



La Metodología ANCORA aplicada a la asignatura Introducción a los Ordenadores de la ETSETB-UPC

Beatriz Otero, Pau Bofill

***Departamento de Arquitectura de Computadores,
Universidad Politécnica de Cataluña,
Campus Nord. 08034. Barcelona***

{botoero,pau}@ac.upc.edu

Resumen

Este trabajo describe las actividades realizadas y los resultados obtenidos al aplicar la metodología ANCORA a 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.

Esta metodología organiza el trabajo por tareas e incorpora actividades que fomentan el trabajo en grupo. Inicialmente, 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 en casa, en el laboratorio y en clase, de tal forma, que el estudiante tiene claro qué trabajos deben entregar y cuando. Los esquemas resumen el contenido teórico de cada etapa. Los resultados obtenidos en los grupos donde se aplicó la metodología muestran un rendimiento entre 3,64 y 21,62 puntos superior al resto de los grupos de la misma asignatura en donde no se aplicó la metodología.

Palabras clave

ANCORA, grupos, fichas.

EL PROYECTO

1. Introducción

La finalidad del proyecto consiste en utilizar la metodología ANCORA [1] y aplicarla en una asignatura del cuatrimestre 1A de ETSETB-UPC. Particularmente, la asignatura elegida fue *Introducción a los Ordenadores*. Esta metodología fue aplicada a dos grupos de estudiantes. El objetivo de esta experiencia era determinar si la metodología ANCORA resultaba

apropiada para mejorar el rendimiento de los estudiantes en donde se realizó la experiencia. Básicamente esto se determinaría comparando el rendimiento de los grupos en donde se realizó la experiencia con el resto de los grupos de la misma asignatura donde no se aplicó la metodología.

2. Descripción

La descripción de la experiencia incluye la especificación de los objetivos formativos asociados a la asignatura *Introducción a los Ordenadores*. Estos objetivos son agrupados formando etapas. De esta forma, se obtienen tres etapas y los objetivos formativos de cada etapa son:

Primera etapa:

- 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 de 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*).

Segunda etapa

- 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.

Tercera etapa

- 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.

En cada etapa se describen las actividades que el estudiante debe realizar. Para esto utilizamos las fichas. Las fichas contienen 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 realizar van desde la realización de una lectura hasta la solución de un proyecto.

Además de las fichas, utilizamos el esquema para resumir el contenido y los objetivos formativos que el estudiante debe asociar a una actividad. El esquema da una visión global y sintetiza los casos en donde cierta información/conocimiento debe aplicarse. Generalmente, los esquemas se realizan para concluir la realización de una actividad.

El curso se gestiona utilizando la herramienta Moodle [2]. 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. El curso se organiza por semanas de clase e incluye la documentación necesaria (ejercicios, fichas, apuntes, esquemas, descripción de los proyectos, etc.).

Para el trabajo en clase se utilizan técnicas de trabajo en grupo. Estas actividades fomentan el trabajo cooperativo y la integración de los estudiantes al curso. La etapa I comprende muchas actividades que tienen carácter individual o que deben realizarse durante la clase con el resto de compañeros del grupo: entregables, problemas, etc.

El proyecto forma parte de la actividad a la que el estudiante deberá dedicar mayor tiempo para realizarlo. La duración de cada proyecto es de 3 semanas y se especifican dos proyectos. El primero de ellos está relacionado con los objetivos que el estudiante deberá alcanzar finalizada la etapa II. El segundo, evalúa los objetivos que el estudiante deberá alcanzar finalizada la etapa III. 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. 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 junto con 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.

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

$$\text{Nota} = 25\% A + 25\% L + 50\% F$$

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.

3. Resultados

Este proyecto aplica la metodología ANCORA a dos grupos completos de aproximadamente 50 estudiantes en la asignatura Introducción a los Ordenadores. Particularmente, los grupos seleccionados fueron el grupo 20 y el grupo 40. La figura 1 muestra la cantidad de estudiantes suspendidos y aprobados con respecto a los estudiantes matriculados.

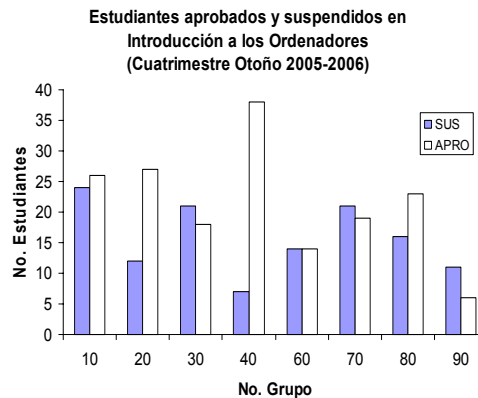


Fig 1. Rendimiento de los estudiantes del cuatrimestre 1A (ETSETB-UPC) en la asignatura Introducción a los Ordenadores.

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 con la metodología.

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 éstas diferencias de rendimiento entre los grupos, pensamos que el grupo 40 podía tener ciertas diferencias de rendimiento con respecto a otros grupos. La figura 2 ilustra el porcentaje de rendimiento de todos los grupos del cuatrimestre 1A (ETSETB-UPC) en las diferentes asignaturas:

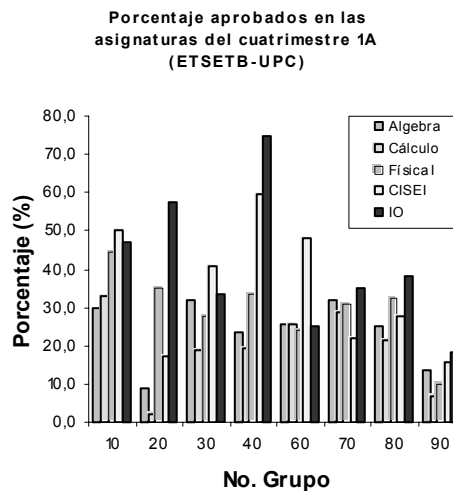


Fig 3. Porcentajes de rendimiento de los estudiantes del cuatrimestre 1A de la ETSETB-UPC (cuatrimestre Otoño 2005-2006).

Los resultados anteriores muestran como el grupo 40 estaba integrado por estudiantes que se encuentran en la media de rendimiento de todas las

asignaturas, mientras que el grupo 20 esta formado por estudiantes que obtienen un rendimiento inferior. De esta forma, los grupos seleccionados no eran los mejores grupos (grupos estándares).

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 de igual forma la evaluación y retroalimentación al estudiante.

Pero curiosamente, si se realiza 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.

4. Conclusiones

La aplicación de la metodología ANCORA a dos grupos de estudiantes de Introducción a los Ordenadores (ETSETB-UPC) mejora el rendimiento entre un 3.64 y 21.62 puntos con respecto al tercer grupo de mayor rendimiento (grupo 10).

Como trabajos futuros, pensamos extender la aplicación de esta metodología a todos los grupos de la asignatura. Pensamos que si resulta exitosa, esta metodología podría extenderse al resto de las asignaturas del 1A, coincidiendo con la idea original del proyecto ANCORA.

5. Referencias/Más información

- [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] Referencia Web Moodle: www.moodle.org