



MEMORIA DE LAS ACCIONES DESARROLLADAS

PROYECTOS DE MEJORA DE LA CALIDAD DOCENTE
VICERRECTORADO DE PLANIFICACIÓN Y CALIDAD



❖ DATOS IDENTIFICATIVOS:

Título del Proyecto

Carpuntal (Carné por puntos del alumno): Herramienta para la evaluación continua del alumno en el marco del EEES

Resumen del desarrollo del Proyecto

La evaluación continua del alumno universitario es una necesidad derivada de las directrices que emanan del EEES. Este proceso va a obligar al profesor universitario a cambiar su dinámica actual de evaluación, basada en muchos casos en pocas calificaciones de sus alumnos, a un sistema en el que posea un perfil mucho más completo de los conocimientos teórico/prácticos de cada alumno y del proceso de progreso en su aprendizaje, para asignar una calificación final.

El proyecto que se solicita plantea desarrollar una herramienta informática que permita tener un perfil de cada alumno para reflejar cual es la evolución del aprendizaje del mismo a lo largo del curso académico. La herramienta será configurable de manera que el profesor pueda usarla bajo sus propios criterios.

Se pretende que esta herramienta ayude al profesor a medir de que forma cada alumno aprovecha los contenidos de cada clase impartida y a la vez motivar al alumno para que asista a clase con una participación activa y positiva.

Nombre y apellidos

Código del Grupo Docente

Coordinador/a:

Francisco Jose Madrid Cuevas 083
Rafael Medina Carnicer 083

Otros participantes:

Angel Carmona Poyato 083
Nicolas Luis Fernandez Garcia 083
Rafael Muñoz Salinas 083

Asignaturas afectadas

Nombre de la asignatura

Área de Conocimiento

Titulación/es

Metod. y Tec. de la Prog.	C. Computación e I.A	I.T. Inf. Sistemas
Metod. y Tec. de la Prog.	C. Computación e I.A	I.T. Inf. Gestión
Fundamentos de Informática	C. Computación e I.A	I.T. Ind. Mecánica
Est. de Datos y de la Inform.	C. Computación e I.A	I.T. Inf. Gestión
Inteligencia Artificial	C. Computación e I.A	Ing. Informática
Sistemas de Percepción	C. Computación e I.A	Ing. AIE
Percepción	C. Computación e I.A	I.T. Inf. Sistemas
Procesadores de Lenguaje	C. Computación e I.A	Ing. Informática

MEMORIA DE LA ACCIÓN

1. Introducción

La puesta en marcha del EEES lleva implícito el cambio en los sistemas tradicionales de evaluación de los alumnos. En las propias directrices del EEES se acentúa en gran medida el hecho de que el objetivo es propiciar el aprendizaje activo del alumno en detrimento de la labor tradicional del profesor universitario que fundamentalmente transmite todos los conocimientos con el mayor detalle posible.

Este hecho tiene como consecuencia inmediata que el proceso de evaluación tradicional, basado en muchos casos en un número reducido de calificaciones sobre las que se basa la evaluación final del alumno, debe necesariamente cambiar. Para ello parece que la única posibilidad es que el profesor universitario debe dedicar cada vez más tiempo a evaluar sobre sus alumnos el grado de aprendizaje continuo de pequeñas unidades temáticas.

No existen, en el momento actual, herramientas informáticas que faciliten esta labor. Por ejemplo, si el profesor quiere conocer el grado de asistencia de cada alumno necesita "apuntar" de forma manual ese dato a diario. Análogamente, si el profesor quiere realizar un proceso de evaluación continua debe usar una hoja de cálculo para almacenar esos datos y tratarlos posteriormente.

En el estado actual de las nuevas tecnologías no parece excesivamente complicado facilitar esa labor diseñando una herramienta "a medida" de lo que puede necesitar el profesor. La búsqueda de alumnos con criterios tales como "aquellos con una media de asistencia inferior al 50%", "aquellos que en un instante posean una calificación inferior a 5" ó incluso criterios combinados tales como por ejemplo "alumnos con alta asistencia a clase, cuya calificación actual sea inferior a 5 y que además hayan sido evaluados menos de 4 veces" pueden ser implementadas para ser llevadas a cabo en tiempo real y por tanto son susceptibles de poder ser usadas en el transcurso de la labor docente diaria.

