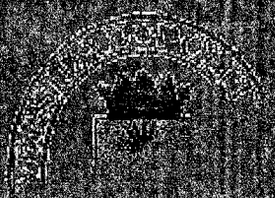

VII Jornadas de Ingeniería del Software y Bases de Datos (JISBD'02)

Editores:
Matilde Celma
Óscar Pastor
Natalia Juristo
Juan José Moreno

Grupo de Ingeniería del Software.
Facultad de Informática.
Universidad Politécnica de Madrid.

El Escorial (Madrid),
19-21 de Noviembre de 2002

VII Jornadas de Ingeniería del Software y Bases de Datos (JISBD'02)

Editores:
Matilde Celma
Óscar Pastor
Natalia Juristo
Juan José Moreno

Grupo de Ingeniería del Software.
Facultad de Informática.
Universidad Politécnica de Madrid.

El Escorial (Madrid),
19-21 de Noviembre de 2002

ACTAS DE LAS VII JORNADAS DE
INGENIERÍA DEL SOFTWARE Y BASES DE DATOS (JISBD'02)

EDITORES:
Matilde Celma
Óscar Pastor
Natalia Juristo
Juan José Moreno

Primera edición, 2002
I.S.B.N.:84-688-0206-9

COMITÉ DE PROGRAMA

PRESIDENTES

Matilde Celma (Universidad Politécnica de Valencia)
Óscar Pastor (Universidad Politécnica de Valencia)

MIEMBROS

Idoia Alarcón (Universidad Autónoma de Madrid)
José Francisco Aldana (Universidad de Málaga)
Manuel Barrena (Universidad de Extremadura)
Pere Botella (Universitat Politecnica de Catalunya)
Nieves Brisaboa (Universidad de La Coruña)
Coral Calero (Universidad de Castilla - La Mancha)
José Hilario Canós (Universidad Politécnica de Valencia)
Juan Carlos Casamayor (Universidad Politécnica de Valencia)
Matilde Celma (Universidad Politécnica de Valencia)
Rafael Corchuelo (Universidad de Sevilla)
Dolors Costal (Universitat Politecnica de Catalunya)
Carmen Costilla (Universidad Politécnica de Madrid)
José Manuel Cueva (Universidad de Vigo)
Carlos Delgado (Universidad Carlos III)
Óscar Díaz (Universidad del País Vasco)
Amador Durán (Universidad de Sevilla)
Xavier Franch (Universitat Politecnica de Catalunya)
Pablo de la Fuente (Universidad de Valladolid)
Juan Garbajosa (Universidad Politécnica de Madrid)
Jesús García (Universidad de Murcia)
Jaime Gómez (Universidad de Alicante)
Alfredo Goñi (Universidad del País Vasco)
Juan Hernández (Universidad de Extremadura)
Ángel Herranz (Universidad Politécnica de Madrid)
Arantxa Illaramendi (Universidad del País Vasco)
Arturo Jaime (Universidad del País Vasco)
Natalia Juristo (Universidad Politécnica de Madrid)
Esperanza Marcos (Universidad Rey Juan Carlos de Madrid)
José Manuel Marqués (Universidad de Valladolid)
Eduardo Mena (Universidad de Zaragoza)
Roberto Morrión (Universidad Autónoma de Madrid)

José Parets (Universidad de Granada)
Óscar Pastor (Universidad Politécnica de Valencia)
Mario Piattini (Universidad de Castilla La Mancha)
Ernesto Pimentel (Universidad de Málaga)
Macario Polo (Universidad de Castilla La Mancha)
Isidro Ramos (Universidad Politécnica de Valencia)
Pedro Sánchez (Universidad Politécnica de Cartagena)
Ernest Teniente (Universitat Politecnica de Catalunya)
Ambrosio Toval (Universidad de Murcia)
José María Troya (Universidad de Málaga)
Felix Saltor (Universitat Politecnica de Catalunya)
José Samos (Universidad de Granada)
María Sancho (Universidad Politécnica de Cataluña)
Miguel Toro (Universidad de Sevilla)
Toni Urpi (Universitat Politecnica de Catalunya)
Antonio Vallecillo (Universidad de Málaga)
Juan Carlos Yelmo (Universidad Politécnica de Madrid)

COLABORADORES EN EL PROCESO DE REVISIÓN

Albert Abelló
Bárbara Álvarez Torres
Jesús Arias Fisteus
J. M. Cavero
Antonio Cesar Gómez
Carlos E. Cuesta
M. Díaz
Oscar Dieste
Ricard Gavaldà
Jose Hernandez-Orallo
Emilio Insfran
Adolfo Lozano
José Antonio Lozano
Esperanza Manso
Noelia Maya
Nelson Medinilla
Ernestina Menasalvas
Pedro Merino
Juan Manuel Murillo
Joaquín Nicolás
Carme Quer
Juan Angel Pastor
Vicente Pelechano
Josep Maria Pujol
Mª Ribera Sancho
Fernando Sánchez
Mª Elena Rodríguez
Antonio Ruiz
Maria-Isabel Sánchez-Segura
Jesús Villamor
Marienma I. Yagüe

COMITÉ ORGANIZADOR

PRESIDENTA

Natalia Juristo (Universidad Politécnica de Madrid)

MIEMBROS

Óscar Dieste (Universidad Politécnica de Madrid)
Ángel Herranz (Universidad Politécnica de Madrid)
Julio Mariño (Universidad Politécnica de Madrid)
Sira Vegas (Universidad Politécnica de Madrid)

Prólogo del Comité de Programa

El presente volumen recoge las actas de las VII[a] Jornadas de Ingeniería de Software y Bases de Datos, celebradas en El Escorial (Madrid), los días 19, 20 y 21 de Noviembre de 2002. Después de una larga andadura que se inició en 1994, y fruto del esfuerzo y empuje de muchos compañeros, las comunidades de Ingeniería de Software y Bases de Datos constituyen actualmente un único colectivo, unido por el objetivo común de impulsar la investigación, el desarrollo y la transferencia tecnológica en sus respectivas campos de interés.

