Validación empírica de métricas para diagramas de clases a través de experimentos controlados

Marcela Genero*, Mario Piattini*, Tamara Rezk**, Coral Calero *

*Departamento de Informática
Universidad de Castilla-La Mancha
Ronda de Calatrava, 5
13071, Ciudad Real (España)
{mgenero, mpiattin, ccalero}@inf-cr.uclm.es

**Departamento de Ciencias de la Computación
FaMAF. Universidad Nacional de Córdoba
Av. Valparaíso y Rogelio Martínez
5000, Ciudad Universitaria, Córdoba (Argentina)
trezk@hal.famaf.unc.edu.ar

Resumen. Los diagramas de clase son un elemento clave en el desarrollo de sistemas de información orientados a objetos (SIOO), ya que constituyen la base del diseño y posterior implementación. De ahí, que asegurar la calidad de los diagramas de clases en las etapas iniciales del ciclo de vida sea realmente un reto en pro de lograr SIOO de mejor calidad. Nosotros centraremos este trabajo en una de las características de la calidad más críticas, la mantenibilidad. Pero como la mantenibilidad es un atributo externo de la calidad que solo puede medirse una vez que se ha terminado el producto, nuestra idea es presentar un conjunto de métricas para evaluar la complejidad estructural (un atributo interno de la calidad) de los diagramas de clases realizados utilizando el Lenguaje Unificado de Modelado (UML). Estas métricas pueden ayudar al diseñador a tomar mejores decisiones en las etapas iniciales, cuando descubrir y corregir un error es menos costoso que a medida que avanza el desarrollo. Para demostrar que realmente las métricas propuestas están relacionadas con la mantenibilidad, las sometimos a validación empírica a través de un experimento controlado. La explicación de cómo realizamos dicho experimento y cómo analizamos sus resultados son los objetivos principales de este trabajo.

Palabras claves. calidad de los SI, diagramas de clases en UML, complejidad estructural, mantenibilidad, métricas, validación empírica, validación teórica

1. INTRODUCCIÓN

En el campo de la medición del software se ha hecho un gran esfuerzo persiguiendo el objetivo de obtener sistemas de información orientados a objetos (SIOO) de mejor calidad (Henderson-Sellers, 1996; Melton, 1996; Zuse, 1998; Fenton y Pflegger, 1997), pero la mayoría de las propuestas existentes persiguen la meta de evaluar -en términos de medidas cuantitativas- la calidad del producto final, i.e. el código o el diseño avanzado. Nosotros creemos que para

alcanzar una mejor calidad de los SIOO deberíamos centrarnos en métricas que permitan evaluar las características de la calidad de diagramas, como los diagramas de clases, que se realizan en las etapas iniciales del ciclo de vida de un SIOO. De esta manera aseguraríamos, en cierta forma, la calidad de los SIOO desde el inicio de su desarrollo.

Como los diagramas de clase constituyen un elemento clave en el desarrollo de SIOO, su calidad es crucial ya que su tiene un gran impacto sobre la calidad de los SIOO, finalmente implementados. En respuesta a la gran demanda de métricas para medir características de calidad de los diagramas de clase, tales como la mantenibilidad (ISO, 1999) y luego de realizar una minuciosa revisión de algunas métricas OO existentes que pueden ser aplicadas en una etapa de diseño de alto nivel (Chidamber y Kemerer, 1994; Lorenz y Kidd, 1994, Brito e Abreu y Carapuça, 1994; Marchesi, 1998), hemos propuesto un conjunto de métricas para la complejidad estructural de los diagramas de clase realizados en UML en Genero et al. (2000a). Como la mantenibilidad es una característica de calidad externa que puede ser evaluada una vez que el producto está terminado, nos centramos en medir una característica interna de la calidad, la complejidad estructural de los diagramas de clase. Una vez validadas estas métricas, nuestra idea es usarlas para construir un modelo de predicción para la mantenibilidad de los diagramas de clase en las etapas iniciales del ciclo de vida de los SIOO. Teniendo en cuenta el Estándar ISO 9126 (ISO, 1999) consideramos que la mantenibilidad se ve influenciada por tres subcaracterísticas:

- Comprensibilidad: Facilidad con la que el diagrama de clases puede ser entendido.
- Analizabilidad: Facilidad que ofrece el diagrama de clases para descubrir sus deficiencias o errores.
- Modificabilidad o Cambiabilidad: Facilidad que ofrece el diagrama de clases para realizar una modificación especificada, ya sea por un error, por un concepto no tenido en cuenta o por un cambio en los requisistos.

