

SQL Programado desde Java: profundización en el diseño y acceso a bases de datos consolidando el conocimiento del lenguaje

Josep Maria Marco Simó

Profesor de los Estudios de Informática y Multimedia de la UOC

e-mail: jmarco@uoc.edu

Resumen

Se presenta la experiencia de desarrollo e implantación de los contenidos sobre SQL programado desde Java para la asignatura *Bases de Datos II*, asignatura que se imparte en los estudios de Informática de las Ingenierías Técnicas y Superior de la *Universitat Oberta de Catalunya -UOC-*.

Nuestra universidad se ha planteado la utilización de este lenguaje para abordar el tema de la programación de SQL dada su evidente implantación en entornos reales y la preferencia que otorga al aprendizaje de este lenguaje.

Este desarrollo, sin embargo, se ha enfocado con una serie de objetivos paralelos que van más allá del propio aprendizaje sobre SQL programado desde Java, y que pretenden que el estudiante desarrolle también otros conocimientos y habilidades adquiridos con anterioridad, en cuanto a:

- El diseño y uso de componentes lógicos de base de datos, así como su acceso.
- La perspectiva y/o introducción al mundo Java.
- El modelo SQL/CLI.

Exponemos en este documento la génesis, el diseño y la implementación de esta propuesta, así como sus resultados y sus perspectivas de futuro.

1. Motivación

En los planes de estudio de las Ingenierías Técnicas de Informática –especialidades Gestión y Sistemas- el aprendizaje sobre bases de datos se

inicia en la asignatura obligatoria *Bases de Datos I* [1], cuyos contenidos se resumen en:

- Ficheros.
- Introducción a las Bases de Datos.
- Modelo relacional y álgebra relacional.
- Introducción a SQL.
- Introducción al diseño de Bases de Datos.

En *Bases de Datos II* [2], obligatoria para la Ingeniería Técnica de Gestión y optativa para la Ingeniería Técnica de Sistemas y la Ingeniería Superior, se amplía este aprendizaje mediante el estudio de:

- Componentes lógicos de datos y de control (SQL avanzado).
- Componentes de almacenamiento.
- Implementación de métodos de acceso.
- Transacciones.
- Bases de datos distribuidas y cliente/servidor.
- Programación con SQL (SQL hospedado y SQL-CLI)

En ambas asignaturas se realizan prácticas. En la primera están centradas en el diseño de bases de datos y en el uso de SQL. En la segunda se pretende consolidar dos aspectos fundamentales que refuerzan, a su vez, lo aprendido anteriormente: la técnica CLI de SQL programado [3] y el uso avanzado de componentes lógicos de datos (tablas, vistas y restricciones) y componentes lógicos de control (procedimientos, disparadores, transacciones, control de acceso e índices) [4]

En el origen de la asignatura *Bases de Datos II*, los modelos de SQL programado se presentaban con aplicaciones en lenguaje C.

Aunque ésta es una aproximación que permite entender con claridad las características estándares de los modelos hospedado y CLI, también es cierto que presenta cierta complejidad en su implementación práctica.

Así, dado que el lenguaje Java es uno de los objetivos y de los lenguajes de referencia de los estudios de Informática de la UOC, se planteó ampliar las perspectivas sobre las técnicas de SQL programado, presentando también el acercamiento de Java al tema. Este acercamiento, por un lado, es relativamente más simple que el tradicional modelo en C y, por otro, permite introducir a los estudiantes en un tema de amplia implantación en los entornos profesionales: sólo hay que tener en cuenta su importancia en las aplicaciones de Internet y en los sistemas distribuidos en general.

Aquí pues tuvo su origen nuestra propuesta: presentar un modelo más actual y más sencillo para la programación con SQL.

Sin embargo, había que tener en cuenta otros aspectos:

En los primeros cuatrimestres en que se iba a implantar esta propuesta, iba a servir también como vía para introducir a muchos estudiantes en el lenguaje Java. Esto sería debido a que, al tratarse de una asignatura de los últimos cursos de las Ingenierías Técnicas y de los centrales de la Ingeniería Superior, muchos de ellos no conocerían el lenguaje por haber empezado los estudios cuando Java todavía no aparecía en las asignaturas de inicio.