La experiencia demuestra que la línea divisoria entre ambos campos científicos es difusa, y que los mismos problemas son en muchas ocasiones objeto de atención por investigadores de ambas comunidades; por ello la acción decidida de los dos colectivos en aproximarse hasta converger en unas únicas Jornadas nacionales, significó en su momento un paso hacia delante que ha fortalecido a ambas comunidades. Con la celebración de las Jornadas de Ingeniería del Software y Bases de Datos, ambas comunidades disponen de un foro de encuentro en el que discutir y poner en común sus puntos de vista, fomentándose de esta forma, la cooperación y la creación de equipos multidisciplinares, imprescindibles hoy en día para abordar los retos que plantea la nueva sociedad de la información.

Una parte importante del éxito de estos encuentros reside en la calidad y actualidad de los trabajos que se presentan a discusión entre los participantes. Conscientes de este hecho y herederos de la experiencia y el buen hacer de los presidentes de los comités de programa que nos han precedido, hemos intentado realizar el proceso de discusión y selección de los trabajos presentados con el mayor rigor posible. La bondad de la labor realizada es mérito exclusivo de los miembros del Comité de Programa que realizan su trabajo de forma anónima y desinteresada, desde aquí queremos agradecerles a todos ellos el espíritu de colaboración y trabajo que han demostrado. A continuación se presenta un resumen de todo el proceso.

A esta séptima edición de las Jornadas se presentaron 70 trabajos. Los trabajos fueron revisados por tres evaluadores. El resultado global de una evaluación se traducía en un valor numérico en el rango de 1 a 5, con el siguiente significado: 1 (rechazado), 2 (débilmente rechazado), 3 (indiferente), 4 (débilmente aceptado) y 5 (aceptado). Fueron aceptados inicialmente todos los trabajos con una nota superior a tres que no tuviesen ninguna evaluación suspendida (menor o igual a dos), y rechazados todos los trabajos con una nota menor o igual a 3 y con al menos dos notas suspendidas. Los trabajos que no caían en ninguno de los grupos anteriores se encontraban en la franja alrededor del tres y presentaban disparidad de evaluaciones, por lo que fueron objeto de discusión por parte de sus evaluadores para decidir el rechazo o aceptación definitivo. El resultado de la aplicación de estos criterios fue: 35 trabajos aceptados, 29 rechazados y 6 propuestos para ser presentados como posters.

En las últimas ediciones, las Jornadas han ido atrayendo la celebración de otros eventos asociados, unidos por el eje común de la tecnología del software. Ello ha conducido a definir en la presente edición un evento marco, Encuentros en Tecnologías de Software, en el que se coordinan todos ellos, iniciativa interesante que sigue incidien-

do en la idea de reunir esfuerzos para fomentar la cooperación interdisciplinar y el intercambio de información científica. En este contexto, los trabajos que han sido aceptados serán presentados en las sesiones técnicas de la Jornadas de Ingeniería del Software y Bases de Datos, sesiones que serán completadas con la celebración de talleres, tutoriales y conferencias invitadas que se han organizado en el marco de los citados Encuentros.

Por último agradecer a toda la comunidad científica su participación en estas Jornadas enviando sus trabajos, y acudiendo al evento. Gracias también al Comité Organizador por su continuo apoyo durante todo el proceso.

Sólo resta por decir que el próximo año tenemos una cita para seguir avanzando en nuestra tarea de impulsar desde distintos frentes la tecnología del software y de prestar a nuestro entorno social el apoyo tecnológico que necesita.

Valencia, Noviembre de 2002

Oscar Pastor & Matilde Celma
Presidentes del Comité de Programa

Prólogo del Comité Organizador

Los Encuentros en Tecnologías Software, cuya primera edición formal se celebra en este año 2002, representan un hito en el panorama de las Tecnologías de la Información en España.

Por una parte, los Encuentros en Tecnologías Software son la reunión más importante de investigadores, docentes y profesionales, en el ámbito del Software, que se celebra en nuestro país, convocando este año a más de 250 participantes. Durante los cinco días que comprenden los Encuentros, los asistentes discutirán acerca de una diversidad de temas, organizados en Jornadas temáticas, que incluyen Ingeniería del Software, Bases de Datos, Programación, Bibliotecas Digitales, Sistemas de Información Geográficos y un variado etcétera de temas relacionados con las Tecnologías de la Información y sus aplicaciones organizados en workshops, talleres, y otros foros de discusión.

