

# **Laboratorio de Ingeniería del Software**

## La ingeniería del software orientada a objetos y cliente/servidor llevada a la práctica

**Vicente Pelechano, M<sup>a</sup> Carmen Penadés**  
Departament de Sistemes Informàtics i Computació  
Universitat Politècnica de València

Camí de Vera, S/N, 46071 València  
Tel: 3877350; Fax: 3877357  
(e-mail: pele | mpenades}@dsic.upv.es)

### **Resumen**

*En este documento se presenta la asignatura Laboratorio de Ingeniería del Software que se imparte en el tercer curso de la Escuela Universitaria de Informática en la Universidad Politécnica de Valencia. Esta asignatura es obligatoria en la titulación de ITIG y optativa en la de ITIS, posee un enfoque eminentemente práctico y basado en las últimas tecnologías, enseñando a los alumnos a trabajar de forma ingenieril y disciplinada en el desarrollo de aplicaciones. La asignatura pretende acercar a los alumnos a las necesidades actuales del mercado de trabajo, enseñándoles a utilizar herramientas de desarrollo visuales, adoptando la orientación a objetos como metodología de desarrollo, aplicar la tecnología cliente/servidor, desarrollar y reutilizar componentes software, y finalmente, introducir Internet como un nuevo ambiente de trabajo para el desarrollador del futuro. El documento presenta las razones que nos llevan a proponer este enfoque, cómo se ha implantado la asignatura (aspectos teóricos y prácticos) y al final se realiza un análisis crítico de las experiencias obtenidas con los alumnos.*

### **1. Introducción**

La aparición de nuevas tecnologías, el auge de la orientación a objetos y del desarrollo cliente/servidor han introducido nuevos conceptos y nuevas necesidades en el campo de la ingeniería del software. En la unidad docente Desarrollo de Software del Departamento de Sistemas Informáticos y Computación de la Universidad Politécnica de Valencia, no hemos permanecido ajenos a estos cambios tecnológicos y hemos preparado la asignatura Laboratorio de Ingeniería

del Software (en adelante, LIS) que se imparte en la ingeniería técnica en Informática y que pretende acercar desde un punto de vista ingenieril al alumno a la nueva ingeniería del software orientada a objetos y cliente/servidor, término utilizado ya en los últimos libros de ingeniería del software [5] que pretende introducir los nuevos conceptos en el marco global de la ingeniería del software.

Presentamos la asignatura denominada LIS que se imparte en el tercer curso de la Escuela Universitaria de Informática en la Universidad Politécnica de Valencia. Esta asignatura posee un enfoque eminentemente práctico y basado en las últimas tecnologías, enseñando a los alumnos a trabajar de forma ingenieril y disciplinada en el desarrollo de aplicaciones. El documento se estructura de la siguiente manera: en el apartado 2 presentaremos las razones que nos han llevado a proponer esta asignatura con el enfoque que le hemos dado, explicando qué necesidades docentes se pretenden cubrir, en los apartados 3 y 4 se presentará cómo se ha implantado la asignatura, presentando cómo se ha organizado la teoría y las prácticas, respectivamente. En el apartado 5 se realiza un análisis crítico de las experiencias obtenidas con los alumnos, para terminar concluiremos con las posibles mejoras que se le pueden efectuar a la asignatura teniendo en cuenta la realimentación que nos proporcionan los alumnos y los próximos avances tecnológicos.

### **2. La necesidad de un Laboratorio de Ingeniería del Software**

La rápida y continua evolución tecnológica, junto con la aparición de las herramientas de desarrollo rápido (RAD), la tecnología orientada a

objetos, el modelo de desarrollo cliente/servidor, la programación basada en componentes e internet ha provocado ciertas necesidades en la formación de un ingeniero técnico en informática.

Los alumnos consideran una necesidad el aprender estas tecnologías emergentes, y ven su aprendizaje como una vía de acceso al mercado de trabajo. Nosotros creemos que es necesaria una docencia que acerque al alumno a las nuevas tecnologías, pero, como profesores en ingeniería del software que somos debemos aportarles una visión seria y disciplinada del desarrollo del software utilizando las nuevas tecnologías.

