

Introducción a los Ordenadores: Una forma diferente de aprender y de enseñar

Beatriz Otero Calviño y Pau Bofill Soliguer

Dpto. de Arquitectura de Computadores
Universidad Politécnica de Cataluña
Campus Nord- 08034 Barcelona
{botero, pau}@ac.upc.edu

Resumen

Este artículo describe la metodología utilizada en la asignatura Introducción a los Ordenadores impartida en el cuatrimestre 1A (curso otoño 2005) de Ingeniería de Telecomunicaciones (ETSETB) de la Universidad Politécnica de Cataluña.

La metodología aplicada se basa fundamentalmente en la utilización de técnicas que fomentan el trabajo en grupo y el desarrollo de proyectos. En esta edición, la segunda, los objetivos del curso son repartidos y agrupados en etapas. Cada etapa contiene un conjunto de objetivos generales, específicos y transversales definidos en la asignatura. Para cada etapa se definen las actividades y las tareas que debe realizar el estudiante.

Además, para complementar la planificación y la asignación de las tareas se incorpora la utilización de fichas y el desarrollo de esquemas. Las fichas describen el trabajo que el estudiante debe realizar y el lugar en donde debe llevar a cabo cada una de estas actividades: casa, laboratorio y/o clase. De ésta forma, el estudiante tiene claro qué trabajos deben entregar y cuando. Los esquemas resumen el contenido teórico de cada etapa.

1. Motivación

La propuesta de esta metodología surge a partir de la descripción de la filosofía Ancora [1]. Sin embargo, la idea no se ha expandido al resto de las asignaturas sino que se ha aplicado a dos grupos diferentes de la asignatura. Particularmente, esta experiencia promueve el trabajo en grupos y el desarrollo de proyectos para cumplir con los objetivos formativos [5].

Pensamos que si la actividad en grupo se promueve el estudiante no trabajará aislado del

resto, lo que motivará la realización de cada una de las actividades que realice en la asignatura. De ésta forma, el profesor funciona como: dinamizador, consultor y experto.

Las clases expositivas desaparecen casi completamente y las clases de aplicación se transforman en clases puramente prácticas. La documentación y los apuntes son materiales digitales desarrollados por los propios profesores de la asignatura. Además, se incluyen referencias bibliográficas que el estudiante puede consultar para documentarse sobre los contenidos teóricos de cada clase.

2. Objetivos por etapas

En [2], especificamos los objetivos de Introducción a los Ordenadores. Básicamente, los objetivos de la asignatura se dividen en tres etapas.

2.1. Primera etapa

Esta etapa la denominaremos “*etapa tortuga y expresiones*”, e incluye los siguientes objetivos formativos:

- Editar, compilar y ejecutar paso a paso un programa, utilizando el entorno gráfico de la tortuga (<tortuga.h>).
- Relacionar un fragmento de código con los resultados que genera.
- Identificar los tipos elementales (carácter, enteros, flotante) y su rango de valores, así como las operaciones asociadas a estos tipos.
- Asignar valores a variables.
- Utilizar funciones desde el primer día de clase, considerando únicamente el paso de parámetros por valor.
- Crear funciones y definir sus parámetros.

- Implementar expresiones y fórmulas matemáticas utilizando operadores lógicos y aritméticos.
- Utilizar el resultado de expresiones como términos de otras expresiones más complejas y como parámetros de funciones.
- Distinguir entre los parámetros y la lectura de datos.
- Utilizar sentencias condicionales (`if`) y sentencias iterativas (`for`, `while` y `do-while`).

Finalizada esta etapa el estudiante deberá ser capaz de:

- Escribir programas utilizando la librería *tortuga* y las instrucciones `scanf`, `printf`, `if`, `for`, `while` y/o `do-while` para realizar dibujos de complejidad media.
- Escribir programas que utilicen fórmulas y/o funciones matemáticas.