Así pues, los Encuentros en Tecnologías Software engloban las Séptimas Jornadas de Ingeniería del Software y Bases de Datos (JISBD), las Terceras Jornadas de Bibliotecas Digitales (JBIDI), las Segundas Jornadas de Sistemas de Información Geográficos (JSIG) y las Jornadas de Lenguajes de Programación (PROLE). Los miembros de los distintos comités de programa, el número de artículos recibidos y la rigurosidad de los criterios de selección son el mejor testimonio de la calidad de los Encuentros.

El esfuerzo que supone la organización de un evento de este tipo es considerable, y los sinsabores frecuentes. No obstante, dicho esfuerzo ha quedado recompensadopor los resultados obtenidos: una espléndida acogida, una alta participación y la seguridad de que estos Encuentros promocionarán el avance de las Tecnologías de la Información en España.

Todo ello no habría sido posible sin la colaboración desinteresada de una gran cantidad de personas. Gracias a los miembros de los distintos Comités de Programa y a sus Presidentes. Es necesario destacar el tremendo esfuerzo de los miembros del Comité Organizador, reservando una mención especial para Oscar Dieste y Sira Vegas. Por último, debe hacerse extensivo este agradecimiento al Ministerio de Ciencia y Tecnología, cuya colaboración económica ha sido fundamental para la realización de estos Encuentros.

Natalia Juristo Juzgado
Presidenta del Comité Organizador

CONTENIDOS

Una comparativa de métodos para la estimación del esfuerzo en programas 4GL: experiencias con Estadística y Minería de Datos

Macario Polo[1], José C. Riquelme[2], Mario Piattini[1], Jesús S. Aguilar-Ruiz[2], Francisco Ruiz[1], Francisco Ferrer[2]

[1]Escuela Superior de Informática
Universidad de Castilla-La Mancha
Ronda de Calatrava, 5
13071-Ciudad Real
macario.polo@ uclm.es

[2]Escuela Técnica Superior de Informática
Universidad de Sevilla
Avenida de Reina Mercedes, s/n
41012-Sevilla
riquelme@lsi.us.es

Reumen. Este trabajo presenta un estudio empírico para analizar las relaciones entre un conjunto de métricas para programas realizados en lenguajes de 4ª generación (4GL) y su mantenimiento. El análisis ha sido realizado usando los datos históricos de diversos proyectos industriales y tres diferentes aproximaciones: la primera relaciona métricas y mantenimiento mediante técnicas de estadística descriptiva, y dos métodos basados en técnicas de Minería de Datos. Se presenta una discusión sobre las tres técnicas, además de un conjunto de ecuaciones y reglas para predecir el esfuerzo de mantenimiento en esta clase de programas.

1. Introducción

La mayoría de las organizaciones tienen sistemas de información que utilizan algún tipo de base de datos para almacenar y reutilizar los datos propios. Aunque en los últimos años han aparecido nuevos modelos y paradigmas para bases de datos (orientadas a objeto, objeto-relacional, etc.) recientes estudios señalan que el modelo relacional es aún el más comúnmente usado [9]. Para permitir a los usuarios la manipulación de los datos existentes en la base de datos, los programadores proporcionan programas basados en instrucciones SQL para acceder a la base de datos. Estos programas pueden haber sido desarrollados en un lenguaje de 3ª generación (3GL) (por ejemplo, Cobol o Visual Basic), directamente en un 4GL, o haber sido migrado desde un 3GL a un 4GL, que son entornos más productivos 8.

Los 4GL son lenguajes no procedurales cuyos programas especifican qué debe ser hecho por el programa sin detalles de "cómo hacerlo". Estos lenguajes aparecieron como un mecanismo poderoso para explotar fácilmente las características de los Sistemas de Gestión de Bases de Datos (DMS en inglés). Ejemplos de 4GL son Oracle Forms, Powerbuilder or CA-OpenIngres. En este trabajo se analiza la relación de un conjunto de métricas para 4GL con el tiempo de mantenimiento de estos programas. Las métricas utilizadas han sido previamente validadas formal y empíricamente:

- La validación formal puede encontrarse en 10, y se realizó teniendo en cuenta los marcos formales propuestos por Briand et al. 4 y Zuse 19.
- Un estudio empírico para validar las métricas fue presentado en 11. Este estudio usaba métodos clásicos de Estadística Descriptiva para encontrar correlaciones lineales entre los valores de las métricas y el tiempo dedicado a mantenimiento.

En este último estudio se encontraron significativas correlaciones entre ambas variables. Además el estudio empírico proporcionó un conjunto de útiles ecuaciones lineales para predecir el tiempo de mantenimiento de programas 4GL. En este trabajo, nos centramos en un re-análisis de los resultados obtenidos usando dos técnicas de Minería de Datos: M5', desarrollada por Quinlan en 1992 14, y HIDER, una herramienta propia basada en un algoritmo evolutivo 16. Los resultados de estas técnicas son reglas de decisión que, además de servir cómo métodos de predicción, también ofrecen un conocimiento adicional sobre los datos.

El resto del trabajo esta organizado de la siguiente manera: La Sección 2 describe las métricas propuestas y almacenadas junto con una breve discusión de las características especiales de los programas en 4GL. En la Sección 3 presentamos una tabla con los datos históricos recolectados y se explican algunas características del método de recolección llevado a cabo. Un resumen del análisis de estos datos mediante estadística descriptiva se presenta en la Sección 4, así como las ecuaciones lineales obtenidas; la Sección 5 está dedicada al análisis mediante Minería de Datos y la presentación de las reglas de decisión.. La Sección 6 es una discusión de los resultados obtenidos con ambos métodos. Finalmente en la Sección 7 están las conclusiones.