Necesitamos proporcionarles un enfoque riguroso y objetivo, no debiendo caer en lo comercial y en la subjetividad, porque todo este tipo de herramientas surgen como verdaderas revoluciones que solucionan todos los problemas del desarrollo: permiten construir aplicaciones más rápidamente, parece que son fáciles de usar, mejoran la calidad de vida del usuario y del propio desarrollador que utiliza la tecnología, en definitiva, prometen mucho pero la realidad es bien diferente: las aplicaciones complejas se vuelven más complejas, este tipo de herramientas cambian el proceso de desarrollo y no existen metodologías de desarrollo que soporten el proceso de producción.

Ante toda esta avalancha tecnológica, parece obvio que la universidad y en particular los responsables en la docencia de ingeniería del software deberían aportar conocimientos, orden, disciplina y un enfoque ingenieril a estas nuevas tendencias. La asignatura Laboratorio de Ingeniería del Software se pensó como una continuación, más práctica y aplicada, de la asignatura Ingeniería del Software que intentara dar soluciones y educar a los futuros técnicos del software intentando proporcionarles métodos, técnicas y sobre todo a saber desenvolverse en el mercado de trabajo. Es muy importante destacar que uno de los factores que nos decidieron a que la asignatura tomara este enfoque eminentemente práctico fue que la mayoría de los alumnos que iban a llegar a nuestra asignatura, no iban a continuar sus estudios en el segundo ciclo, lo que implicaba que saldrían al mercado de trabajo sin conocimientos teóricos y prácticos en las nuevas tecnologías.

Por lo tanto, el objetivo general de la asignatura es introducir al alumno (de forma básica y práctica) en las nuevas tecnologías software utilizadas para la construcción de aplicaciones a nivel industrial. Estudiando de la forma más genérica posible cada una las propuestas tecnológicas más relevantes e influyentes en la actualidad, enseñando métodos y técnicas para conseguir buenos desarrollos.

A nivel docente se pretende que el alumno adquiera ciertos hábitos de trabajo y domine diversas disciplinas que le permitan desenvolverse satisfactoriamente dentro de estas nuevas tecnologías, intentando que:

- no pierda la globalidad del proyecto al centrarse en aspectos tecnológicos muy concretos.
- piense que las nuevas tecnologías y las herramientas que las soportan no son el fin, sino el medio para conseguir la aplicación
- planifique correctamente la puesta en marcha de la aplicación
- sepa elegir de forma objetiva las herramientas de trabajo y el lenguaje de desarrollo
- sepa diseñar la aplicación utilizando metodologías OO
- sepa programar para reutilizar, y que domine la técnica de la reutilización de componentes
- sepa diseñar interfaces de usuario
- construya buenos diseños y que sepa documentar el software que desarrolla

### 3. Prerrequisitos y fundamentos teóricos

En este apartado vamos a explicar el carácter de la asignatura, en qué titulaciones se imparte, cuáles son los conocimientos teóricos y prácticos que debe conocer un alumno para aprovechar adecuadamente la asignatura, y para finalizar se explican los contenidos teóricos que se imparten durante el curso.

La docencia de la asignatura la imparte el Departamento de Sistemas Informáticos y Computación en los títulos de Ingeniero Técnico en Informática de Gestión (ITIG) e Ingeniero Técnico en Informática de Sistemas (ITIS), siendo troncal en ITIG y optativa en ITIS. La asignatura

se imparte en el segundo semestre del tercer curso de la Escuela Universitaria de Informática y consta de 3 créditos teóricos y 3 prácticos.

Como ya se ha comentado anteriormente, esta asignatura está concebida como la continuación aplicada de la asignatura de Ingeniería del Software, por lo que se presupone que para el buen aprovechamiento de los contenidos de la misma, el alumno ha cursado la asignatura que le precede.