Adicionalmente, se pretendía reincidir en las destrezas de diseño adquiridas anteriormente en *Bases de Datos I* y cuyo refuerzo pasa siempre por la vía de los ejercicios prácticos.

Teniendo en cuenta todas estas consideraciones, se confeccionaron unos contenidos adicionales [5] a la asignatura *Bases de datos II* con la idea de integrarlos. En resumen, los objetivos que se pretendían cubrir eran:

- 1) Presentar la programación SQL desde Java.
- 2) Profundizar en el diseño, uso de componentes lógicos y acceso a bases de datos.
- 3) Ofrecer una perspectiva y/o introducción al mundo Java.
- 4) Presentar a los estudiantes un sistema simplificado del modelo SQL/CLI.

2. Construcción de la propuesta

El proceso seguido para construir la propuesta tuvo dos partes diferenciadas. En primer lugar se verificaron las posibilidades prácticas de la programación SQL en Java. Una vez constatada la problemática subyacente, se planteó cuál sería un método válido para su exposición teórica.

2.1. Los componentes prácticos

Dado el carácter introductorio que se pretendía dar a los contenidos y la facilidad del modelo, se optó por presentar JDBC, la propuesta CLI de Java, en lugar de otras propuestas Java desarrolladas por los propios fabricantes de SGBD. Estas propuestas alternativas resultan ser mucho más específicas y, evidentemente, mucho menos estándar.

Como es conocido, JDBC es un API de Java que simplifica el modelo CLI (fuertemente ligado al interfaz ODBC) y lo extiende siguiendo los principios del lenguaje. Entre las simplificaciones que aporta cabe destacar:

- El proceso de conexión a la base de datos.
- La gestión del entorno de ejecución de sentencias
- La creación y uso de sentencias SQL parametrizadas.
- La gestión de cursores y la recuperación de resultados mediante éstos.
- La oferta de métodos que van más allá del propio SQL.

En el momento de la elaboración de estos contenidos, la especificación del JDBC se hallaba en su segunda versión e incluía ya, entre otras cosas, soporte para tipos SQL:1999 (el correspondiente al *draft SQL3*), movimiento libre de cursores, modificaciones *batch*, métodos específicos de manipulación de filas y tablas, etc.

Para la aplicación práctica de los contenidos, era necesario disponer de una implementación del API (implementación que se conoce como *driver*) dependiente del SGBD utilizado. Estos *drivers*, adicionalmente, tienen diferentes tipologías atendiendo al grado en que suscriben un modelo de tres capas.

En nuestra universidad, y hasta el momento, el SGBD suministrado para estas asignaturas es *Informix*. El mismo fabricante ofrece una

implementación de libre distribución del JDBC 2.0. En ese mismo sentido, la *web* de *Sun* ofrece un sistema de localización de *drivers* para un elevado número de SGBD comerciales [6].

El entorno Java utilizado fue el Java 2 (JDK 1.2 y posteriormente JDK 1.3) de *Sun*. Se eligió *Windows 9x* como plataforma de desarrollo por ser la habitual para el uso de nuestro campus virtual.

Desde el momento en que se pudo disponer de las herramientas prácticas se empezó a verificar sus capacidades. Como es habitual, fue necesario detectar aquellos puntos en los que la realidad de nuestro entorno (*Informix + Windows 9x + Java 2 + driver JDBC Informix type 4*) no seguía las especificaciones estándar. En esa misma línea, se verificó el comportamiento de otros productos SGBD, de diferentes tipos de *drivers* y de varias plataformas operativas.

Tras este trabajo se estaba en condiciones de determinar qué capacidades podían formar parte del eje de una descripción generalista del modelo.