2. Breve descripción de las métricas propuestas

La definición de métricas para programas en 4GL presenta el problema de la gran heterogeneidad de posibles tipos de sentencias que pueden ser compuestas en este tipo de lenguajes. De esta manera, al menos siete sublenguajes pueden ser identificados: Control procedural, Control visual, Manejo de excepciones, Definición de la Base de Datos, Manipulación de la Base de Datos, Control de Seguridad y Control de Transacciones.

Las métricas utilizadas en este trabajo están diseñadas para el sublenguaje de manipulación de bases de datos. Otros autores han propuesto métricas para medir diferentes atributos de programas 4GL, aunque la mayoría de ellos se han centrado en la estimación del esfuerzo de desarrollo y en la correlación de éste con el tamaño de los programas 2717.

El sublenguaje de manipulación de la base de datos está compuesto principalmente por instrucciones SQL. Las métricas utilizadas tienen en cuenta el número total de instrucciones Select, Insert, Delete y Update, el número de anidamientos, así como el número total de tablas y de cláusulas Where en el programa. Así, las métricas definidas y formalmente validadas (ver [10]) son:

- NS = número total de instrucciones Select en el programa considerado
- NI = número total de instrucciones Insert en el programa considerado
- ND= número total de instrucciones Delete en el programa considerado
- NU= número total de instrucciones Update en el programa considerado
- NT = número total de Tablas usadas en el programa considerado
- NA = número total de Anidamientos en el programa considerado
- Where = número total de cláusulas Where en el programa considerado
- La variable dependiente es el tiempo de mantenimiento (TIME) del programa. En este trabajo consideramos que el "tiempo de mantenimiento" es el tiempo dedicado a las tareas de mantenimiento desde el momento en que la aplicación fue instalada hasta el final de su segunda versión.

3. Recogida de datos históricos

El sistema del que hemos extraído los datos utilizados en este trabajo es una aplicación desarrollada y mantenida por el Centro Provincial de Informática (CENPRI), perteneciente a la Diputación Provincial de Ciudad Real. El CENPRI desarrolla y mantiene software para su propia gestión, para 100 municipios de la provincia y para su relación institucional con los ciudadanos. Esto incluye gestión de impuestos, gestión del catastro, infraestructuras públicas, permisos de obras, salarios y personal, etc. También incluye interfaces con algunas aplicaciones de bancos, del Ministerio de Hacienda y del gobierno de la comunidad autónoma.

El nivel de calidad del software desarrollado por el CENPRI fue reconocido a nivel nacional con un premio del Ministerio de Industria y Energía hace dos años. Una reciente evaluación 13 de la madurez de su Proceso de Mantenimiento de acuerdo con el *IT Service Capability Maturity Model* 12 situó a esta organización en el segundo nivel CMM (Repetible). Aunque el lector puede pensar que éste no es un nivel adecuado para una organización como ésta, la realidad es que la entidades públicas probablemente tienen un conjunto de especiales características que pueden hacer innecesario alcanzar niveles más altos. Aunque esta idea es rebatible y fácilmente discutible, se encuentra desarrollada en la ya mencionada referencia 13.

El sistema analizado consta de 143 programas escritos en CA-OpenIngres/4GL, y está compuesto principalmente por sentencias de manipulación de datos escritas en SQL. El sistema fue desarrollado completamente por el mismo equipo de trabajo, que también lo mantiene. Estos factores pueden ser considerados como una constante y, por tanto, la fiabilidad de este estudio puede considerarse alta 3.Los datos utilizados en este estudio correspondientes a los 143 programas se encuentran representados en una tabla, cuyas primeras líneas se encuentran a continuación:

Programa	NS	NI	ND	NU	NA	NT	WHERE	TIME
timer_on.osq	0	0	0	0	0	1	0	5
compr_li.osq	1	0	0	0	2	5	27	65
consulta.osq	12	0	0	0	3	10	65	130
cont000.osq	3	0	0	0	1	3	5	30
...	...	...	...	...	...	...	...	...

Tabla 1. Datos históricos.

4. Análisis de los datos usando Estadística Descriptiva

Inicialmente se aplicó a los datos un modelo de regresión lineal multivariable. Los resultados mostraron que sólo tres de las siete métricas recogidas tenían una correlación significativa con la variable dependiente TIME. En 11 se puede encontrar una descripción más detallada del experimento, incluyendo los distintos coeficientes y algunos comentarios adicionales. La ecuación lineal que en aquel estudio se obtuvo para predecir el tiempo de mantenimiento es:

$$TIME=10.01*NA + 10.22*NT+ 0.08*WHERE - 9.84$$

Ecuación 1. Ecuación para predecir TIME.

Como se puede observar, sólo NA, NT y WHERE están significativamente correlacionadas con TIME. Los coeficientes de correlación son:

Métrica	Correlación con TIME
NS	0.29
NI	0.39
ND	0.13
UN	0.08
NA	0.88
NT	0.99
WHERE	0.96

Tabla 2. Correlación de las métricas con TIME.