Hemos obtenido dichas métricas de forma metodológica (Calero et al., 2001), realizando tres tareas principales: definición, validación teórica y validación empírica de las métricas. A pesar de que las tres tareas son relevantes para definir métricas correctas y fiables, nos centraremos en este artículo solo en la validación empírica.

La validación empírica es crítica para el éxito de cualquier proyecto de medición (Kitchenham et. al., 1995; Fenton y Pflegeer, 1997; Schneidewind, 1992; Basili et al., 1999). A través de la validación empírica podemos demostrar con evidencia real que las medidas que proponemos sirven para el propósito para el que fueron definidas y que realmente son útiles en la práctica. Dicha validación se puede realizar a través de experimentos controlados o a través de casos de estudios realizados con "datos reales". Ambos son relevantes, los primeros son útiles para obtener resultados preliminares y los últimos para obtener conclusiones finales.

En este artículo se persiguen los siguientes objetivos:

1. Presentar un conjunto de métricas para la complejidad estructural de los diagramas de clase realizados con UML (ver sección 2).

2. Mostrar un experimento controlado que hemos realizado para evaluar si hay evidencia empírica de que las métricas de la complejidad estructural de los diagramas realizados utilizando UML están correlacionadas con su mantenibilidad (ver sección 3).

Finalmente en la sección 4, presentamos algunas conclusiones y líneas de trabajo futuras relacionadas con las métricas para el modelado conceptual orientado a objetos usando UML.

2. DEFINICIÓN DE MÉTRICAS PARA LA COMPLEJIDAD ESTRUCTURAL DE LOS DIAGRAMAS DE CLASE EN UML

Solamente presentamos aquí aquellas métricas definidas en Genero et al. (2000a) que pueden aplicarse a nivel de diagrama de clases como un todo (ver tabla 1). Estas métricas miden la complejidad estructural de los diagramas de clase en UML debido al uso de relaciones, como asociaciones, generalizaciones, agregaciones y dependencias. También consideramos las métricas tradicionales tales como, el número de clases, el número de atributos, etc.

Nombre de la métrica	Definición de la métrica
NÚMERO DE CLASES (NC)	Número total de clases.
NÚMERO DE ATRIBUTOS (NA)	Número total de atributos.
NÚMERO DE MÉTODOS (NM)	Número total de métodos.
NÚMERO DE ASOCIACIONES (NAssoc)	Número total de relaciones de asociación.
NÚMERO DE AGREGACIONES (NAgg)	Número total de relaciones de agregación (cada par parte-todo en una relación de agregación)
NÚMERO DE DEPENDENCIAS (NDep)	El número total de relaciones de dependencia.
NÚMERO DE GENERALIZACIONES (NGen)	Número total de relaciones de generalización (cada par padre-hijo en una relación de generalización)
NÚMERO DE JERARQUÍAS DE GENERALIZACIÓN (NgenH)	Número total de jerarquías de generalización en un diagrama de clase.
DIT MÁXIMO	Es el valor DIT máximo obtenido para cada clase de un diagrama de clase. El valor DIT para una clase dentro de una jerarquía de generalización es la longitud el camino más largo desde la clase hasta la raíz de la jerarquía.
HAGG MÁXIMO	Es el valor HAgg máximo obtenido para cada clase del diagrama de clase. El valor HAgg para una clase dentro de una jerarquía de agregación es la longitud el camino más largo desde la clase hasta las hojas.

Tabla 1. Métricas para la complejidad estructural de diagramas de clases en UML

Estas métricas pueden ser útiles a los diseñadores de SIOO para:

1. Comparar cuantitativamente alternativas de diseño, y por lo tanto una selección objetiva entre varios diagramas alternativos que sean semánticamente equivalente.
2. Predecir características de la calidad, como la mantenibilidad en las fases iniciales del ciclo de vida de los SIOO y realizar una mejor asignación de los recursos basada en tales predicciones.