2.2. La exposición teórica.

Entendiendo que inicialmente el material que se debía elaborar iba a ser complementario a otro suficientemente largo y complejo, se optó por una exposición basada en ejemplos y en la comparación con los modelos de SQL/CLI en C ya estudiados. No se pretendía, pues, rechazar la experiencia de la que se disponía utilizando el modelo en C, si no de que ésta, por comparación, subrayase las posibilidades de Java. Se trataba, pues, de reforzar una serie de aspectos básicos:

- La apuesta por el modelo CLI que evita la dependencia de base con el SGBD.
- La simplificación del modelo CLI de JDBC en relación al modelo en C tradicional.
- La comprensión en profundidad de los fundamentos de dicho modelo al liberarlo de las complejidades artificiales que inducen otros lenguajes.
- La justificación teórica de los elementos prácticos.
- La integración del modelo JDBC con la filosofía de Java.

De esta forma, la exposición empieza detallando los fundamentos teóricos que justifican el modelo JDBC, para seguir con la presentación de la estructura de un programa sencillo basado en dicho modelo:

- Localización y carga del driver.
- Conexión con el SGBD.
- Operaciones con el SGBD y tratamiento de excepciones.
- Desconexión del SGBD.

Para cada punto se especifica cómo realizarlo en el entorno de trabajo de referencia (variables de sistema, configuraciones de las herramientas, librerías requeridas por el lenguaje, etc.)

El programa presentado utiliza una sencilla interfaz texto para no mezclar las complejidades de un entorno gráfico con las necesidades JDBC. Este programa pasa a utilizarse entonces como una aplicación patrón y a partir de ella se construyen todos los ejemplos que introducen los diferentes aspectos detallados:

- Sentencias de modificación.
- Sentencias de consulta.
- Cursores con movimiento libre.
- Control de transacciones.
- Sentencias preparadas.
- Procedimientos almacenados.
- Modificaciones por lotes.

Se intenta también perfilar con claridad cuáles son las clases del API protagonistas de la estructura básica del patrón, así como sus métodos elementales y los que las interrelacionan (fig.1).

Finalmente se revisan las capacidades adicionales del modelo JDBC en aspectos relacionados con los tipos SQL:1999.

En definitiva, la idea subyacente es que la lectura motive al estudiante a probar los contenidos expuestos. De hecho, cada uno de los aspectos puede incorporarse a la aplicación patrón y probarse contra una pequeña base de datos de ejemplo.

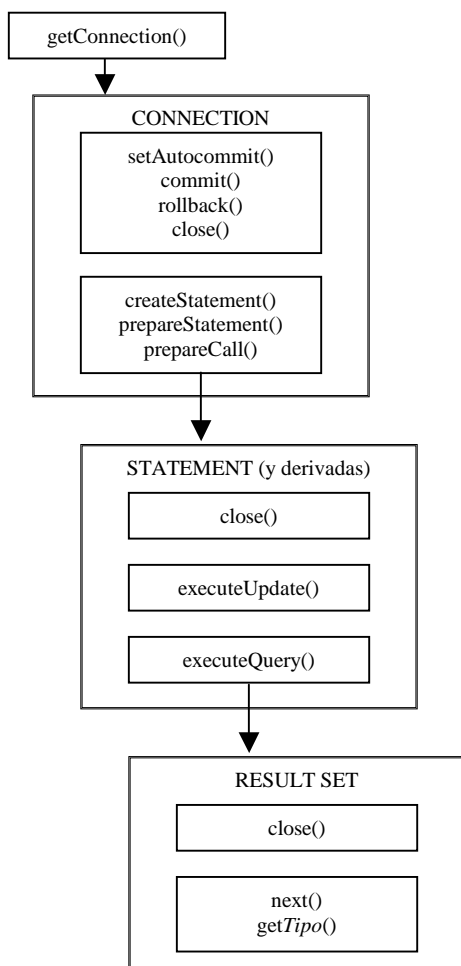


Fig.1: Clases y métodos elementales de JDBC

3. Implantación y aspectos metodológicos de la propuesta

En los tres cuatrimestres que lleva en funcionamiento esta propuesta, se ha ido ajustando atendiendo a requisitos académicos y, principalmente, a criterios metodológicos destinados a facilitar y asegurar el aprendizaje.

3.1. Redistribución de los contenidos

Como se ha dicho, *Bases de datos II* tiene un capítulo en el que se exponen las características

del SQL programado. Los contenidos sobre JDBC se han suministrado como un complemento a este material.