5. Análisis de los datos mediante técnicas de Minería de Datos

La minería de datos es un campo interdisciplinar con el objetivo general de predecir resultados y/o descubrir relaciones en los datos. El énfasis de la minería de datos se sitúa más en la comprensión de los datos que en la predicción por sí misma. Así, las técnicas basadas en reglas de decisión, árboles de decisión, reglas de asociación, etc. se centran en encontrar modelos que sean fácilmente interpretables. Los métodos estadísticos pueden proporcionar información válida, aunque en algunos casos no proporcionan un conocimiento tan comprensible como la minería de datos.

De la regresión lineal global de la Ecuación 1, se obtiene un error medio sobre el conjunto de ejemplos de 3.8622 (6.28% de error relativo medio) usando una validación cruzada de 10 celdas. Para establecer la comparación se han usado dos técnicas de minería de datos con objetivos diferentes: por un lado una técnica orientada a predicción, denominada M5, que genera un árbol de decisión en el que a cada hoja se le asocia una regresión lineal; y por otro lado HIDER, una técnica propia, que encuentra un conjunto de reglas de decisión mediante un algoritmo evolutivo. Las reglas de decisión proporcionan un conocimiento distinto al de una herramienta de predicción, pues en principio necesita que la variable a analizar (TIME en este caso) se encuentre discretizada, estableciendo entonces una relación entre los valores de las métricas y la etiqueta correspondiente de TIME.

5.1. Análisis con M5

M5 es una herramienta bien conocida por los investigadores de minería de datos 14, que construye árboles de decisión cuyas hojas tienen un modelo de regresión lineal, resultando entonces un modelo análogo a una función lineal a trozos. M5' escoge el test (nodos no hojas) sobre el atributo que maximiza la reducción del error esperado como una función de la desviación estándar de la variable de salida.

La versión de M5 utilizada es M5', programa implementado en la librería WEKA 18. La aplicación de M5' sobre los datos proporciona el árbol de regresión que se muestra en la Figura 1: este árbol reduce el error a 2.8 (4.5% en términos relativos) dando , por tanto, una mejor predicción que la regresión lineal. M5' encuentra seis categorías de la variable dependiente (LM1 a LM6). TIME en algunas categorías recibe un valor constante y en otras debe ser calculado mediante una ecuación lineal.

```
--NT <= 5.5 :                          con:
|  NA <= 1.5 :                         LM1:  TIME = 5.48
|  |  NT <= 1.5 : LM1                  LM2:  TIME = -10.2 + 10.1NA + 9.88NT
|  |  NT >  1.5 :                      LM3:  TIME = 31.4
|  |  |  NT <= 3.5 : LM2               LM4:  TIME = 42.1
|  |  |  NT >  3.5 :                   LM5:  TIME = -15.2 + 13.8NA + 9.64NT
|  |  |  |  WHERE <= 8.5 : LM3         LM6: TIME = -11.3 + 10.3NA + 11NT - 0.887NI
|  |  |  |  WHERE >  8.5 : LM4
|  NA >  1.5 : LM5
+--NT >  5.5 : LM6
```

Figura 1. Resultados proporcionados por M5'.

Como información adicional, el número de programas que fueron clasificados en cada categoría por M5' fueron: *LM1*: 21 programas; *LM2*: 34 programas; *LM3*: 7 programas; *LM4*: 7 programas; *LM5*: 27 programas; *LM6*: 47 programas

5.2. Análisis con HIDER

HIDER 16 es un sistema basado en un algoritmo evolutivo (AE) que proporciona un conjunto de reglas de decisión a partir de una base de datos etiquetada. Una regla de decisión es una cláusula de la forma:

$$Si\ p_1 \in [2.1,3.7]\ y\ p_2 \in [6.1,9.7]\ y\ ...entonces\ variable\ es\ A$$

donde los p_i son parámetros independientes, *variable* es una variable dependiente y *A* un estado de la misma.

El AE se usa como método para encontrar una buena solución en el complejo espacio de búsqueda de relacionar los parámetros (en este caso los valores de las métricas) con valores de la variable dependiente. La aplicación de un AE a un problema de aprendizaje requiere la selección de una representación interna del espacio de búsqueda y la definición de una función externa que asigne un valor de bondad o adaptación de las soluciones candidatas. Ambas componentes son críticas para la correcta aplicación de un AE a cualquier problema. El algoritmo escoge el mejor individuo del proceso evolutivo y lo transforma en una regla que es usada para eliminar los datos que la cumplen del fichero de entrenamiento. De esta manera, el fichero de entrenamiento es reducido para la siguiente iteración y así la siguiente regla no tiene en cuenta los puntos eliminados, encontrando otra regla para el resto de puntos y así sucesivamente. El criterio de terminación es que no existan más puntos a cubrir en el fichero de entrenamiento.