3. VALIDACIÓN EMPÍRICA DE LAS MÉTRICAS PROPUESTAS

En esta sección describiremos un experimento que hemos llevado a cabo para validar empíricamente las métricas propuestas (ver sección 2). Como Wholin et al. (2000) establecieron, seremos capaces de extraer conclusiones con respecto a la relación causa-efecto para la que establecimos una hipótesis (que queremos corroborar a través de experimentos), solo si el experimento es realizado de forma apropiada. Por lo tanto, hemos seguido algunas sugerencias provistas por Wholin et al. (2000), Perry et al. (2000) y Briand et al. (1999) con respecto a cómo realizar experimentos controlados.

En el resto de esta sección explicamos cómo llevamos a cabo cada una de las actividades del experimento.

3.1 Definición

Como sugirieron Wholin et al. (2000), seguimos la plantilla de GQM (Basili y Weiss, 1984; Basili y Rombach, 1988) para la definición de los objetivos. Esto dio como resultado el siguiente objetivo:

Analizar	*Métricas para la complejidad estructural de los diagramas de clase realizados utilizando UML*
Con el propósito de	*Evaluar*
Con respecto a	*La capacidad de ser utilizados como indicadores de las subcaracterísticas de la mantenibilidad*
Desde el punto de vista de	*Diseñadores de SIOO*
En el contexto de	*Estudiantes de la Ingeniería en Informática de la Escuela Superior de Informática de la Universidad de Castilla-La Mancha*

3.2 Planificación

Una vez definido el objetivo del experimento se debe realizar la planificación. La definición determina el *por qué* del experimento, mientras que la planificación se refiere a *cómo* se realizará el experimento.

3.2.1 Selección del Contexto

El contexto del experimento es un grupo de alumnos de la universidad. El experimento fue realizado por 26 estudiantes que estaban cursando del tercer año de la Ingeniería en Informática de la Escuela Superior de Informática en la Universidad de Castilla-La Mancha en España.
El experimento es específico ya que se centra en métricas de la complejidad estructural de los diagramas de clase realizados en UML. La habilidad para generalizar los resultados del experimento partiendo de un contexto específico se analizará más adelante cuando se discuten las amenazas a la validez del experimento. El experimento apunta a un problema real, la correlación entre métricas y la mantenibilidad de los diagramas de clase.

3.2.2 Formulación de la Hipótesis

Un aspecto importante de los experimentos es saber establecer de manera clara y formal qué se intenta evaluar en el experimento. Esto nos conduce a la formulación de una hipótesis (o varias hipótesis). Deseamos corroborar dos hipótesis:

Primera hipótesis

Hipótesis nula, H_0: No existe una correlación significativa entre las métricas propuestas y el tiempo de mantenimiento.

Hipótesis alternativa, H_1: Existe una correlación significativa entre las métricas propuestas y el tiempo de mantenimiento.

Segunda hipótesis

Hipótesis nula, H_0: Existe una correlación significante entre las métricas propuestas y la evaluación que los sujetos hacen de las tres subcaracterísticas de la mantenibilidad: la comprensibilidad, la analizabilidad y la modificabilidad.

Hipótesis alternativa, H_1: Existe una correlación significante entre las métricas propuestas y la evaluación que los sujetos hacen de las tres subcaracterísticas de la mantenibilidad: la comprensibilidad, la analizabilidad y la modificabilidad.

3.2.3 Selección de las variables

La variable independiente es la complejidad estructural de los diagramas de clase en UML. Las variables dependientes son las subcaracterísticas de la mantenibilidad: la comprensibilidad, la analizabilidad y la modificabilidad.

3.2.4 Selección de los sujetos

Los sujetos fueron elegidos por conveniencia, i.e. los sujetos son estudiantes que poseen experiencia en el diseño y desarrollo de SIOO.

3.2.5 Diseño del Experimento

Todos los test del experimento fueron realizados por el mismo grupo de sujetos. Los test fueron entregados a cada sujeto en distinto orden.

3.2.6 Instrumentación

Los objetos fueron diagramas de clase realizados usando UML.

La variable independiente fue medida a través de las métricas, presentadas en la sección 2.