Ejemplos de los programas que el estudiante podrá realizar finalizada esta etapa son los siguientes:

```
#include <stdio.h>
#include <tortuga.h>
void cuadrado(int lado)
{
    int i;
    for ( i=0; i<4; i++ )
    {
        avanzar(lado);
        girar(90);
    }
}
void main()
{
    int lado;
    inicializar();
    avanzar(10);
    girar(90);
    scanf("%d%c", &lado);
    cuadrado(lado);
    finalizar();
}
```

Código 1. Programa con primitivas *Tortuga*

```
#include <stdio.h>
#include <math.h>
#define pi 3.141516
float circulo(float radio)
{
    float longitud;
    longitud = 2.0*pi*radio;
    return( longitud );
}
void main()
{
    float longi, radio;
    printf("Radio círculo:\n");
    scanf("%lf%c", &radio);
    longi=circulo(radio);

    printf("Longitud:%f\n", longi);
}
```

Código 2. Programa utilizando expresiones

2.2. Segunda etapa

Esta etapa es llamada “*etapa vectores y matrices*” y los objetivos formativos que incluye son:

- Utilizar la sentencia `switch`.
- Identificar la dirección y el valor de las variables.
- Hacer un mapa de memoria de las variables y de los parámetros locales.
- Definir y utilizar vectores y matrices.
- Codificar las operaciones asociadas con vectores y matrices: búsqueda, inserción y eliminación de un elemento.
- Pasar vectores y matrices como parámetros de una función.
- Utilizar vectores y matrices para almacenar resultados/datos.
- Asociar el pase de parámetros por referencia con un puntero que contiene una dirección.

Al finalizar esta etapa el estudiante debe ser capaz de escribir programas que utilicen vectores y/o matrices e implementar las funciones asociadas con estas estructuras de datos. Entre estas funciones se encuentran: la inicialización, la inserción, la eliminación y la búsqueda de un elemento.

2.3. Tercera etapa

Esta etapa la denominaremos “etapa *structs* y *ficheros*” e incluye los siguientes objetivos formativos:

- Definir nuevos tipos de datos (*typedef*).
- Construir nuevas estructuras de datos (*struct*).
- Pasar *structs* como parámetros.
- Implementar las operaciones básicas de los *structs* definidos.
- Gestionar ficheros utilizando las instrucciones: *fopen*, *fclose*, *fscanf* y *fprintf*.
- Utilizar los dispositivos predefinidos *stdin*, *stdout* y *stderr*.
- Utilizar la función *feof* para leer la información de ficheros de longitud desconocida.

Al finalizar esta etapa el estudiante debe ser capaz de implementar aplicaciones sencillas y modulares, conocida la especificación del problema, utilizando *structs* y *ficheros*.

3. Fichas y esquemas

Las fichas y los esquemas son recursos didácticos utilizados durante el curso.

Las fichas describen las actividades que el estudiante debe realizar y están asociadas con una etapa. La ficha incluye la planificación del trabajo que el estudiante debe realizar en: clase, laboratorio y casa. Básicamente, la ficha es una tabla de seis columnas donde cada columna describe, respectivamente:

1. la fecha en la que se debe realizar la actividad,
2. la descripción de la actividad,
3. si la actividad debe entregarse y cómo debe presentarse,
4. el tiempo que el estudiante debe dedicar,
5. el lugar en donde debe realizar la actividad y,
6. con quién debe trabajar (profesor, individual, colectivo, grupos pequeños).

Las actividades que el estudiante debe realizar van desde la lectura de las notas de clase o una referencia bibliográfica hasta, la solución de un proyecto.

Por otra parte, el esquema se utiliza para resumir el contenido y los objetivos formativos asociados a una actividad. El esquema da una visión global y sintetiza los casos en donde determinada información o conocimiento debe aplicarse. Generalmente, los esquemas se realizan para concluir una actividad. Por ejemplo, uno de los esquemas realizados consistía en sintetizar las diferencias entre *función* y *procedimiento*, considerando: cómo se realiza la llamada desde el programa principal y cómo se realiza el paso de parámetros.

4. Entorno de trabajo

El curso se gestiona utilizando la herramienta *Moodle* [4]. La elección de *Moodle* como herramienta de trabajo facilita el trabajo en el aula y permite una conexión directa con la clase. Se utilizan las sesiones de foros y se motiva a la participación. La planificación del curso se realiza por semanas de clase e incluye la siguiente documentación:

- *Ejercicios iniciales*. Esta documentación esta pensada para reforzar el aprendizaje de los objetivos de la etapa I.
- *Ejercicios de dificultad progresiva*. Esta documentación contiene ejercicios de autoaprendizaje para las etapas I y II. Esta colección de ejercicios tiene soluciones.
- *Ejercicios propuestos*. Esta documentación contiene ejercicios de dificultad similar al de las preguntas del examen final. Esta colección de ejercicios tiene soluciones.
- *Proyectos*. Esta documentación contiene la descripción de los proyectos de las etapas II y III.
- *Fichas*. Esta documentación contiene la planificación para cada semana.
- *Apuntes*. Esta documentación contiene las notas de clase o la referencia bibliográfica a la documentación apropiada,
- *Esquemas*. Esta documentación contiene una síntesis de los casos y la aplicación de los conocimientos de cada etapa.