Esto ha provocado que la carga del tema se haya incrementado considerablemente. Por esta razón se está optando por presentar las técnicas de SQL hospedado y SQL/CLI en C desde una perspectiva meramente generalista e introductoria, poniendo en relieve su filosofía, sus condicionantes y su estudio comparativo. Con esta base adquirida, es el tema práctico desde la perspectiva JDBC el que adquiere la máxima relevancia.

3.2. Obligatoriedad de la práctica

En un principio, la práctica no era obligatoria. Sin embargo, sí se hizo una amplia difusión entre los estudiantes de la importancia de afrontarla. Esto, unido con que tenía un peso del 20% de la nota final, produjo que la gran mayoría de los estudiantes la presentaran.

Como consecuencia de esto y de que se había confirmado como una herramienta excelente para consolidar paulatinamente los objetivos de aprendizaje, se decidió convertirla en obligatoria. Para conseguir además que realmente fuera un trabajo continuo en el tiempo, se introdujo la entrega en dos partes, inicialmente con la primera entrega voluntaria (acumulándose toda la entrega en la segunda si no se realizaba la primera) y posteriormente con las dos entregas obligatorias en el tiempo.

También parecía claro que lo importante no era escatimar el tiempo dado a los estudiantes entre la entrega del enunciado y la recogida de las soluciones y, así, se ha llegado a un intervalo aproximado de ocho semanas, en unos cursos que tienen una duración lectiva de alrededor de trece semanas.

3.3. Detalle del caso objeto de la práctica

También el contenido del caso de la práctica ha pasado de ser muy guiado (dando el diseño de la base de datos) a dejarse prácticamente abierto (mediante la exposición descriptiva de un caso), y planteando una serie de preguntas-objetivo que han de constituir el núcleo de la respuesta del estudiante.

Con esta apertura del caso hemos reforzado el objetivo de desarrollar las capacidades del estudiante en cuanto a:

- Diseño conceptual de la base de datos, aprendido en *Bases de Datos I*. Este aprendizaje previo es objeto de uso en muchas de las asignaturas de la carrera basadas en casos o proyectos, sin mencionar, por supuesto, la relevancia que tiene en la práctica profesional.
- Decisión sobre el uso adecuado de los componentes lógicos (de control y de datos) presentados. El estudiante ha de ser capaz, por ejemplo, de ver cuáles son las vías para implementar una restricción y elegir la que le parezca más adecuada.

3.4. Trabajo en grupo de la práctica

Al principio, el trabajo de la práctica se planteó individualmente, cosa que se justificaba por la voluntariedad de la misma. Eliminado este condicionante se optó por permitir que aquellos estudiantes que así lo quisieran pudieran entregar la solución en grupos de dos personas.

No se ha establecido éste como un criterio obligatorio por dos motivos básicos:

- Los estudiantes de nuestra universidad ya tienen un elevado número de asignaturas donde el trabajo en grupo es obligatorio con lo que se cubre el objetivo de dar prioridad a este aspecto.
- Es interesante que las tareas de diseño de la base de datos de un caso complejo se puedan pensar también individualmente. Si en la práctica profesional es habitual trabajar en grupo, también lo es que, en situaciones más o menos comunes, haya que tomar decisiones de diseño individualmente (y no sólo sobre aspectos sencillos).

3.5. Soporte Java

En cuanto al soporte que se da al estudiante para enfrentarse al lenguaje, se queda básicamente en el programa patrón y en las indicaciones de configuración e instalación dadas en la exposición teórica de los contenidos.

Se explica también la importancia de la documentación y los tutoriales sobre el API JDBC suministrado en el JDK [7,8]. Además se les recomienda tener a mano la documentación del propio fabricante del *driver* para *Informix* [9].

Una vez más, se prima que el estudiante indague por su cuenta, como pasa en la realidad, las capacidades de un nuevo contenido en el que se introduce. Evidentemente, no se está pidiendo un nivel elevado sobre el uso del lenguaje y, en consecuencia, para cumplir los mínimos no es necesario un gran trabajo de investigación.