Las variables dependientes fueron medidas de dos maneras: Por un lado consideramos la valoración de los expertos usando una escala de etiquetas lingüísticas, y por el otro consideramos el tiempo que los sujetos tardaron en realizar una de las tareas del experimento que se refiere al cálculo de las métricas. A nuestro entender, dicho tiempo se ve enormemente influenciado por el

tiempo que tardan los sujetos en entender cada diagrama, el cual a su vez puede tener un gran impacto sobre el tiempo necesario para realizar tareas de mantenimiento, por lo que lo hemos llamado "tiempo de mantenimiento".

Esta elección de medir de dos formas distintas las variables independientes, se debe a que queríamos corroborar si las conclusiones obtenidas en ambos casos eran similares, tanto si utilizamos una medida objetiva como es el tiempo de mantenimiento o si usamos la apreciación subjetiva de los expertos.

3.3 Operación

3.3.1 Preparación

En el momento en que se realizó el experimento todos los estudiantes habían tomado dos cursos de Ingeniería del Software, en los cuáles aprendieron en profundidad cómo construir SIOO utilizando UML. Más aún, todos los sujetos recibieron un entrenamiento intensivo antes de la realización del experimento, respecto al modelado usando UML y a métricas para diagramas de clases. Los sujetos no fueron conscientes de los objetivos perseguidos por el experimento. Tampoco conocían la hipótesis establecida.

Preparamos el material que le dimos a los sujetos (ver ejemplo en el Apéndice A), que consistió en 8 diagramas de clase referidos al mismo dominio, relacionado a Sistemas de Información relativos a un consorcio de bancos. Los diagramas diferían en complejidad ya que se intentó cubrir un amplio rango de valores de las métricas. Cada diagrama tenía adjunto un test que indicaba la realización de dos tareas:

1. Coger cada diagrama, anotar el tiempo inicial (expresado en horas, minutos y segundos), calcular 11 métricas (ver sección 2), y una vez que calculaban todas las métricas debían anotar el tiempo final. La diferencia entre el tiempo inicial y final es lo que consideramos el tiempo de mantenimiento(expresado en minutos y segundos).
2. Valorar cada subcaracterística de la mantenibilidad utilizando una escala de siete etiquetas lingüísticas (ver Apéndice A).

3.3.5 Ejecución

Los sujetos recibieron todo el material descrito en la sección previa. Nosotros les explicamos cómo llevar a cabo el experimento. Cada sujeto debía realizar el experimento de forma individual y podía utilizar tiempo ilimitado para resolverlo.

Recogimos todos los datos del experimento incluyendo, los diagramas de clases con el cálculo de las métricas, el tiempo que anotaron los sujetos, la valoración de los sujetos de cada una de las sub-características de la mantenibilidad y los valores de las métricas que calculamos automáticamente utilizando la herramienta MANTICA (Genero et al., 2000b).

3.3.6 Validación de los datos

Una vez que recogimos todos los datos del experimento, controlamos cada uno de los tests para ver si estaban completos y si las métricas estaban bien calculadas. Descartamos los tests de 5

sujetos, porque o bien no estaban completos o alguna de las métricas no estaba bien calculada. Con lo cual solo consideramos válidos los tests de 21 sujetos.

3.4 Análisis e Interpretación

Utilizamos los datos obtenidos en el experimento para corroborar las hipótesis formuladas en la seccion 3.2.2.

En primer lugar, aplicamos el test Kolmogrov-Smirnov para averiguar si los datos recogidos tenían una distribución normal. Como la distribución de los datos no era normal, decidimos usar un test no paramétrico, el test de correlación de Spearman., con un nivel de significación α = 0.05, i.e. con un nivel de confianza del 95% (i.e. la probabilidad de que rechacemos H_0 siendo H_0 falsa es al menos del 95%, los cual es estadísticamente aceptable).

Para corroborar la primera hipótesis, usando el coeficiente de correlación de Spearman, cada métrica fue correlacionada con lo que consideramos el tiempo de mantenimiento, medido en minutos (ver tabla 2).