Una herramienta que incorpore las ideas expuestas podría ser usada por el profesor universitario durante el transcurso de la clase para llevar a cabo una evolución continúa profunda de cada alumno. Los beneficios derivados de esta herramienta que se plantea no solo repercuten en la labor del profesor sino que pueden constituir un estímulo muy importante para el alumno universitario, ya que estimularía la asistencia a clase y motivaría la participación activa. Como efecto beneficioso añadido. El alumno conocería "cada día de clase" cual es su evolución en el aprendizaje de la asignatura teniendo siempre la posibilidad de rectificar si su progreso no es el adecuado.

2. Objetivos

El objetivo general es desarrollar una aplicación informática que facilite la labor del profesor universitario en el proceso de evaluación continua a sus alumnos y que simultáneamente estimule al alumno en su labor de aprendizaje.

Los objetivos específicos que se pretenden conseguir con la herramienta son:

- El profesor debe conocer en qué medida un alumno ha asistido a clase.
- El profesor debe saber en qué medida un alumno aprovecha la clase (en función de las respuestas a las preguntas que se realizan).
- El profesor debe motivar con el sistema de puntos la participación activa y positiva por parte del alumnado durante el desarrollo de la asignatura.

3. Descripción de la experiencia

El fruto de este proyecto ha consistido en una herramienta software para asistir al profesor durante la clase con los siguientes objetivos:

- Conocer en qué medida los alumnos han asistido a las clases.
- Conocer en qué medida los alumnos han aprovechado las clases.
- Motivar la participación activa de los alumnos durante las clases.

La herramienta desarrollada se denomina CARPUNTAL, acrónimo de “Carné de puntos del alumno” y cumple con los objetivos marcados.

Utilizando la herramienta, en cada clase, se realiza un sondeo aleatorio de todo el alumnado matriculado, forma que permite estimar a asistencia o no a clase de un alumno a lo largo del curso académico.

A través de las preguntas/respuestas obtenidas por los alumnos el profesor puede estimar el grado de aprovechamiento de las clases por parte del alumno y utilizar esta información en el proceso de evaluación del rendimiento del alumno.

Por último, la herramienta genera automáticamente un informe de progreso del alumno (su estado de puntos del carné) en formato HTML de forma que el profesor pueda fácilmente publicar este informe tras cada clase, por ejemplo mediante la plataforma moodle, y de esta forma

motivar a los alumnos con un pobre aprovechamiento de las clases a realizar una participación mas activa de las mismas.

4. Materiales y métodos

Para el desarrollo del proyecto, dado que trata de desarrollar una aplicación informática se ha seguido una metodología para desarrollo de software, concretamente se ha utilizado el paradigma orientado a objetos [1]. Para la implementación se ha utilizado el lenguaje C++ [2].

Se ha seleccionado este paradigma porque tiene una serie de ventajas frente al paradigma tradicional de descomposición funcional. Tradicionalmente, un problema informático se ataca produciendo algún tipo de representación de datos y procedimientos que operan sobre esos datos. Bajo este modelo, los datos son inertes, pasivos e incapaces. Están a la completa merced de un gran cuerpo procedimental, que es activo, lógico y todopoderoso.

El problema con esta aproximación es que los programas los escriben programadores, que son humanos y sólo pueden retener cierto número de detalles en sus cabezas en un momento determinado. A medida que crece el proyecto, el núcleo procedimental crece hasta un punto que se hace difícil recordar cómo funciona todo el conjunto. Pequeños lapsos de pensamiento o errores tipográficos llegan a ser errores muy ocultos. Empiezan a surgir interacciones complejas e inintencionadas dentro de este núcleo y el mantenimiento se convierte en algo parecido a transportar un calamar gigante intentado que ninguno de sus tentáculos te alcance la cara. Existen políticas de programación que ayudan a minimizar y localizar errores dentro de este paradigma tradicional pero existe una solución mejor que pasa fundamentalmente por cambiar la forma de trabajar.