5.2.1. Discretización de la variable dependiente

Para poder aplicar HIDER, la variable dependiente debe ser discretizada en un conjunto de categorías disjuntas. En la literatura de minería de datos existen distintos métodos para este etiquetado:

- La aplicación de una técnica como los k-vecinos más cercanos (k-NN) 6para clase continua encuentra tres categorías, mostradas en la Figura 2. Asimismo los resultados de predicción confirman que las tres últimas métricas (NA, NT y WHERE) tienen una influencia mayoritaria en la variable TIME. Los resultados de predicción son muy parecidos a los de la regresión lineal, aunque con un error mayor

debido sobre todo a que la regresión lineal ajusta mejor los valores grandes de TIME para reducir el error absoluto, fallando más en los valores bajos de TIME (sobre todo 5 y 10). Por el contrario, k-NN ajusta muy bien los valores bajos de TIME pero falla en los valores altos. Esto se debe a que en los valores bajos existen muchos puntos con igual valor de TIME mientras que para los valores altos la frecuencia relativa es 1 y, por tanto, la estimación de un nuevo valor es más complicado. Por tanto, k-NN ha discretizado TIME basándose en los valores que producen error.

Etiqueta A, cuando **TIME<=40**	(74 programas)
Etiqueta B, cuando **40<TIME<=105**	(33 programas)
Etiqueta C, cuando **TIME>105**	(36 programas)

Figura 2. Discretización de la variable dependiente mediante la técnica k-NN.

- Una discretización manual también es posible. Para poder comparar HIDER y M5 y teniendo en cuenta que ésta encontraba seis categorías se discretizó la variable TIME en 6 intervalos de forma que cada uno contenga el mismo número de programas. Estos intervalos son mostrados en la Figura 3.

Etiq. A, cuando **TIME <= 10**	(29 programas)
Etiq. B, cuando **10 < TIME <= 25**	(20 programas)
Etiq. C, cuando **25 <TIME <= 40**	(25 programas)
Etiq. D, cuando **40 <TIME <=75**	(23 programas)
Etiq. E, cuando **75 < TIME <=150**	(21 programas)
Etiq. F, cuando **TIME > 150**	(25 programas)

Figura 3. Discretización manual de la variable dependiente.

5.2.2. Aplicación de HIDER

Una vez que la variable ha sido discretizada y se le ha asignado una etiqueta (de acuerdo con la Figura 3), aplicamos HIDER para extraer un conjunto de reglas. La salida es un conjunto de reglas de decisión que indica que intervalos de valores de los parámetros de entrada (los valores de las métricas) dan lugar a determinadas etiquetas de la variable TIME. HIDER dispone de un parámetro definido por el usuario para ajustar el error esperado durante el proceso de aprendizaje. Con este parámetro el usuario puede intervenir en el número de reglas que HIDER proporciona, que lógicamente es inversamente proporcional al número de errores de predicción permitidos durante el entrenamiento: si se necesita una tasa de error mínima entonces este parámetro será situado a cero y el número de reglas crecerá.

HIDER es capaz de proporcionar dos clases de resultados: un solo conjunto jerárquico de reglas que involucra a todas las etiquetas de clase o un conjunto de reglas parar cada etiqueta de manera independiente. El primer caso es útil cuando el propósito es clasificar nuevos casos y la legibilidad de las reglas es menos importante. En el segundo caso se proporciona para cada etiqueta de manera independiente un conocimiento comprensible acerca de la influencia de los parámetros de entrada sobre la variable discretizada.

En 1, la calidad de las reglas proporcionadas por HIDER fue comparada con la conocida herramienta C4.5 15. De este estudio se deduce que HIDER obtiene mejores resultados que C4.5 dado que el número de reglas proporcionadas fue menor, a la vez

que cubrían más registros. El propósito de nuestro análisis mediante HIDER es conseguir un mejor conocimiento de las relaciones existentes entre las métricas definidas y la variable TIME (reglas para cada etiqueta de manera independiente), por tato la segunda de las opciones señaladas fue la usada. Para ello, HIDER fue ejecutado seis veces (una para cada etiqueta), de forma que en cada ejecución la población inicial del AE sólo contenía reglas para esa etiqueta. Con una tamaño de población de 100 y 100 generaciones, el coste computacional de cada ejecución es muy bajo (menos de un minuto en un Pentium II 450MHz). La Tabla 3 muestra las reglas obtenidas de la aplicación de HIDER a los datos.

6. Discusión de los resultados

La Tabla 4 resume los resultados de aplicar las tres técnicas a los datos:

- TIME es el observado valor de la variable dependiente.
- LR es el tiempo estimado mediante regresión lineal.
- M5' es el valor de TIME estimado por M5'.

Para HIDER, hemos señalado la categoría estimada y si es o no correcta. Las predicciones de HIDER son correctas en un 95,10% de los casos. Los tres métodos de estimación son muy diferentes, pero es posible establecer comparaciones entre ellos. Es posible realizar una comparación objetiva entre M5' y la regresión lineal usando estimadores estadísticos. El hecho de que HIDER proporcione sus resultados en forma de intervalos dificulta su comparación con otras técnicas; no obstante, también realizamos alguna discusión.

6.1. LR y M5'

Para comparar la bondad de estas técnicas de predicción usaremos la magnitud media del error relativo (MMRE en inglés) y PRED(q), propuestas por Conte et al. 5:

- MMRE es definida por la Ecuación 2. En una muestra de tamaño n, $\hat{e}_i$ es el valor estimado para el i-ésimo elemento, y e_i es el valor real. Conte et al. sugieren que un valor aceptable para MMRE es un valor menor o igual a 0.25.
- PRED(q), siendo q un porcentaje, es el número de casos cuyas estimaciones están por debajo de q, dividido por el número total de casos. Por ejemplo, si PRED(0.1)=0.9, significa que el 90% de los casos tienen estimaciones dentro del 10% de su valor actual. Para 5, una estimación técnica es aceptable si PRED(0.25)>=0.75. PRED(q) es calculado mediante la Ecuación 2.

$$MMRE = \frac{1}{n} \cdot \sum_{i=1}^{n} \left| \frac{e_i - \hat{e}_i}{e_i} \right|$$

$$PRED(q) = \frac{k}{n}$$

donde:

k es el número de elementos en la muestra cuyo MMRE es menor o igual que q.
n es el número de elementos de la muestra.