5. Estrategias de trabajo en grupo

Para el desarrollo de los proyectos y el afianzamiento de conocimientos teóricos

utilizamos técnicas de trabajo en grupo. Particularmente las técnicas utilizadas fueron:

5.1. Concursos

Esta técnica consiste en que los estudiantes compitan por grupos realizando una serie de pruebas/ejercicios en las que deben demostrar su dominio y su iniciativa para resolver problemas de forma inesperada. Un ejemplo es *la carrera de los caballos de feria*. Esta actividad consiste en dibujar en la pizarra un camino (numerado) para cada equipo. Cuando el integrante de un grupo resuelve exitosamente un ejercicio, el caballo de ese equipo avanza a posición. En esta actividad todos los integrantes del grupo deben participar, y cuando se hayan concluido todos los ejercicios asignados, el caballo asociado a ese equipo alcanza la máxima posición. Un ejemplo de una actividad de este tipo realizada en clase (2 h) es la siguiente:

La clase se divide en grupos de 5 integrantes. Cada grupo tiene un responsable. El trabajo consiste en dar solución a un conjunto de ejercicios tipo *cuestionario con selección múltiple*. El responsable de cada grupo le comunica al profesor la respuesta de cada pregunta. Las respuestas se comunican lo más pronto posible. Si la respuesta de la pregunta es acertada el grupo se anota un punto (avance del caballo). En caso contrario, el grupo sólo tiene otra oportunidad para contestar esa pregunta. Si nuevamente la respuesta elegida no es correcta, el grupo pierde el turno de contestar esa pregunta y debe continuar contestando el resto de preguntas. La actividad la gana aquel grupo que logre el mayor número de puntos. Si hay más de un grupo con el mismo número de puntos se produce un empate.

5.2. Detectives y sospechosos

La clase se divide en 10 grupos, 9 de los cuales son llamados “*detectives*”. Sólo un grupo recibe el nombre de “*sospechosos*”.

Cada grupo esta formado por 5 estudiantes. La actividad consiste en descubrir qué estudiante del grupo “*sospechosos*” ha realizado una implementación incorrecta (*culpable*) de un ejercicio y quiénes de ellos siendo “*sospechosos*”, han realizado correctamente la actividad. Cada *sospechoso* es interrogado por todos los grupos “*detectives*”. Cada grupo tiene 3 minutos para

hacer preguntas. En un momento determinado, cada grupo puede interrogar a un *sospechoso*. De esta forma, los *sospechosos* van pasando por todos los grupos. Interrogados todos los sospechosos, el grupo tiene 3 minutos para determinar el *culpable*. Gana el/los equipo(s) que descubren al *culpable*. Si no descubren al *culpable*, gana el equipo “*sospechosos*”.

5.3. Proyecto

El proyecto es la actividad de mayor duración que los estudiantes deben realizar en grupos de 5 personas. El objetivo del proyecto es que el estudiante pueda visualizar de manera tangible el resultado de su esfuerzo resolviendo un problema [3]. Particularmente, para esta edición definimos dos proyectos. La duración de cada proyecto es de 3 semanas. El enunciado de cada proyecto lo realiza el profesor y es discutido en horas de clase.

Cada estudiante de forma individual realiza la lectura y comprensión del proyecto.

Posteriormente, los estudiantes se agrupan formando grupos. Cada grupo realiza una discusión del problema y estudia los posibles inconvenientes y los casos a desarrollar. Identificado el trabajo a realizar, cada grupo realiza la distribución de las funciones que cada estudiante debe hacer. De esta forma, cada estudiante propone el algoritmo de las funciones que debe implementar y lo discute con el resto. En una sesión de clase, el profesor revisa y comenta los algoritmos presentados y da el visto bueno para realizar la implementación (generación de códigos).

Durante el desarrollo de las funciones el profesor, junto con el resto de integrantes, comenta y sugiere algunas mejoras a posibles problemas detectados. Finalizada la fase de desarrollo y comprobada la funcionalidad del programa, se integran rutinas de validación de datos y otros juegos de pruebas.

La evaluación de cada proyecto se realiza en el *Moodle*. Cada grupo adjunta el fichero con la implementación del proyecto que ha realizado y la documentación necesaria. El profesor corrige el proyecto y comenta los errores y/o posibles mejoras. Adicionalmente, cada integrante del grupo realiza una evaluación individual del proyecto. Esta evaluación es común para todos los integrantes de todos los grupos y se evalúa como pregunta de examen.