Los conocimientos básicos que se requieren son:

- conocer los ciclos de vida de desarrollo del software utilizados en la actualidad, en especial el ciclo de vida en espiral
- saber planificar proyectos y desenvolverse dentro de un plan de proyecto
- conocer metodologías de análisis y diseño orientado a objetos en particular OMT y el diseño dirigido por responsabilidades
- tener un buen nivel en la programación orientada a objetos y en especial que domine la filosofía general que el enfoque orientado a objetos lleva consigo

Siguiendo el enfoque disciplinar que se ha presentado en el punto anterior, el temario teórico de la asignatura se puede resumir en los siguientes puntos o áreas temáticas que más adelante explicaremos detalladamente:

- Presentación de los conceptos básicos de la programación en entornos visuales.
- Introducción de conceptos avanzados de la programación orientada a objetos: construcción y utilización de componentes de software, reutilización...
- Introducción del modelo Cliente / Servidor y la repercusión del mismo en la construcción de aplicaciones.
- Introducción a Internet cómo un nuevo ambiente de trabajo para el desarrollador actual y Java como su herramienta de desarrollo.

Cada una de las anteriores áreas temáticas esta compuesta a su vez de diversos temas que presentamos a continuación explicando el enfoque que se le da a cada uno de ellos:

### Área temática 1: Introducción al desarrollo de aplicaciones visuales.

#### 1. El desarrollo rápido de aplicaciones y la Ingeniería del Software

En este tema se presenta una visión global del desarrollo rápido de aplicaciones desde la perspectiva de la ingeniería del software, explicando las pautas a seguir para utilizar correctamente el tipo de herramientas que soportan este tipo de desarrollo.

#### 2. Introducción a los entornos de programación visuales orientados a objetos.

En este tema se explican las características comunes de los entornos de desarrollo visuales y qué facilidades ofrecen. Se presenta el concepto teórico de programación visual haciendo ver al alumno que no se corresponde con lo que predica el mundo de la industria. Y para terminar se realiza un recorrido a través de gran cantidad de entornos visuales de desarrollo, presentando una taxonomía de este tipo de entornos según sean más o menos visuales y según soporten mejor o peor el desarrollo orientado a objetos.

#### 3. El entorno de desarrollo integrado *BORLAND DELPHI*.

- Introducción a *Borland Delphi*.
- Construcción de Proyectos con el IDE de Delphi.
- El diseño de la Interfaz de Usuario.
- Construcción de aplicaciones (SDI y MDI)
- Conceptos avanzados de Object Pascal.
- Gestión de excepciones.

El entorno de desarrollo elegido para explicar los conceptos esenciales de la asignatura es *Borland Delphi*, gracias a su facilidad de uso, su buen soporte a la programación orientada a objetos y porque es mucho más didáctico y coherente a nivel conceptual y de manejo que otro tipo de entornos de desarrollo.

En este tema se introduce al alumno en el concepto de proyecto, se explica cómo construir interfaces gráficas de usuario siguiendo algunas normas de diseño, se les explica el tipo de aplicaciones que pueden construir con los entornos visuales (SDI y

MDI), y para finalizar algunos conceptos avanzados de la orientación a objetos que proporciona Object Pascal (no se les introduce a la orientación a objetos porque se supone que ya la conocen) incluyendo el tratamiento de las excepciones que poseen algunos lenguajes OO como Object Pascal.

**Área temática 2:** Desarrollo de aplicaciones de bases de datos y la nueva era cliente/servidor.

1. Desarrollo de aplicaciones de bases de datos en Delphi.

- Introducción al *Borland Database Engine*. BDE vs ODBC
- Bases de datos locales y SQL.
- El control de la concurrencia y las transacciones.

Se introduce al alumno en el desarrollo de aplicaciones de bases de datos en entornos visuales, explicando de forma genérica la serie de componentes que proporcionan las herramientas de desarrollo visuales para encapsular el acceso a las bases de datos. Se presentan los distintos patrones de desarrollo, por ejemplo, el maestro/detalle típico de las aplicaciones de gestión.

En este mismo tema se hace una pequeña revisión del SQL y se muestra cómo utilizar diversas técnicas de acceso a bases de datos (ODBC, Gateways específicos, BDE), y se dan algunas pautas para la utilización de transacciones en los desarrollos cliente/servidor.

2. Introducción al modelo Cliente/Servidor.

Este tema recoge los conceptos básicos sobre el modelo Cliente/Servidor: tipos de arquitecturas, el modelo de cinco niveles de Gartner-Group, se presentan los costes, riesgos y beneficios que conlleva el adoptar esta nueva forma de desarrollo e implantación del software.