Se entiende también que, dado que cada vez hay menos estudiantes de planes antiguos, estos estudiantes ya están introducidos en los elementos básicos de la orientación a objetos. Además, llegados a esta altura de sus estudios (donde, en buena lógica, deben haber cursado asignaturas con un fuerte contenido en orientación a objetos como *Fundamentos de Programación II*, *Ingeniería del Software I* o *Técnicas de Desarrollo de Software*) es de suponer que los estudiantes han de tener recursos suficientes para seguir avanzando.

4. Resultados

En general se ha observado una acogida muy buena de los contenidos propuestos por parte de los estudiantes. Esencialmente tenemos a nuestro favor varios hechos:

- Las asignaturas sobre bases de datos tienen habitualmente una consideración positiva por parte del estudiante.
- La programación en SQL es percibida como un componente fundamental para el trabajo con bases de datos.
- Java abre las puertas a otros aspectos muy bien apreciados: la elaboración no sólo de aplicaciones sino de *applets*, el mundo cliente/servidor o de Internet y el acceso remoto o distribuido.

Por otro lado, en el primer cuatrimestre se detectaron algunos problemas, principalmente técnicos, derivados del uso del *driver* y del JDK sobre plataformas que no eran las recomendadas o que no estaban en buenas condiciones de mantenimiento. Estas situaciones se han repetido

cada vez menos ya que se han ido dando indicaciones muy claras al respecto. Adicionalmente, se han ido realizando tareas periódicas de verificación con los nuevos entornos que han ido apareciendo.

4.1. Consecución de objetivos

En cuanto al objetivo básico referido a la programación en SQL desde Java, creemos que se ha logrado ampliamente. A esto contribuye la complejidad de los casos prácticos propuestos que obligan al estudiante a plantearse soluciones para acceder intensiva y extensivamente a la base de datos, imaginando estrategias para una programación eficiente. Así, acaba interrogándose sobre cómo mejorar aspectos concretos, lo que le lleva a profundizar en las posibilidades y limitaciones del sistema.

Del mismo modo, el objetivo referido al diseño de base de datos y al uso de componentes lógicos (de datos y de control) se puede considerar alcanzado, al observar, una vez más, la complejidad de los casos desarrollados. El carácter abierto del caso deja espacio a la profundización sobre todo un conjunto de situaciones diferentes. En la mayoría de los casos, el análisis de estas situaciones realizado por el estudiante ha sido muy completo y las soluciones aportadas han demostrado que ha entendido cómo lograrlas, tanto desde la perspectiva de diseño como desde la de implementación con los componentes lógicos.

Por lo que respecta al objetivo de la introducción al mundo Java, hemos constatado una serie de comportamientos distintos aunque previsibles:

- El estudiante con conocimiento de orientación a objetos y Java ha dedicado tiempo a mejorar aspectos del código de la programación así como del interfaz, aunque ello no era objetivo del curso ni mejoraba la puntuación.
- El estudiante con conocimiento de orientación a objetos pero sin experiencia en Java, no ha tenido problemas en cumplir lo exigido y, en la mayoría de casos, no ha percibido el lenguaje como un problema.
- El estudiante que, por formación académica antigua, sólo está introducido mínimamente a

la orientación a objetos y que no tiene conocimiento de Java, ha sabido utilizar el programa patrón suministrado y llegar con él a todas las respuestas planteadas. Es cierto que, en algunos casos, no se ha planteado entender completamente cómo se justificaba la propuesta del patrón, con lo que se puede decir que ha visto el aspecto de un programa mínimo en Java, pero no que será capaz de elaborar uno desde cero. Sea como sea, con esta situación de partida, este resultado se puede considerar un hito relevante.

En general, pues, Java no ha resultado ser un elemento problemático, sino más bien al contrario. El interés por profundizar en él ha beneficiado la consecución de los otros objetivos de aprendizaje. Es decir, lo que en principio podía parecer más conflictivo se ha acabado convirtiendo en un motor para el seguimiento de los contenidos.

4.2. Otros efectos constatados

Como se ha ido demostrando en las diferentes asignaturas y carreras que imparte la UOC, en un entorno de aprendizaje a distancia y virtual la constancia en el estudio y en el mantenimiento de un ritmo regular de dedicación al mismo tienen una importancia si cabe mayor que en los entornos presenciales.