	NC	NA	NM	Nassoc	NAgg	NDep	NGen	NAggH	NGenH	MaxHAgg	MaxDIT
Tiempo de mantenimiento	0,642	0,642	0,671	0,488	0,589	0,099	0,676	0,681	0,641	0,635	0,676

Tabla 2. Correlación de Spearman entre las métricas propuestas y el tiempo de mantenimiento

Analizando los datos de la tabla 2, concluimos que hay una alta correlación entre las métricas propuestas y el tiempo de mantenimiento. Esto se deduce del hecho de que casi todos los coeficientes tienen un valor mayor a 0,6.

Para corroborar la segunda hipótesis cada métrica fue correlacionada con la valoración dada por los sujetos de para la comprensibilidad, la analizabilidad y la modificabilidad (ver tabla 3).

	NC	NA	NM	NAssoc	NAgg	NDep	NGen	NAgg H	NgenH	MaxHa gg	MaxDI T
Comprensibilidad	0,694	0,694	0,688	0,593	0,594	0,036	0,711	0,683	0,706	0,765	0,711
Analizabilidad	0,695	0,695	0,695	0,585	0,609	0,023	0,715	0,698	0,711	0,677	0,715
Modificabilidad	0,723	0,723	0,725	0,725	0,641	0,013	0,733	0,716	0,726	0,721	0,733

Tabla 3. Correlación de Spearman entre las métricas propuestas y la comprensibilidad, la analizabilidad y la modificabilidad.

Analizando los coeficientes de correlación de Spearman mostrados en la tabla 3, concluimos que hay una alta correlación entre la mayoría de las métricas de complejidad estructural de los diagramas de clase UML y la valoración dada por los sujetos para la comprensibilidad, la

analizabilidad y la modificabilidad. Podemos deducir ya que casi todas todos los coeficientes tienen un valor mayor a 0,5.

En ambos casos (ver tabla 2 y 3) NDep es la única métrica que tiene una correlación menor. Este hecho debería ser estudiado en detalle a través de más experimentación.

Además queremos destacar que tanto la valoración subjetiva de los expertos como la medida objetiva del tiempo de mantenimiento, están relacionadas con las métricas, que era otras de puntos que queríamos analizar.

3.5 Evaluación de las amenazas a la validez del experimento

A continuación discutiremos brevemente ciertos aspectos que pueden amenazar a la validez del experimento y de que forma tratamos de solucionarlos o aliviarlos:

- AMENAZAS A LA VALIDEZ DE CONSTRUCTO. Propusimos métricas subjetivas para medir cada una de las variables dependientes (subcaracterísticas de la mantenibilidad) basados en la valoración de los expertos (ver sección 3.2). Como los sujetos involucrados en el experimento tienen experiencia media en el diseño de diagramas de clase UML pensamos que sus respuestas pueden ser consideradas significantes. Las variables independientes (cada una de las métricas propuestas en la sección 2) que miden la complejidad estructural de los diagramas de clase pueden también ser consideradas constructivamente válidas, porque desde un punto de vista de la teoría de sistemas, un sistema se considera complejo si está compuesto de muchos (diferentes tipos de) elementos, con muchos (diferentes tipos de) (dinámicamente cambiantes) relaciones entre ellos (Poels y Dedene, 2000a).
- AMENAZAS A LA VALIDEZ INTERNA. Mirando los resultados del experimento podemos concluir que la evidencia empírica de la relación entre variables independientes y dependientes existe. Hemos abordado diferentes aspectos que podrían poner en riesgo la validez interna del estudio, tal como: diferencias entre sujetos, conocimiento del universo del discurso entre diagramas de clase, exactitud de las respuestas del sujeto, efectos de aprendizaje, efectos de la fatiga, efectos de persistencia y motivación del sujeto.
- AMENAZAS A LA VALIDEZ EXTERNA. Identificamos dos aspectos que pueden amenazar a la validez externa y que por ello limitan la posibilidad de generalizar los resultados obtenidos y tratamos de alivianarlos, ellos son: los materiales utilizados, las tareas requeridas y la selección de los sujetos. En general para extraer una conclusión final que pueda ser generalizada, necesitamos replicar este experimento con un número mayor de sujetos, incluyendo además a profesionales. A través de la replicación de los experimentos podremos obtener lo que Basili et al. (1999) llamaron "Un cuerpo de conocimiento", el cuál nos conducirá a confirmar si las métricas presentadas podrían realmente ser usadas como indicadores de la calidad en las etapas iniciales del ciclo de vida de los SIOO, y además si podrían ser utilizadas para predecir la mantenibilidad de los diagramas de clase.