3. Construcción de aplicaciones distribuidas basadas en el modelo de servicios. El modelo de tres niveles.

- Construcción de aplicaciones multinivel con Delphi.

En este tema se explica como llevar a la práctica utilizando Delphi, los conceptos que se han aprendido en el tema de desarrollo de bases de datos y en el de Cliente/Servidor, proponiendo la construcción de una aplicación multinivel y cliente/servidor.

**Área temática 3:** Conceptos Avanzados para el Desarrollo de Aplicaciones. Componentes y reutilización

1. La biblioteca de componentes VCL

2. Desarrollo de componentes VCL y reutilización en Delphi.
- Un ejemplo práctico.

En esta área temática se pretende introducir al alumno en el mundo del desarrollo de componentes y en las técnicas de reutilización. El desarrollo de componentes es un tema complejo, aunque *Borland Delphi* ofrece un marco para el desarrollo de componentes muy sencillo y didáctico que nos permite explicar de forma genérica como sería el desarrollo de componentes en este tipo de entorno de desarrollo, por lo que inicialmente se presenta la VCL (la biblioteca de componentes visuales de Delphi) para conocer la jerarquía de componentes de las que dispone el desarrollador y luego se explica como desarrollar componentes VCL mediante un ejemplo (el tipo de componentes propio de Delphi)

3. Desarrollo con OLE/ActiveX

- ¿Qué es la tecnología OLE/ActiveX?
- COM un nuevo estándar de componentes.
- Automatización OLE y controles ActiveX

Una vez el alumno domina el tema de las componentes, se presenta el estándar COM y se explica como desarrollar componentes OLE/ActiveX para Windows. Este quizás es el tema más complejo, aunque no se entra en detalle porque se intenta que el alumno conozca la tecnología para poder aplicarla en un futuro.

**Área temática 4:** Internet y el desarrollo de aplicaciones para la WEB

1. Introducción a Internet y sus repercusiones en el desarrollo del software.

En este tema se presenta Internet visto desde el punto de vista del desarrollador, explicando como va a cambiar la forma de desarrollar y vender software, con la intención de preparar al alumno para los nuevos retos que Internet nos va a plantear.

## 2. El lenguaje Java. Pasado, presente y futuro

En este tema se explica el desarrollo en Internet a través del lenguaje Java, explicando las expectativas de futuro que nos puede ofrecer.

## 4. La ingeniería del software en la práctica

En este apartado se da una descripción de la parte práctica de la asignatura, una vez expuestos los objetivos teóricos de la misma y cómo están estructurados en áreas temáticas. En primer lugar, antes de enumerar las prácticas a realizar por los alumnos, resaltar el carácter eminentemente práctico de la asignatura, que ya viene reflejado en su propio título: Laboratorio de Ingeniería del Software. Se intenta complementar, de forma práctica, la formación teórica de los alumnos en las nuevas tecnologías, muchas de ellas predominantes en el mercado actual.

El objetivo esencial de las prácticas lo podríamos resumir de la siguiente manera: poner en práctica los conocimientos adquiridos en Ingeniería del Software (programación orientada a objetos y el análisis y diseño orientado a objetos) desde un punto de vista más aplicado y abordar la programación en entornos visuales, la utilización y reutilización de componentes, la creación de los mismos, y la implementación de aplicaciones que accedan a bases de datos locales y remotas, aplicando algunos conceptos del modelo Cliente/Servidor.

Las prácticas de la asignatura se realizan en un laboratorio docente, que dispone de 20 puestos de trabajo, dotados de un PC compatible, donde los alumnos organizados en grupos de dos personas, desarrollarán las aplicaciones propuestas en las diferentes prácticas, utilizando como herramienta de desarrollo *Borland Delphi versión 3.0*.

La elección de *Borland Delphi* como herramienta de desarrollo no ha sido fortuita, sino que se eligió pensando que era la más apropiada para desarrollar las prácticas de la asignatura, de acuerdo con los objetivos fijados y los

conocimientos previos de los alumnos. En este sentido, *Borland Delphi* tiene como lenguaje de programación Object Pascal, que es un lenguaje de programación orientado a objetos, lo cual permite poner en práctica los conceptos básicos de la orientación a objetos (noción de clase, objeto, atributo, método, herencia...), cuyo aprendizaje queda dentro de los objetivos de LIS. Por otra parte, en cuanto al diseño de Interfaces Gráficas de Usuario (IGU), *Borland Delphi* es similar al resto de herramientas disponibles en el mercado, donde la IGU se construye a partir del conjunto de controles que ofrece el entorno, e igual ocurre con el tema de acceso a Bases de Datos.

