

El concepto de eje de actividad, una buena ayuda metodológica para el diseño curricular en el marco EEES: aplicación al caso de la informática como materia básica en los planes de estudio de los grados en ingenierías no informáticas

Ferran Virgós Bel
EUETIB, UE Electrónica, Dep. LSI
Universidad Politécnica de Cataluña, UPC
Urgell 187, 08011 Barcelona
Ferran.Virgos@upc.edu

Resumen

JENUI ha sido, tradicionalmente, un foro sensible a experiencias, metodologías docentes y planificación curricular aplicables a la enseñanza de la informática en el nuevo marco del Espacio Europeo de Enseñanza Superior (EEES), pero la mayoría de aportaciones presentadas en las últimas ediciones, se han proyectado sobre los estudios de grado en informática, cosa lógica, por otra parte.

En este trabajo, con una vocación complementaria, recogemos las recientes consideraciones metodológicas, el nuevo enfoque del marco de Bolonia, las directrices curriculares de los libros blancos, y nuestra experiencia personal, deseando compartir un replanteamiento global del diseño curricular EEES, proponemos un método para ayudar a este replanteamiento y lo aplicamos al caso de la enseñanza de la informática como materia básica de las nuevas titulaciones de grado de las ingenierías no informáticas, aunque la propuesta puede generalizarse, en forma inmediata, a otras materias y titulaciones.

1. A modo de Introducción

1.1. Presentación

En línea con las declaraciones de Sorbona (25 mayo 1998), Bologna (19 de junio 1999), Praga (19 de mayo 2001), Berlin (19 septiembre 2003) y Bergen (19-20 de mayo 2005) [4], en las últimas ediciones de Jenui hemos podido ver numerosos trabajos dedicados a la adaptación de la enseñanza de la informática al nuevo marco del Espacio Europeo de Enseñanza Superior (EEES),

del que el Reino Unido tiene asignada la responsabilidad de pilotar el proceso hasta la reunión de ministros de mayo 2007 en Londres [5].

No obstante, la mayoría de aportaciones presentadas en las últimas ediciones de Jenui se han proyectado sobre los estudios de grado en informática, existiendo, únicamente, un limitado número de aproximaciones que pudieran marcar alguna dirección en la futura tarea de adaptación al nuevo entorno de la enseñanza de informática para ingenierías no informáticas. En [7] y [8], por ejemplo, pueden verse algunas propuestas, aunque se trata, todavía, de meritorias experiencias parciales, desarrolladas sobre la base del contexto curricular actual.

1.2. Los libros blancos de ANECA

Si buscamos un marco en los libros blancos de ANECA para ingenierías no informáticas ([1] y [2]), podemos encontrar casos como el libro blanco de ingeniería de minas y energía, donde no hemos sabido identificar materias específicas de informática, hasta el libro blanco del grado en ingeniería de telecomunicación que sugiere, para ingeniería electrónica, una materia de “Fundamentos de Informática y Telemática” con un total de 9 créditos ECTS (como es sabido, 1 crédito ECTS = 25 a 30 horas de trabajo del alumno).

Por su parte, la mayoría de grados en ingeniería industrial clásica (mecánica, electricidad, electrónica, organización, etc.) [2], optan por una materia de Informática a la que asignan 4,5 créditos ECTS. La verdad es que este planteamiento resulta sospechosamente equivalente a los 6 créditos convencionales asignados en la troncalidad actual de las

ingenierías técnicas. En estas referencias, cuando aparecen descriptores, éstos también son muy parecidos, incluso idénticos, a los actuales: *estructura de ordenadores, programación y sistemas operativos*. Unos descriptores muy ambiciosos y, necesariamente, poco homogéneos en peso para estos estudios.

Ahora, los conocedores del área no sólo tenemos la opción, sino la obligación de intentar desarrollar y “ajustar” los objetivos didácticos, dentro de las limitaciones del marco establecido, de modo que se nos permita mejorar el resultado final.