3.6 Presentación

La difusión de los resultados experimentales y la forma en que son presentados son relevantes ya que hace a su disponibilidad. Nosotros publicamos los resultados obtenidos en este artículo, y

planeamos publicar un paquete de laboratorio en la WEB para que éste y otros experimentos estén disponibles para quienes quieran replicarlo.

4. CONCLUSIONES Y TRABAJO FUTURO

En este artículo, hemos presentado las métricas propuestas por Genero et al. (2000a), que sirven para medir la complejidad estructural de los diagramas de clase realizados utilizando UML, debido al uso de relaciones, como asociaciones, agregaciones, generalizaciones y dependencias. Estas métricas, disponibles desde las etapas iniciales del ciclo de vida de los SIOO, pueden ayudar a los diseñadores a tomar mejores decisiones en sus tareas de diseño, la cuál es una meta importante de cualquier propuesta de métricas que apunte a ser útil (Fenton y Neil, 2000).

Con el objetivo de corroborar si existe correlación entre los valores de estas métricas y la mantenibilidad de un diagrama de clases, llevamos a cabo un experimento controlado con alumnos de nuestra universidad que estaban cursando el tercer año de la Ingeniería Informática.

Analizando los datos obtenidos en el experimento, utilizando el coeficiente de correlación de Spearman, concluimos que la mayoría de las métricas propuestas están altamente relacionadas tanto con el tiempo de mantenimiento como con las subcaracterísticas de la mantenibilidad: la comprensibilidad, la analizabilidad y la modificabilidad, ya que en la mayoría de los casos el coeficiente era mayor que 0,5 (ver tablas 2 y 3).

No obstante, y a pesar de los resultados prometedores, hacia la búsqueda de métricas OO válidas aplicadas a una etapa de diseño de alto nivel, somos conscientes que necesitamos hacer más validación de dichas métricas, tanto empírica como teórica para obtener evidencia concluyente de la utilidad de las métricas propuestas. Para ello es necesario contar con datos sobre proyectos reales, como puede ser el tiempo utilizado en tareas de mantenimiento.

Una vez que las métricas propuestas sean refinadas (i.e. validadas o descartadas) planeamos construir un modelo de predicción para la mantenibilidad de los diagramas de clases, que reciba como entrada los valores de las métricas propuestas. Para ello utilizaremos técnicas utilizadas en inteligencia artificial, como es una extensión del "Knowledge Discovery in Databases" tradicional (Fayyad et al., 1996) llamado "Fuzzy prototypical knowledge Discovery" (Olivas y Romero, 2000), que se a utilizado en otros entornos, como son la predicción de incendios y diagnósticos médicos obteniendo muy buenos resultados.

Con respecto a la validación teórica de las métricas propuestas, nos queda pendiente utilizar el marco formal denominado DISTANCE, propuesto por Poels y Dedene (1999; 2000b), el cual a nuestro entender, es el mas apropiado para medidas OO.

Con respecto a la validación empírica estamos planeando otro experimento, en el cual le entregaremos a los sujetos varios diagramas de clases, con una breve especificación, y nuevos requisitos que se desean añadir a los diagramas. En es caso la variable dependiente, la mantenibilidad, será medida utilizando el tiempo utilizado para realizar las modificaciones necesarias para cumplir con los nuevos requisitos, que seguro será mas significativo que el tiempo de cálculo de las métricas.