Así, el hecho de que la práctica sobre JDBC se plantee periodificada con dos puntos de entrega y que se disponga de un elevado número de semanas para resolverla ha facilitado el aprendizaje paulatino. Cuando no existía la obligación de una entrega intermedia, se produjeron bastantes casos de abandono ya que el estudiante dejaba acumular todo el trabajo al final, muy cerca de la fecha de entrega definitiva.

Un aspecto que ha aparecido espontáneamente ha sido el trabajo colaborativo entre los estudiantes. En nuestra universidad, la existencia de un espacio común de intercambio (el foro) ha facilitado que expusieran en público sus problemas para solicitar ayuda de los compañeros. La respuesta del grupo, en general, ha sido estimulante: en muchas ocasiones no se esperaban a que el profesor diera alguna indicación al problema sino que se ofrecían soluciones y experiencias particulares. Curiosamente, los problemas expuestos no han sido tanto de soluciones de diseño (los contenidos originales y

propios de cada estudiante), sino de temas más prácticos sobre la implementación con el lenguaje o con el *driver* y la configuración del entorno. La generación de este comportamiento colaborativo nos parece especialmente importante porque denota una capacidad para compartir experiencias que, de mantenerse como una constante en la actividad del estudiante, constituirá un valor muypreciado para el trabajo en grupo en un entorno profesional.

Las principales críticas de los estudiantes han venido por aspectos menos relacionados con nuestros objetivos: el SGBD utilizado (que resulta una herramienta didáctica suficiente pero que no representa un producto con mucha implantación actual), así como la dificultad para encontrar información adicional, incluso de las propias fuentes. Estas consideraciones se irán teniendo en cuenta progresivamente, como vía natural para incrementar la calidad de la propuesta.

4.3. Datos comparativos

Como colofón, hemos querido contrastar nuestra percepción sobre la aceptación de la propuesta entre los estudiantes, así como su efecto positivo en el rendimiento académico.

En el primer aspecto sólo podemos remitirnos a los comentarios públicos y voluntarios de los estudiantes, dado que no se ha realizado una encuesta formal entre ellos. De estos comentarios, y con las reservas que impone su origen, hemos creído detectar una elevada satisfacción con el aprendizaje adquirido a pesar de su complejidad y del esfuerzo que requiere. Incluso hemos

constatado que algunos de los matriculados más recientes lo han hecho por recomendación de antiguos estudiantes.

En cuanto al rendimiento académico hemos resumido los datos sobre la nota media de cada cuatrimestre, así como las diferencias objetivas entre ellos, resumen que se concreta en la siguiente tabla:

	<i>Media</i>
<i>Semestre 1:</i>	
▪ Voluntaria	
▪ Una entrega	
▪ Caso guiado.	
▪ Estudiantes Ing. Técnicas	6.93
<i>Semestre 2:</i>	
▪ Obligatoria	
▪ Dos entregas (primera voluntaria)	
▪ Caso abierto	
▪ Estudiantes Ing. Técnic. y Superior	7.48
<i>Semestre 3:</i>	
▪ Obligatoria	
▪ Dos entregas	
▪ Caso abierto	
▪ Estudiantes Ing. Técnic. y Superior	7.98

A continuación, para intentar detectar si el incremento aparente en dicha nota puede considerarse significativo, hemos sometido los datos a un test de ANOVA (figura 2).

El resultado de este test indica que, en efecto, la nota media obtenida en el tercer semestre es significativamente superior (a un nivel de significación del 95%) a la nota media del primer semestre (los intervalos correspondientes son