1.3. Elementos conceptuales genéricos a considerar en el nuevo planteamiento

Todo el mundo está de acuerdo en definir como denominador común del nuevo marco el desplazamiento del paradigma docente “de la enseñanza” al “aprendizaje”. Con esta base, se han dedicado numerosos trabajos a “recoger” elementos potencialmente integrantes del nuevo modelo:

- Formulación de objetivos didácticos de asignaturas por niveles [11].
- Diseño curricular orientado a obtención de competencias (Buendía et al, en [6]).
- Modelización del proceso de aprendizaje (Kolb en [10]).
- Taxonomía de Bloom y su influencia en la definición de objetivos didácticos y su proceso de evaluación (en [3]).
- Evaluación formativa de competencias (ver [14]).
- Un sin fin de aspectos metodológicos tales como PBL (*Problem, o Project, Based Learning*), aprendizaje cooperativo (AC), congreso de alumnos (Ibáñez et al, en [9]), biblioteca compartida de grupo (en [15]), “*Blended learning*” (Piña, en [12]), etc.
- Elementos a considerar en la dedicación ECTS del estudiante (por ejemplo en [16]).

A nivel más práctico, y como consecuencia de lo anterior, casi todo el mundo parece estar, también, de acuerdo en algunas consideraciones generales hacia las que deberíamos tender:

- Conceder mayor importancia a las “necesidades” del “perfil” de la titulación en el puesto de trabajo.

- Visión más global del aprendizaje (competencias genéricas, específicas y transversales, actitudes y habilidades) con clara orientación a los “resultados” de ese aprendizaje y su utilidad en la integración laboral del estudiante.
- En consecuencia, conceder menos peso al concepto clásico de “programa”, menos peso a las clases de “teoría” y a la planificación “del profesor”. Por el contrario, mayor necesidad de planificación, más detallada, del trabajo de los estudiantes.
- Cambio, obvio, del “*role*” del profesor a una posición más de soporte. Naturalmente no puede hacerse abuso, y debe matizarse según el “nivel” de los estudios. Parece de sentido común aceptar que no es lo mismo el primer cuatrimestre que el quinto.
- Motivación de los alumnos conseguida a través de su capacidad de decisión durante el proceso de aprendizaje (PBL se insertaría en este punto).
- Potenciación del trabajo entre iguales como un elemento más de motivación de los alumnos y parte integrante de estrategia de la propia metodología docente (las metodologías de aprendizaje cooperativo se situarían en este bloque).
- Consideración del proceso y mecanismo de evaluación como elemento esencial del diseño curricular (lo ideal sería que se diseñara la estructura de evaluación después de los objetivos, pero antes de programar la materia).
- Aceptación del hecho de que el alumno aprende más y mejor cuando tiene claro el objetivo a alcanzar, cuando trabaja él (en forma activa), cuando interactúa con iguales y cuando sabe que hay una relación entre el trabajo a hacer hoy y la culminación del proceso global: “no puedo librarme porque si hoy no hago esto, ¡se acabó!” (la celebrada charla en Jenui del profesor Valero con su metáfora entre ECTS y el tour de Francia es un buen exponente de esta idea).
- Aceptación del hecho que en el caso de decidir utilizar clases expositivas, es bueno incorporar aspectos de interactividad. En particular, intercalar actividades cuña para implicar al alumno y recuperar su atención (ya sean ejercicios cortos, discusiones de grupo, preparar preguntas, etc.).

2. A modo de “nudo”

2.1. A efectos prácticos: antes éramos tontos y ¿lo estábamos haciendo todo mal?, ¿debemos partir de cero?

Los más tradicionales tenderían, seguramente, a hacer más o menos lo mismo que ahora en el nuevo marco (¡todos sabemos multiplicar créditos por un coeficiente y seguir haciendo lo de siempre!).

Por el contrario los más innovadores podrían pensar que quizás deberíamos olvidar el pasado, comenzando de nuevo por la definición del perfil del titulado, desde cero, e ir descomponiendo éste en sus competencias y habilidades en forma “descendente” como si se tratara del diseño de un algoritmo por refinamientos.

En este punto, me vienen a la memoria una frase oída en clase, de boca del excelente profesor Manolo Costa. Decía Don Manuel: “Diseñar en forma *top-down* es un planteamiento teóricamente impecable”. Y añadía: “... pero no siempre es posible (ni conveniente), en la práctica”. Naturalmente, se refería a aquellos casos en que la persona que diseña no conoce con suficiente nivel de detalle “todos” los aspectos del proceso a diseñar,... o es muy costoso renunciar a lo ya hecho.