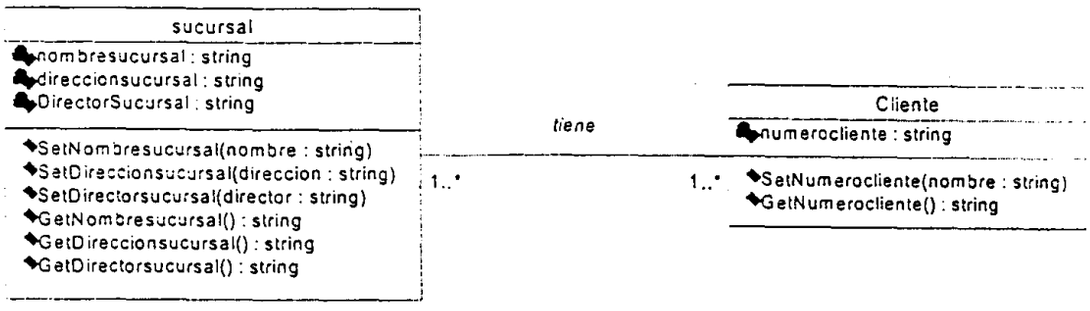
En trabajos futuros, también abordaremos la medición de otros factores de calidad como aquellos propuestos en la ISO 9126 (ISO, 1999), teniendo en cuenta no solo a los diagramas de clases, sino que también otros diagramas propuestos por UML, como son los diagramas de casos de uso, los diagramas de estado, etc. Según nuestro conocimiento, son escasos los trabajos realizados con respecto a la medición de modelos dinámicos y funcionales (Poels y Dedene, 2000a). Como se cita en Brito e Abreu et al. (1999) ésta es un área que necesita mayor investigación.

APÉNDICE A

A continuación mostramos como ejemplo uno de los test entregados a los sujetos.

Diagrama 1

Dado el siguiente diagrama de clases realizado con UML:



Realizar las siguientes tareas:

1) Anotar la hora de inicio: _____ (hora, minutos y segundos)

2) Calcular las siguientes métricas :

Número de clases	
Número de atributos	
Número de métodos	
Número de asociaciones	
Número de agregaciones	
Número de dependencias	
Número de generalizaciones	
Número de jerarquías de agregación	
Número de jerarquías de generalización	
DIT Máximo (profundidad)	
HAgg Máxima (altura)	

3) Anotar la hora de finalización: _____ (hora, minutos y segundos)

4) De acuerdo a su criterio valore cada uno de las siguientes subcaracterísticas de la mantenibilidad:

Comprensibilidad: Facilidad con la que el diagrama de clases puede ser entendido.

Extremada-mente Difícil de entender	Muy difícil de entender	Algo Difícil de entender	Ni Difícil Ni Fácil de entender	Algo Fácil de entender	Muy fácil de entender	Extremada-mente fácil de entender

Analizabilidad: Facilidad que ofrece el diagrama de clases para descubrir sus deficiencias o errores, y para identificar que partes deben ser modificadas.

Extremada-mente Difícil de analizar	Muy difícil de analizar	Algo Difícil de analizar	Ni Difícil Ni Fácil de analizar	Algo Fácil de analizar	Muy fácil de analizar	Extremada-mente fácil de analizar

Modificabilidad: Facilidad que ofrece el diagrama de clases para realizar una modificación especificada, ya sea por un error, por un concepto no tenido en por un cambio en los requisitos.

Extremada-mente Difícil de modificar	Muy difícil de modificar	Algo Difícil de modificar	Ni Difícil Ni Fácil de modificar	Algo Fácil de modificar	Muy fácil de modificar	Extremada-mente fácil de modificar

AGRADECIMIENTOS

Esta investigación es parte del proyecto DOLMEN soportado por la CICYT (TIC 2000-1673-C06-06) y SeCyT(UNC).

REFERENCIAS

- Basili V. y Weiss D. (1984). A Methodology for Collecting Valid Software Engineering Data, *IEEE Transactions on Software Engineering*, 10, 728-738.
- Basili V. y Rombach H. (1988). The TAME Project: Towards Improvement-Oriented Software Environments. *IEEE Transactions on Software Engineering* 14, 758-773.
- Basili V., Shull F. y Lanubile F. (1999). Building knowledge through families of experiments. *IEEE Transactions on Software Engineering*, 25(4), 435-437.
- Briand L., Bunse C. y Daly J. A Controlled Experiment for evaluating Quality Guidelines on the Maintainability of Object-Oriented Designs. *Technical Report IESE 002.99/E, Fraunhofer Institute for Experimental Software Engineering*, Kaiserslautern, Germany, (1999).
- Brito e Abreu F. y Carapuça R. (1994). Object-Oriented Software Engineering: Measuring and controlling the development process. *4th Int Conference on Software Quality*, Mc Lean, Va, USA.
- Brito e Abreu F., Zuse H., Sahraoui H. y Melo W. (1999). Quantitative Approaches in Object-Oriented Software Engineering. *Object-Oriented technology: ECOOP'99 Workshop Reader*, Lecture Notes in Computer Science 1743, Springer-Verlag, 326-337.
- Calero C., Piattini M. y Genero M. (2001). Metrics for controlling database complexity. In *Developing Quality Complex Databases*. Shirley Becker Ed. Idea Group Publishing.

