *The Process of Knowledge Discovery in Databases: A First Sketch, Workshop on Knowledge Discovery in Databases*. 1994.

[8] S. Chaudhuri. Data mining and database systems: Where is the intersection? In *Bulletin of the Technical Committee on Data Engineering*, volume 21, Marzo 1998.

[9] M. Chen, J. Han, and P. Yu. Data mining: An overview from database perspective. In *IEEE Transactions on Knowledgeand Data Engineering*, 1996.

[10] Quadrillion Corp. Q-yield. http://www.quadrillion.com/qyield.shtm, 2001.

[11] IBM Corporation. Intelligent miner. http://www-4.ibm.com/software/data/iminer, 2001.

[12] J Demsar and B Zupan. Orange: From experimental machine learning to interactive data mining. Technical report, Faculty of Computer and Information Science, University of Liubliana, Slovenia, http://www.ailab.si/orange/wp/orange.pdf, 2004.

[13] U. Fayyad, G. Piatetsky-Shapiro, and P. Smyth. From data mining to knowledge discovery: An overview, in advances in knowledge discovery and data mining. In *AAAI Pres / The MIT Press*, 1996.

[14] M. Goebel and L. Gruenwald. A survey of data mining and knowledge discovery software tools. In *SIGKDD Explorations*, volume 1 of *1*, June 1999.

[15] Waikato ML Group. Attribute-relation file format (arff). http://www.cs.waikato.ac.nz/ ml/weka/arff.html.

[16] Waikato ML Group. Collections of datasets. *http* : *//www.cs.waikato.ac.nz/ml/weka/index*$_d$*atasets.html*.

[17] Waikato ML Group. The waikato environment for knowledge analisys. http://www.cs.waikato.ac.nz/ml/weka.

[18] J. Han, J. Chiang, S. Chee, J. Chen, Q. Chen, S. Cheng, W. Gong, M. Kamber, K. Koperski, G. Liu, Y. Lu, N. Stefanovic, L. Winstone, B. Xia, O. Zaiane, S. Zhang, and H. Zhu. Dbminer: A system for data mining in relational databases and data warehouses. In *CASCON: Meeting of Minds*.

[19] J. Han, Y. Fu, and S. Tang. Advances of the dblearn system for knowledge discovery in large databases. In *International Joint Conference on Artificial Intelligence IJCAI*, Montreal, Canada, 1995.

[20] J. Han, Y. Fu, W. Wang, J. Chiang, K. Koperski, D. Li, Y. Lu, A. Rajan, N. Stefanovic, B. Xia, and O. Zaiane. Dbminer: A system for mining knowledge in large relational databases. In *The second International Conference on Knowledge Discovery & Data Mining*.

[21] J. Han, Y. Fu, W. Wang, J. Chiang, O. Zaiane, and K. Koperski. *DBMiner: Interactive Mining of Multiple-Level Knowledge in Relational Databases*. ACM SIGMOD, Montreal, Canada, 1996.

[22] J. Han and M. Kamber. *Data Mining Concepts and Techniques*. Morgan Kaufmann Publishers, 2001.

[23] J. Han and J. Pei. Mining frequent patterns by pattern-growth: Methodology and implications. In *SIGKDD Explorations*, volume 2:14-20, 2000.

[24] J. Han, J. Pei, and Y. Yin. Mining frequent patterns without candidate generation. In *ACM SIGMOD*, Dallas, TX, 2000.

[25] Java Hispano. El abc de jdbc. http://javahispano.org/tutorials.type.action?type=j2se, 2004.

[26] T. Imielnski and H. Mannila. A database perspective on knowledge discovery. In *Communications of the ACM*.

[27] RuleQuest Research Inc. C5.0. http://www.rulequest.com, 2001.

[28] E-Business Technology Institute. E-business technology institute, the university of hong kong. http://www.eti.hku.hk, 2005.

[29] Quinlan J.R. *C4.5: Programs for Machine Leraning*. Morgan Kaufmann Publishers, 1993.

[30] Kdnuggets. http://www.kdnuggets.com/software, 2001.