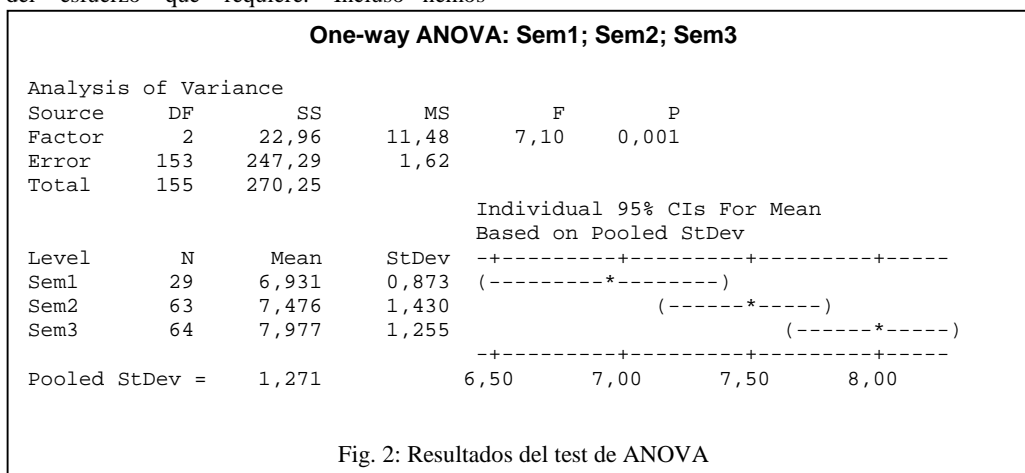


Fig. 2: Resultados del test de ANOVA

disjuntos), lo que corrobora nuestra impresión de mejora.

5. Perspectivas de futuro

Para terminar queremos esbozar unas líneas sobre cómo podría seguir avanzando esta propuesta.

En primer lugar, se constata que las herramientas de desarrollo rápido de aplicaciones (RAD) tienen un uso relativamente poco extendido entre los estudiantes y los ejercicios que se les exigen en diferentes asignaturas. Esto contrasta con la elevada implantación que tienen en los entornos profesionales.

Por este motivo, podría considerarse en un futuro, y dado que los estudiantes cada vez llegan con un conocimiento más profundo en Java y en orientación a objetos, realizar la práctica en un entorno gráfico y utilizando alguna de estas herramientas para Java. Las necesidades de los casos que se trabajan lo hacen perfectamente aplicable y justificable. De esta forma, a su vez, al estudiante podrá introducirse en el código Java que generan este tipo de aplicaciones. En este código conviven objetos generados automáticamente y que representan clases sólo necesarias para la implementación del GUI, con los objetos que representan las clases que responden al problema planteado. Los modelos que implementan estas clases específicas para el interfaz gráfico, constituyen un conjunto muy importante sobre el que el estudiante debería tener alguna referencia a lo largo de sus estudios.

En segundo y último lugar, el acceso a bases de datos desde programación en Java es un aspecto que podría exportarse e incluirse extendidamente en otro tipo de asignaturas

orientadas a la resolución de proyectos, a sistemas distribuidos o a aplicaciones Internet entre otras. Es decir, se podría llegar a reorientar el planteamiento y el uso de este material y que el modelo JDBC se explicara en un contexto de aprendizaje más general, no restringido sólo al de las bases de datos.

Referencias

- [1] Sistac Planas, Jaume (Coord) y otros. *Bases de dades I*. Ed. UOC. Barcelona, 1999.
- [2] Sistac Planas, Jaume (Coord) y otros. *Bases de dades II*. Ed. UOC. Barcelona, 1999.
- [3] Rodríguez González, M.Elena. *Programació amb SQL* en: Sistac Planas, Jaume (Coord) y otros. *Bases de dades II*. Ed. UOC. Barcelona, 1999.
- [4] Mallafré i Porta, Francesc Xavier. *Components lògics d'una base de dades* en: Sistac Planas, Jaume (Coord) y otros. *Bases de dades II*. Ed. UOC. Barcelona, 1999.
- [5] Marco Simó, Josep Maria. *SQL i Java*. UOC. Material de la asignatura *Bases de dades II*, 2001.
- [6] <http://industry.java.sun.com/products/jdbc/drivers>
- [7] Sun Microsystems, Inc. (1996, 1997) *JDBC Guide: Getting Started* Documentación del JDK 1.2.
- [8] Fisher, (M.) Sun Microsystems, Inc. (1997) *JDBC Database Access: Trail*. Documentación del JDK 1.2.
- [9] Informix Software, Inc. (1997) *Informix JDBC Driver: Programmers Guide*. Documentación del driver JDBC ver. 2.11 .