- Chidamber S. y Kemerer C. (1994). A Metrics Suite for Object Oriented Design. *IEEE Transactions on Software Engineering*. 20(6), 476-493.
- Fayyad U., Piatetsky-Shapiro G. y Smyth P. (1996). The KDD Process for Extracting Useful Knowledge from Volumes of Data. *Communications of the ACM*, 39(11), 27 – 34.
- Fenton N. y Neil, M. (2000). Software Metrics: a Roadmap. *Future of Software Engineering*. Ed:Anthony Finkelstein, ACM, 359-370.
- Fenton N. y Pfleeger S. (1997). *Software Metrics: A Rigorous Approach*. 2nd. edition. London, Chapman & Hall.
- Genero, M., Piattini, M. y Calero, C. (2000a). Early Measures For UML class diagrams. *L'Objet*. 6(4), Hermes Science Publications, 489-515.
- Genero M., Piattini M., Rincón-Cinca J. y Serrano M. (2000b). MANTICA: Una Herramienta de Métricas para Modelos de Datos, *CACIC 2000*, Usuahia, Argentina.
- ISO/IEC 9126-1.2. (1999). Information technology- Software product quality – Part 1: Quality model.
- Henderson-Sellers B. (1996). *Object-Oriented Metrics - Measures of complexity*. Prentice-Hall, Upper Saddle River, New Jersey.
- Kitchenham, B., Pflegger, S. y Fenton, N. Towards a Framework for Software Measurement Validation. *IEEE Transactions of Software Engineering*, 21(12), (1995) 929-943.
- Lorenz M. y Kidd J. (1994). *Object-Oriented Software Metrics: A Practical Guide*. Prentice Hall, Englewood Cliffs, New Jersey.
- Marchesi M. (1998). OOA Metrics for the Unified Modeling Language. *Proceedings of the 2nd Euromicro Conference on Software Maintenance and Reengineering*, 67-73.
- Melton, A. (ed.) (1996). *Software Measurement*. London. International Thomson Computer Press.
- Olivas J. y Romero F. (2000). FPKD. Fuzzy Prototypical Knowledge Discovery. Application to Forest Fire Prediction. *Proceedings of the SEKE'2000*, Knowledge Systems Institute, Chicago, Ill. USA, 47 – 54.
- Perry D., Porte A. y Votta L. (2000). Empirical Studies of Software Engineering: A Roadmap. *Future of Software Engineering*. Ed. Anthony Finkelstein, ACM, 345-355.
- Poels G. y Dedene G. (1999). DISTANCE: A Framework for Software Measure Construction, research report DTEW9937, Dept. Applied Economics, Katholieke Universiteit Leuven, Belgium, 46 p.
- Poels G. y Dedene G. (2000a). Measures for Assessing Dynamic Complexity Aspects of Object-Oriented Conceptual Schemes. *In: Proceedings of the 19th International Conference on Conceptual Modeling (ER 2000)*, Salt Lake City, 499-512.
- Poels G. y Dedene G. (2000b). Distance-based software measurement: necessary and sufficient properties for software measures. *Information and Software Technology*, 42(1), 35-46.
- Schneidewind N. (1992). Methodology For Validating Software Metrics. *IEEE Transactions of Software Engineering*, 18(5), 410-422.
- Wohlin C., Runeson P., Höst M., Ohlson M., Regnell B. y Wesslén A. (2000) *Experimentation in Software Engineering: An Introduction*. Kluwer Academic Publishers.
- Zuse H. (1998). *A Framework of Software Measurement*. Berlin, Walter de Gruyter.