Lo que hace la programación orientada a objetos es delegar la mayoría del trabajo mundano y repetitivo a los propios datos; modifica el concepto de los datos que pasan de pasivos a activos. Dicho de otra forma:

- dejamos de tratar cada pieza de dato como una caja en la que se puede abrir su tapa y arrojar cosas en ella, y
- empezamos a tratar cada pieza de dato como una máquina funcional cerrada con unos pocos interruptores y diales bien definidos.

Lo que se define anteriormente como una “máquina” puede ser, en su interior, algo muy simple o muy complejo. No se puede saber desde el exterior y no se nos permite abrir la máquina (excepto cuando estamos completamente seguros de que algo está mal en su diseño), por lo que se nos obliga a conmutar interruptores y leer los diales para interactuar con los datos. Una vez construida, no queremos tener que pensar en como funciona internamente.

Un aspecto muy importante a tener en cuenta es asegurar la calidad del software desarrollado. Con este fin se han utilizado dos herramientas:

- Se ha utilizado el paradigma de la programación por contrato [1] en el desarrollo de las interfaces de las clases.
- Se han implementado pruebas de regresión para asegurar que la re-factorización/adición de funcionalidad a la herramienta no afecta a lo ya desarrollado. Para ello se ha utilizado la herramienta CppUnit [3].

Otro aspecto muy importante en el desarrollo de una herramienta software es generar la documentación necesaria para su utilización tanto por parte del usuario como por parte de futuros desarrolladores que pretendan corregir fallos o extender su funcionalidad. Para este fin, se ha utilizado la herramienta de auto documentación Doxygen [4]. Además se ha desarrollado una guía para el usuario de la herramienta.

El entorno de desarrollo utilizado ha sido un PC estándar con sistema operativo GNU/Linux, distribución Debian [5].

Para facilitar su portabilidad a distintos entornos de desarrollo se han utilizado las herramientas autotools [6].

5. Resultados obtenidos y disponibilidad de uso

El resultado de este proyecto ha sido un programa software denominado “Carpuntal” que proporciona la siguiente funcionalidades:

- Gestionar la lista de alumnos matriculados en una asignatura, obteniendo la información a partir del fichero “lista de clase” generada por la aplicación de gestión de la Uco. Posteriormente a la incorporación de los datos iniciales, se permite añadir/modificar alumnos e incluso actualizar la lista inicial con una nueva lista obtenida de a partir de la mencionada aplicación de gestión de la Uco.
- Sondear aleatoriamente al alumnado matriculado en una asignatura con objeto de controlar la asistencia a clase.
- Evaluar de forma continua el aprovechamiento de las clases “presenciales” por parte del alumnado.
- Generar informes de aprovechamiento de las clases “el carne de puntos” para cada alumno en formato compatible WEB.

Así mismo la aplicación es extensible para añadir otras futuras funcionalidades, como nuevos métodos de sondeo, nuevas formas de calificación, nuevas formas de participaciones, etc...

Además, la aplicación desarrollada tiene como características principales: la facilidad de uso, así como su portabilidad a plataformas Unix y MsWindows} (utilizando el entorno Cygwin [7]).

6. Utilidad

La aplicación informática desarrollada facilita la labor del profesor universitario para realizar un proceso de evaluación continua a sus alumnos y simultáneamente también proporciona una herramienta poderosa para estimular al alumno en su labor de aprendizaje.

Los beneficios específicos derivados del uso de la aplicación desarrollada son:

- El profesor puede conocer un estimación de en qué medida un alumno ha asistido a clase, sin necesidad de “pasar lista” al comienzo de cada clase con la consiguiente pérdida de tiempo.
- El profesor puede conocer en qué medida un alumno aprovecha la clase gracias al proceso de evaluación continua que se realiza fácilmente con la aplicación (las respuestas a las preguntas que se realizan).
- El profesor puede utilizar la aplicación una herramienta para motivar, con el sistema de puntos, la participación activa y positiva por parte del alumnado durante el desarrollo de la asignatura.