La planificación de las prácticas a realizar durante el curso es la siguiente:

### **Práctica 1.** Introducción a la programación en el IDE de *Borland Delphi*.

Tiene como objetivo familiarizarnos con el entorno integrado de desarrollo que nos proporciona Delphi. Construyendo una sencilla aplicación de ventanas, para que el alumno empiece a utilizar los componentes visuales que proporciona Delphi y conozca la filosofía general del desarrollo visual.

### **Práctica 2.** Desarrollo de aplicaciones en *Borland Delphi*.

Esta práctica engloba un conjunto de prácticas a realizar que consistirán en el desarrollo completo de varios tipos de aplicaciones.

*1. Construcción de un editor de textos MDI.* El objetivo de esta práctica es aprender a crear e integrar aplicaciones MDI (Interfaz de múltiple documento), frente a las aplicaciones SDI (Interfaz de simple documento), a la vez que se abordan conceptos fundamentales y técnicas de presentación y manipulación de texto. Con este objetivo, la práctica presenta los pasos necesarios para crear un sencillo editor de texto MDI, que además de manipular el texto, utiliza los cuadros de diálogo estándar de Windows para abrir y salvar ficheros, así como para cambiar la fuente de los ficheros de texto que crearemos con la aplicación. Además se aborda la creación y fusión de menús, el final controlado de la aplicación y la creación de las típicas barras de herramientas y barra de estado.

*2. Construcción de una aplicación de bases de datos.* Esta práctica tiene por objetivo desarrollar

un sistema de información que nos permita aplicar la mayoría de las técnicas necesarias para el desarrollo de aplicaciones de gestión utilizando bases de datos relacionales y entornos de desarrollo visuales. Con ello se pretende que el alumno sea capaz de abordar las fases esenciales del desarrollo de una aplicación de gestión y que el esfuerzo invertido le sea útil a todos los niveles (docente, práctico y personal), por lo que se proponen una serie de temas a elegir que podríamos englobar bajo el título genérico de "Sistema de Información Personal".

El desarrollo de la práctica, pretende acercar a los alumnos, lo más posible al desarrollo real de aplicaciones de este tipo, de acuerdo con las diferentes tecnologías a aplicar en cada fase del desarrollo. Por ello, el desarrollo de la práctica, no se centra únicamente en la implementación, sino que la realización de la misma, conlleva los siguientes pasos: realizar un análisis orientado a objetos del problema (construyendo el modelo de objetos de la aplicación mediante la notación OMT), la obtención de un diseño relacional a partir del modelo de objetos, la construcción de una arquitectura de clases para la implementación de la aplicación, el diseño de la interfaz de usuario para la actualización, consulta y búsqueda de información, la generación automática de informes, y la adecuada documentación del desarrollo seguido y la elaboración de un manual de usuario.

### **Práctica 3. Creación de componentes en *Borland Delphi*.**

En esta práctica, se aplicarán los conceptos vistos en teoría para crear nuevas componentes software y cómo añadirlas a la biblioteca de componentes que proporciona *Borland Delphi*.

## **5. Resultados de este enfoque y la realimentación de los alumnos.**

Desde nuestra perspectiva, el enfoque dado a LIS pensamos que es el que debería tener una asignatura que se imparte en el último curso de la titulación de ingeniero técnico en informática, y que viene a completar la formación del alumno en el campo de la Ingeniería del Software, como hemos expuesto anteriormente.

Desde el punto de vista de los alumnos, la asignatura ha tenido una muy buena aceptación, no sólo reflejada en el interés mostrado por los alumnos, sino también en el aumento progresivo de matrícula. La asignatura empezó a impartirse en el curso 95/96 con 223 alumnos matriculados, en el siguiente curso se incrementó a 256 y en el curso 97/98 ha pasado a ser de 267, lo que ha motivado un aumento del número de grupos de teoría y de prácticas impartidos. Este aumento se debe principalmente al carácter optativo de la asignatura para la titulación de ITIS, que ha provocado un aumento de peticiones de matriculación en la misma.