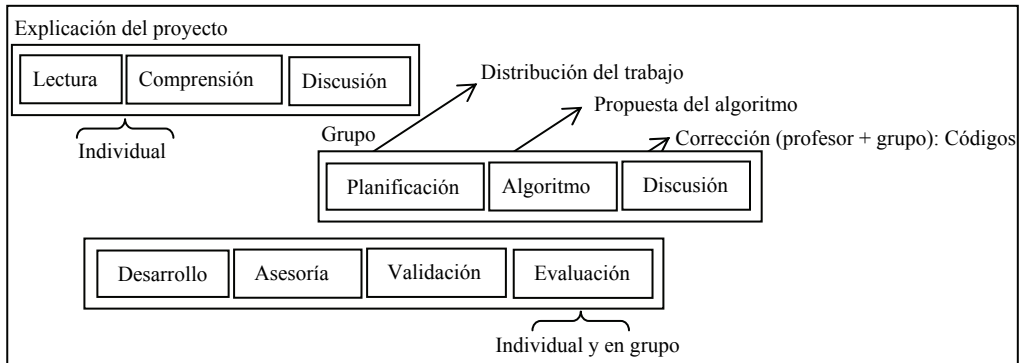


Figura 1. Etapas del desarrollo de proyectos

La figura 1 resume las etapas de desarrollo del proyecto.

Los proyectos desarrollados fueron:

- Etapa II: implementación del juego del Oteló.
- Etapa III: implementación del movimiento enroque corto del ajedrez.

6. Evaluación

La nota de un estudiante se obtiene a partir de siguiente fórmula:

$$Nota = 25\% A + 25\% L + 50\% F$$

con

$$A = 49\% T + 51\% E,$$

$$L = 55\% P_1 + 45\% P_2$$

donde

- A: es la nota de aplicación,
- L: es la nota de laboratorio,
- F: es la nota del examen final,
- T: es la nota de las actividades (excluye los proyectos),
- E: es la nota media de los 3 exámenes individuales realizados en clase,
- P₁: es la nota del primer proyecto y,
- P₂: es la nota del segundo proyecto.

7. Resultados

La experiencia se realizó a dos grupos completos de aproximadamente 50 estudiantes cada uno. Los

grupos seleccionados fueron el grupo 20 y el grupo 40.

La figura 2 muestra la cantidad de estudiantes suspendidos y aprobados con respecto a los estudiantes matriculados.

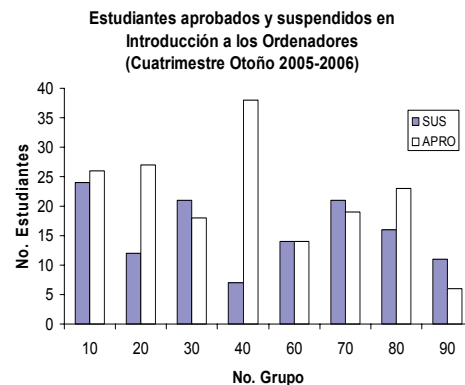


Figura 2. Rendimiento de los estudiantes del cuatrimestre 1A (ETSETB-UPC) en la asignatura Introducción a los Ordenadores

Como puede observarse, la cantidad de estudiantes aprobados en los grupos seleccionados es superior a la obtenida en el resto de los grupos. Sin embargo, la relación de estudiantes aprobados en el grupo 20 es muy similar a la obtenida en el grupo 10, pero existe una diferencia significativa en la cantidad de estudiantes suspendidos. Esto determina que la metodología garantiza que son pocos estudiantes los que no tienen éxito.

La metodología de trabajo fue aplicada por el mismo profesor en ambos grupos. A pesar de que el profesor es el mismo, el rendimiento es diferente en cada grupo. Esto nos permite inferir que no solo es el profesor y la metodología de enseñanza-aprendizaje los únicos componentes importantes que influyen en el rendimiento del curso. Pensamos que existen otros factores, tales como la motivación del grupo y la conexión que el profesor tenga con el mismo.

Para justificar estas diferencias de rendimiento entre los grupos, pensamos que el grupo 40 podía tener diferencias de rendimiento con respecto a otros grupos.