Ecuación 2. MMRE y PRED(q).

NS	NI	ND	NU	NA	NT	WHE-RE	Cla-se	Registros/ errores
				=0	<=2		A	25 / 0
				=0	=3		B	13 / 0
	<=1			=0	>=4		C	6 / 1
				<=1	<=1		A	3 / 0
				>=1	>=5	$\in$ [12, 29]	D	3 / 0
				=1	=2		B	6 / 0
				=1	>=3	<=18	C	13 / 1
	<=1			=1	>=3		C	11 / 1
				$\in$ [2,3]	<=5	>=17	D	17 / 1
	=0			<=2	<=4	<=18	C	5 / 1
		<=2		>=2	<=6	$\in$ [17,39]	D	18 / 1
				=2	<=3	>=10	C	5 / 1
		<=2		>=2	$\in$ [4,6]	<=39	D	18 / 1
				=3	<=3		D	2 / 0
					$\in$ [4,5]	<=15	C	8 / 1
					$\in$ [6,11]	>=35	E	19 / 1
>=7					>=12		F	24 / 0

Tabla 3. Resultados proporcionados por HIDER.

				HIDER	
NAME (.osq)	TIME	LR	M5'	Clase	Co-rrecto
borra_u	75	73.53	74.64	$\in$ (40, 75]	1
compr_li	65	63.44	60.76	$\in$ (40, 75]	1
consulta	130	127.59	129.6	$\in$ (75, 150]	1
cont000	30	31.23	29.54	$\in$ (25, 40]	1
cont100	30	31.23	29.54	$\in$ (25, 40]	1
cont101	95	95.06	94.64	$\in$ (75, 150]	1
cont102	75	72.62	75.3	$\in$ (40, 75]	1
cont103	160	158.68	161.01	>150	1
cont104	30	31.23	29.54	$\in$ (25, 40]	1
cont105	30	31.44	31.4	$\in$ (25, 40]	1
cont110	5	0.78	5.48	<=10	1
cont120	65	63.47	65	$\in$ (40, 75]	1
cont121	105	107.1	106.9	$\in$ (75, 150]	1
...	...	...	...	...	...

Tabla 4. Resultados de todas las técnicas

Con estos estimadores, el comportamiento predictivo de ambas técnicas es bueno. Los valores son:

	MMRE	PRED(0.25)
LR	19.46%	81.81%
M5'	4.5%	95.80%

Tabla 5. Estimadores de error para LR y M5'.

Es evidente que la bondad de ajuste de M5' es mayor que la de una regresión lineal. La causa del alto valor de MMRE en la LR está en los valores bajos de TIME, dado que la predicción lineal da un tiempo menor de 1, aun cuando la media del valor real es 5.

Más aun, M5' descubre algunos factores que permanecían ocultos con la regresión lineal, tales como la relativa importancia de algunas de las métricas dependiendo de las características del programa: ambas técnicas destacan NT, NA y WHERE como las únicas métricas representativas para la estimación del esfuerzo, pero en realidad NA y WHERE tienen sólo influencia cuando el número de Tablas es bajo (5 o menos). Por otra parte, cuando el número de Anidamientos es 2 o más, entonces el valor de WHERE no tiene influencia. Estas reglas pueden ayudar a los programadores a escribir software de más fácil mantenimiento..

6.2. LR, M5' y HIDER

Dependiendo de la importancia de la exactitud en la estimación (el punto de vista cuantitativo), es obvio que tal vez dar un intervalo para la variable TIME como predicción del esfuerzo no sea suficiente. Esta cuestión, depende de la organización y su situación. Sin embargo, en términos cualitativos, la alta fiabilidad de la técnica HIDER proporciona una información con un valor añadido. Las reglas encontradas por HIDER pueden ser expresadas textualmente como siguen:

- TIME toma valor menores o iguales a 10 si NA y NT toman valores muy bajos, concretamente si ambos parámetros no suman más de 2.
- Si la suma de NA y NT es tres entonces TIME estará entre 15 y 25.
- La variable TIME tomará valores entre 30 y 40 en tres casos posibles dependiendo de NA:
  - Si NA es igual a 0 y NT es mayor de 3 y WHERE es menor de 16.
  - Si NA es igual a 1 y NT toma valores entre 3 y 5.
  - Si NA es igual a 2 y NT es menor de 3 y WHERE es mayor de 9.
- Los valores de TIME están entre 45 y 75 en tres escenarios:
  - Cuando NA es igual a 2 o 3 y NT es menor que 5 y WHERE es mayor de 16.
  - Si ND es menor o igual a 2, NA mayor de 1, NT menor de 6 y WHERE entre 17 y 39.
  - Por último, si NA es mayor o igual a 1, NT mayor de 4 y WHERE está entre 12 y 29.
- Si NT toma valores entre 6 y 11 y WHERE es mayor de 34 entonces TIME estará entre 80 y 150.
- Finalmente, TIME será mayor de 150 si NS y NT son mayores de 6 y 12, respectivamente.