Aunque la situación actual de la asignatura podemos decir que es buena o incluso muy buena, ello no quiere decir que no existan carencias, que año a año y con la propia aportación de los alumnos, mediante encuestas, resultados de exámenes, interacción directa en clase, en prácticas o en tutorías, intentamos mejorar.

Algunas de estas carencias o puntos más débiles se repiten curso tras curso, y en ellos es donde insistimos especialmente. Por ejemplo, muchas veces al desarrollar una aplicación, los alumnos se pierden en aspectos puramente tecnológicos, que les impide avanzar, llegando incluso a perder la noción de globalidad del proyecto a realizar. Otro tipo de problema con el que se encuentran es el diseño de la interfaz de usuario, los alumnos se ven inexpertos en este campo y muchas veces les resulta difícil construir interfaces verdaderamente usables, incluso conociendo bien cuál debe ser la funcionalidad de la aplicación, son incapaces de construir diseños que la ofrezcan de forma sencilla, nosotros les recomendamos que utilicen el sentido común y que muestren a gente inexperta (informáticamente hablando) para que les proporcionen ideas y les critiquen su diseño. Otro punto conflictivo, es llegar a comprender y aplicar correctamente el modelado y diseño orientado a objetos como metodología de desarrollo, inicialmente se defienden bien en el análisis pero después les es difícil aplicarlos los conceptos aprendidos para construir un buen diseño e implementar la aplicación utilizando al máximo las ventajas de la orientación a objetos. Por último otro factor que induce a conflictos y que es necesario promocionar en el mundo del desarrollo del software es el trabajo en equipo ya que es fundamental que el

alumno sepa desenvolverse trabajando conjuntamente con compañeros.

## 6. Mejoras futuras

La experiencia adquirida durante tres años de docencia y la realimentación que los alumnos aportan cada año no ha hecho más que reforzar la tesis de que actualmente es necesario un enfoque como el que la asignatura presenta, quizás el problema más grave sea el poco tiempo disponible para dar un temario tan ambicioso, pero el interés de los alumnos compensa el problema del tiempo, puesto que son ellos mismos los que solicitan información sobre aquellos temas que no se llegan a explicar con todo detalle en clase, llegando incluso a pedir seminarios y horas de clase extra.

El trabajo futuro a realizar en la asignatura será pulir los defectos o pequeños problemas que vayan saliendo, adaptando el temario a la evolución tecnológica, lo cual supone actualizar ciertos temas y eliminar aquellos que queden obsoletos para introducir nuevos métodos, técnicas o herramientas. Muchos dirán que mantener viva una asignatura cada año puede ser muy costoso, pero dados los resultados obtenidos es un esfuerzo que merece la pena.

## Referencias

- [1] Argawal S., Jensen R., and Keller A. M. Architecting object applications for high performance with relational databases. In OOPSLA Workshop on Object Database Behaviour, Benchmarks, and Performance, Austin, 1995
- [2] Arnold K., Gosling J. The Java Programming Language. Sun Microsystems. Addison-Wesley, 1996.
- [3] Budd, T., Introducción a la programación orientada a objetos, Addison-Wesley
- [4] Marteens, I. La cara oculta de Delphi, Danysoft Internacional, 1998
- [5] Pressman, R. S., Ingeniería del Software: un enfoque práctico, 4ª ed., McGraw-Hill, 1997.
- [6] Renaud, Paul E. ; Introduction to client/server systems : A practical guide for system professionals ; John Wiley, cop. 1993.
- [7] Rumbaugh J., Blaha M., Permerlani W., Eddy F., Lorensen W. *Object Oriented Modeling and Design*. Englewood Cliffs, Nj. Prentice-Hall Iberoamericana, 1994.
- [8] Todd, B. ; Kellen, Vince , *Delphi : A Developer's Guide*, M & T Books, 1995.
- [9] Miller, T; Powell, D. Special Edition Using Delphi 3. QUE Corporation 1997
- [10] Wirfs-Brock R., Wilkerson B., Wiener L., *Designing Object Oriented Software*. Englewood Cliffs NJ. Prentice- Hall