[31] C. Matheus, P. Chang, and G. Piatetsky-Shapiro. Systems for knowledge discovery in databases. In *IEEE Transactions on Knowledge and Data Engineering*, volume 5, 1993.

[32] I Mierswa, M Wurst, R Klinkenberg, M Scholz, and T Euler. Yale: Rapid prototyping for complex data mining tasks. 12th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD-06), 2006.

[33] Faculty of Computer and Slovenia Information Science, University of Liubliana. Orange, fruitful and fun. http://www.ailab.si/orange, 2006.

[34] Faculty of Computer and Slovenia Information Science, University of Liubliana. Orange's interface to mysql. http://www.ailab.si/orange/doc/modules/orngMySQL.htm, 2006.

[35] Goverment of Hong Kong. Innovation and technology fund. http://www.itf.gov.hk, 2006.

[36] Artificial Inteligence Unit of the University of Dortmund. Artificial inteligence unit of the university of dortmund. http://www-ai.cs.uni-dortmund.de, 2006.

[37] G. Piatetsky-Shapiro, R. Brachman, and T. Khabaza. *An Overview of Issues in Developing Industrial Data Mining and Knowledge Discovery Applications*. 1996.

[38] J.R. Quinlan. *Induction of decision trees. Machine Learning.* 1986.

[39] R Rakotomalala. Tanagra. In *TANAGRA: a free software for research and academic purposes*, volume 2, pages 697–702. EGC'2005, 2005.

[40] R Rakotomalala. Tanagra project. http://chirouble.univ-lyon2.fr/ ricco/tanagra/en/tanagra.html, 2006.

[41] Isoft S.A. Alice. http://www.alice-soft.com/html/prod_alice.htm, 2001.

[42] S. Sarawagi, S. Thomas, and R. Agrawal. Integrating association rule mining with relational database systems: Alternatives and implications. In *ACM SIGMOD*, 1998.

[43] SPSS. Clementine. http://www.spss.com/clementine, 2001.

[44] Information Technology The University of Alabama in Huntsville and Systems Center. Adam 4.0.2 components. http://datamining.itsc.uah.edu/adam/documentation.html.

[45] Information Technology The University of Alabama in Huntsville and Systems Center. Algorithm development and mining system. http://datamining.itsc.uah.edu/adam/index.html.

[46] R. Timarán. Arquitecturas de integración del proceso de descubrimiento de conocimiento con sistemas de gestión de bases de datos: un estado del arte, en revista ingeniería y competitividad. *Revista de Ingeniería y Competitividad, Universidad del Valle*, 3(2), Diciembre 2001.

[47] R. Timarán. Descubrimiento de conocimiento en bases de datos: Una visión general. In *Primer Congreso Nacional de Investigación y Tecnología en Ingeniería de Sistemas*, Universidad del Quindío, Armenia, Octubre 2002.

[48] R. Timarán. *Nuevas Primitivas SQL para el Descubrimiento de Conocimiento en Arquitecturas Fuertemente Acopladas con un Sistema Gestor de Bases de Datos.* PhD thesis, Universidad del Valle, 2005.

[49] R. Timarán and M. Millán. Equipasso: an algorithm based on new relational algebraic operators for association rules discovery. In *Fourth IASTED International Conference on Computational Intelligence*, Calgary, Alberta, Canada, July 2005.

[50] R. Timarán and M. Millán. Equipasso: un algoritmo para el descubrimiento de reglas de asociación basado en operadores algebraicos. In *4Âª Conferencia Iberoamericana en Sistemas, Cibernética e Informática CICI 2005*, Orlando, Florida, EE.UU., Julio 2005.

[51] R. Timarán, M. Millán, and F. Machuca. New algebraic operators and sql primitives for mining association rules. In *IASTED International Conference Neural Networks and Computational Intelligence*, Cancun, Mexico, 2003.

[52] I. Waitten and F. Eibe. *Data Mining: Practical Machine Learning Tools and Techniques with Java Implementations*. 2001.

[53] YALE. Yale - yet another learning environment. http://rapid-i.com, 2006.