En resumen, NT es la métrica más decisiva para TIME: Si NT es igual a 0, 1 ó 2 TIME tomará valores menores de 25, dependiendo también del valor de NA. Si NT toma valores entre 3 y 6 entonces TIME estará entre 30 y 75, dependiendo de los valores de NA y WHERE. Si NT está entre 6 y 11 entonces TIME estará en [80, 150] y, por último, TIME es mayor de 150 cuando NT toma valores mayores de 12.

7. Conclusiones

En este trabajo se presentan los resultados de aplicar tres técnicas diferentes (regresión lineal, M5' y HIDER) para predecir el esfuerzo de mantenimiento de programas 4GL. Desde un punto de vista cuantitativo, la técnica de minería de datos M5' se muestra mucho mejor que la regresión lineal. Cuando sea importante predecir el valor de TIME, entonces M5' debe ser usada.

Por otro lado, como ha sido discutido en las subsecciones 6.1 y 6.2, las reglas proporcionadas por M5' y HIDER proporcionan un valor añadido sobre las características de los programas. El valor y los intervalos facilitados para TIME, junto con los valores de las métricas, pueden ayudar a los programadores y gestores de proyectos a situar los umbrales para escribir programas con unos niveles altos de calidad de mantenimiento. Aunque nuestros resultados no puedan ser generalizados a todas las situaciones, la utilización de otras técnicas aparte de la estadística clásica se demuestra correcta para la construcción de modelos de predicción: no sólo por una mayor bondad en la estimación sino también por el conocimiento e información adicional que

proporcionan. Otros investigadores pueden descargarse los datos utilizados en este trabajo de http://www.inf-cr.uclm.es/www/mpolo/varios/4gl.xls

8. Agradecimientos

Este trabajo se encuentra parcialmente financiado por los proyectos TIC2001-1143-C03-02 y TAMANSI (Consejería de Ciencia y Tecnología de la Junta de Comunidades de Castilla-La Mancha, PBC-02-001).

9. Referencias

1. Aguilar-Ruiz J, Ramos I, Riquelme JC y Toro M. (2001). An evolutionary approach to estimating software development projects. *Information and Soft. Tech.*, 43(14), 875-882.
2. Bourque P y Côté V. (1991). An experiment in software sizing with structured analysis metrics. *Journal of Systems and Software*, 15, 159-172.
3. Basili V, Shull F y Lanubile F. (1999). Building knowledge through families of experiments. *IEEE Transactions on Software Engineering*, 25(4), 453-473.
4. Briand, LC, Morasca S y Basili V. (1996). Property-based software engineering measurement. *IEEE Transactions on Software Engineering*, 22(1) 68-85.
5. Conte SD, Dunsmore HE y Shen V. (1986). Software Engineering Metrics and Models, Benjamin/Cummings.
6. Dasarathy, B.V. Ed. (1991). *Nearest Neighbor (NN) Norms: NN pattern classification techniques*, IEEE Computer Society Press.
7. Dolado J.J. (1997). A Study of the Relationships among Albrecht and Mark II Function Points, Lines of Code 4GL and Effort. *J. Systems Software*, 37, 161-173.
8. Holloway S. (1990). Fourth-Generation Systems, their scope application and methods of evaluation. London: Chapman Hall.
9. Leavit N. (2000). Whatever happened to Object-Oriented Databases? *IEEE Computer*, 33(8), 16-19.
10. Martínez A y Piattini M. (2001). *Validation of measures to assess the maintainability of SQL programs*. Proc. of the 11th ESCOM-SCOPE Conference.
11. Martínez A y Piattini M. (2001). *Measuring for Database Programs Maintainability*. Proc. of the 11th DEXA conference. LNCS, vol. 1873, pp. 65-78.
12. Niessink, F. y van Vliet, H. (1999). *The Vrije Universiteit IT Service Capability Maturity Model*. TR IR-463, rel. L2-1.0. Vrije Universiteit, Holanda.
13. Polo M, Piattini M, Ruiz F y Jiménez M. (2001). *Assessment of Maintenance Maturity in IT Departments of Public Entities: Two Case Studies*. Product Focused Software Process Improvement. LNCS, vol. 2188, pp. 86-97.
14. Quinlan JR. (1992). *Learning with continuous class*. Proc. of the 5th Australian Joint Conference on Artificial Intelligence, pp. 343-348, World Scientific.
15. Quinlan, J.R. (1993). *C4.5: Programs for Machine Learning*. Morgan Kaufmann Pub., Inc..
16. Riquelme JC, Aguilar J. y Toro M. (2000). Discovering hierarchical decision rules with evolutive algorithms in supervised learning. *Int. J. of Computer, Systems and Signal*, 1(1), 73-84.
17. Verner J y Tate G. (1988). Estimating size and effort in Fourth-Generation Development. *IEEE Transactions on Software Engineering*, 5(4), 15-22.
18. Witten I. y Frank E. (1999). Data Mining Practical: Machine Learning Tools and tecniques with Java implementations. Morgan Kaufmann.
19. Zuse H. (1998). A Framework for Software Measurement. Walter de Gruyter.