La figura 3 ilustra el porcentaje de rendimiento de todos los grupos del cuatrimestre 1A (ETSETB-UPC) en las diferentes asignaturas:

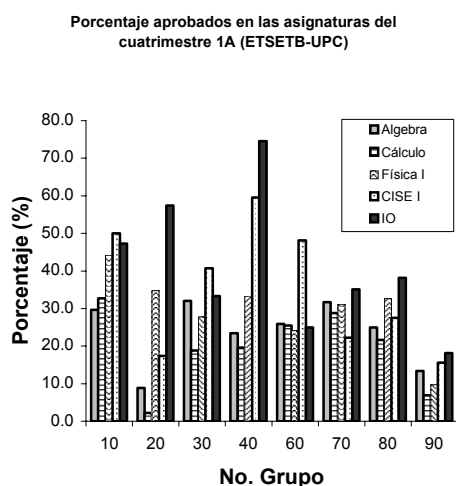


Figura 3. Porcentajes de rendimiento de los estudiantes del cuatrimestre 1A de la ETSETB-UPC (cuatrimestre Otoño 2005-2006)

Los resultados anteriores muestran como el grupo 40 esta formado por estudiantes que se encuentran en la media de rendimiento de todas las asignaturas. Sin embargo, los estudiantes del grupo 20 tienen un rendimiento más bien bajo comparado con el resto de los grupos. De esta forma, los grupos seleccionados no eran mejores que el resto de los grupos.

Por otra parte, se detectaron algunos inconvenientes a la hora de realizar la experiencia. El profesor debe dedicar mucho tiempo a dar retroalimentación. Para solventar este problema debemos incorporar técnicas de trabajo cooperativo que nos permitan evaluar el trabajo y proporcionar evaluación y retroalimentación al estudiante. Sin embargo, observamos que si realizamos un uso excesivo de estas técnicas, el estudiante se desmotiva. Debemos combinar los dos estilos de retroalimentación: la que proporcionan directamente los compañeros y la que realiza el profesor.

Como trabajos futuros, pensamos extender la aplicación de esta metodología a todos los grupos de la asignatura. Además, pensamos incorporar otras estrategias docentes que mejoren: el trabajo en grupo, el seguimiento y la participación de los estudiantes.

8. Conclusión

En este artículo hemos descrito la metodología del curso aplicada a dos grupos de la asignatura de Introducción a los ordenadores de ETSETB-UPC del 1A (cuatrimestre otoño 2005-2006).

El artículo comenta las actividades desarrolladas por los estudiantes y las estrategias utilizadas para fomentar el trabajo en grupos. Se utilizan los proyectos como actividades de motivación, desarrollo y evaluación del curso.

El trabajo muestra un mejor rendimiento de los grupos donde se realizó la experiencia, y muestra como la metodología resulta efectiva. Los estudiantes obtienen un rendimiento superior al resto de los grupos. Si comparamos las diferencias de rendimiento entre los grupos donde se aplico la metodología (grupos 20 y 40) y el grupo con mayor rendimiento (grupo 10), observamos que el rendimiento mejora entre 3.64 y 21.62 puntos.

Agradecimientos

Agradecemos la colaboración del Jefe de Estudios de la ETSETB al proporcionarnos la información global del rendimiento de todos los estudiantes matriculados en los cursos del 1A (cuatrimestre otoño 2005-2006).

Además, agradecemos la colaboración prestada por el resto de compañeros que imparten conjuntamente con nosotros la asignatura, al

realizar sus aportaciones durante el desarrollo de esta experiencia.

Por último, también queremos agradecer al Departamento de Arquitectura de Computadores de la UPC por el soporte a este trabajo.

Referencias

- [1] Bofill P., Otero B., Toribio E., Aroca J. M., Breitman M., Garcias P. y Sancho J. M. “Áncora: Aprendizaje organizado por tareas”. Actas de la XI Jornadas de Enseñanza Universitaria de la Informática, pp. 71-78, 2005.
- [2] Otero B. y Bofill P. “La asignatura Introducción a los Ordenadores en el contexto de la experiencia Áncora en la ETSETB-UPC”. Actas de la XI Jornadas de Enseñanza Universitaria de la Informática, pp. 229-236, 2005.
- [3] Project Based Learning Handbook (Second Edition). “A Guide to Standard-Focused Project Based Learning for Middle and High School Teachers”. Buck Institute for Education (www.bie.org).
- [4] Referencia web Moodle: www.moodle.org
- [5] Sánchez F. y Gavalda. “Objetivos formativos y estrategias docentes para el primer curso de las ingenierías informáticas”. Actas de la X Jornadas de Enseñanza Universitaria de la Informática, pp. 187-194, 2004.