7. Observaciones y comentarios

La aplicación Carpuntal se ha desarrollado bajo licencia GPL [8] por lo que se da permiso para su instalación, distribución, modificación y estudio bajo los términos indicados por esta licencia.

En el CDROM anexo a esta memoria se encuentra una copia fuente de la aplicación donde se puede consultar también una copia verbatim de la licencia GPL.

Para poder instalar la aplicación se necesita contar con un sistema operativo tipo Unix, o en su defecto para usuarios de la plataforma MsWindows, instalar el entorno Cygwin [7].

La aplicación se distribuye en código fuente, por lo que es necesario compilarla previamente a su instalación. Sin embargo todo el proceso está automatizado mediante un script denominado 'configure'. Para compilar e instalar la aplicación siga los pasos siguientes:

Configuración de la aplicación

Una vez descomprimido el fichero con los fuentes de la aplicación, entramos en el directorio creado con los mismos y ejecutamos el script configure:

```
cd <PATH>/carpuntal
```

```
./configure
```

Esto configura y prepara el entorno para que la aplicación se pueda compilar en su sistema. Es posible que el script no encuentre herramientas necesarias para compilar la aplicación. Si esto ocurre, usted deberá instalar dichas herramientas en su sistema y volver a ejecutar el script.

Compilar la aplicación

Una vez comprobado el entorno y configurada correctamente la aplicación, ésta ya se puede compilar (desde el directorio raíz de la distribución) deberá ejecutar el comando:

```
make
```

Esto compilará la aplicación.

Instalar la aplicación

Una vez compilada correctamente la aplicación, para instalar la aplicación y, dependiendo del sitio que usted haya decidido para ello (se puede modificar con una opción del script configure, ver 'configure --help') puede que necesite privilegios de “administrador” para realizar este paso. Si es así acceda como “administrador” para realizar el siguiente paso.

```
make install
```

8. Auto-evaluación de la experiencia

Debido a que el producto de esta experiencia ha sido una aplicación software, la auto-evaluación realizada ha consistido en un ciclo “clásico de pruebas” en el desarrollo de una aplicación software.

Dada que se ha utilizado una metodología de desarrollo orientado a objetos y el paradigma de programación por contrato, se ha utilizado pruebas de regresión sobre cada clase desarrollada utilizado para ello la herramienta de pruebas CppUnit.

También se han desarrollado baterías de test de esfuerzo simulando listas de clase muy grandes con una gran cantidad de operaciones sobre las mismas.

Por último un prototipo de la aplicación también ha sido utilizada durante el desarrollo de la asignatura “Fundamentos de Informática” de la titulación de Ing. Técnico en Informática de Sistemas con éxito.

9. Bibliografía

- [1]B. Meyer. *Construcción de Software Orientado a Objetos*. Prentice Hall, segunda edición, 1997.
- [2]Bjarne Stroustrup. *The C++ programming language*. Addison-Wesley, Massachusetts, 3rd. edición, 1997.
- [3]Cpunit. *A C++ unit testing framework*. Accesible en <http://cppunit.sourceforge.net/doc/1.8.0/index.html>.
- [4]Dimitri van Heesch. *A documentation system for C++, C, Java, Objective-C, Python, IDL (Corba and Microsoft flavors), Fortran, VHDL, PHP, C#, and to some extent D*. Accesible en <http://www.stack.nl/~dimitri/doxygen/manual.html>.
- [5]Debian. The Debian project a GNU/Linux distribution. Accesible en <http://www.debian.org>.
- [6]Autoconf. *Produces shell scripts which automatically configure source code*. Accesible en <http://www.gnu.org/software/autoconf.html>.
- [7]Cygwin. *A Linux-like environment for Windows*. Accesible en <http://www.cygwin.com>.
- [8]GPL. The Gnu Public License. Accesible en <http://www.fsf.org/licenses/licenses.html>.

Lugar y fecha de la redacción de esta memoria

En Córdoba, a 29 de agosto de 2008.