Yo, desde luego, no me considero con un conocimiento “completo y detallado” de lo que debo hacer ante un diseño curricular, ni estoy dispuesto a renunciar a “todo” mi material. En consecuencia, no creo que la mejor solución sea partir de cero e iniciar un solitario y prepotente proceso “*top-down*” según mi único criterio.

Sin duda, en muchos años, tantos profesores, no hemos sido tan tontos como para hacerlo todo mal. Pero SI puede ser un buen principio reconocer que, en forma mayoritaria y reiterada, hemos cometido algunos errores:

- Demasiado a menudo funcionábamos por mimetismo: “a mi me explicaron esto así... y me fue bien...¿porqué lo voy a cambiar?”, o “siempre lo hemos hecho así”. Reconozcámoslo, hemos sido poco innovadores (“de tanto enseñar, ¡se olvidó de aprender!... “).
- Nos gustaba tanto la teoría que, a menudo, no llegábamos a la aplicación. O nos gustaba

tanto un tema que se nos acababa el curso en él (algunas veces he oído en mi escuela, a los profesores de electrónica quejándose de sus compañeros de física porque nunca llegan a la electricidad.... ¡como mucho se quedan en el magnetismo ...!). Seguramente ellos mismos, alguna vez, han hecho algo parecido, ... ¡y yo, también, claro !.

- El contenido de la clase de hoy estaba en un marco tan genérico que no veremos su aplicación hasta mucho más adelante. Incluso en alguna ocasión habremos dicho a los alumnos “esto os servirá en otra asignatura”.
- Los programas eran esencialmente correctos pero su desarrollo cronológico no: estaban orientados a dar la base, no tanto pensando en la utilidad o la aplicación inmediata. Además, no había realimentación o ésta era demasiado tardía.
- Cuando intentábamos valorar el aprendizaje pensábamos en un ejercicio integrador que los alumnos no habían visto nunca completo hasta el día del examen. El resultado era aparentemente desastroso y nuestra depresión inminente. Entonces llegaba la frustración y la sospecha de que los objetivos no se habían alcanzado, pero ¡era demasiado tarde!.
- Además, la evaluación no era justa porque pedíamos al alumno algo que nosotros habíamos pensado “después” de exponer el tema en clase. Dicho de otro modo, el examen lo escribíamos después de terminar la asignatura cuando está bastante claro que debería ser al contrario. Incluso cometíamos grandes errores en los pesos de las partes del temario: en el desarrollo del curso quizás habíamos asignado unos pesos y dedicado mucho tiempo a unos temas, mientras en el examen, utilizábamos unas ponderaciones distintas.

Pero parece claro que no debemos tirar todo nuestro material y nuestros planteamientos anteriores. Pienso que no se trata tanto de aplicar de manera sistemática PBL y aprendizaje cooperativo a lo de siempre (que, a lo mejor, ¡también lo tenemos que hacer!), sino de buscar un nuevo planteamiento global alineado con el nuevo marco y que eluda nuestros errores históricos.

2.2. Diseño curricular orientado a eje de actividad, una posible solución.

Pienso, por el contrario, que deberíamos darnos cuenta que las materias tienen que orientarse a conseguir unos objetivos didácticos claros que puedan ser “medibles”. Y esos niveles a alcanzar y sus criterios de evaluación asociados deben ser conocidos “a priori”.

El objetivo debe llevar asociado un criterio claro de medición, que podría ser una actividad de solución de un problema, o bien la realización de un trabajo, que justifique la adquisición de alguna capacidad o competencia.

Por su parte, la planificación docente deberá ir orientada a optimizar el conjunto objetivo-aprendizaje mediante la mejora del propio proceso de adquisición.

Ahora, sólo queda percibirse que el criterio-herramienta de medición y el objetivo deberían apuntar en la misma dirección y se convierten en la base del proceso enseñanza-aprendizaje: es el EJE de ACTIVIDAD.

En realidad, el eje de actividad es el indicador de la adquisición del conocimiento, competencia o habilidad. El programa no desaparece, pero ya no es el elemento motor sino que estará en un segundo nivel.

De hecho, el eje de actividad, para estructurar el proceso, deberá descomponerse en ejes de actividad parciales que serán los elementos que constituirán el esquema del diseño curricular.

Detrás de estos ejes, organizaremos todas las actividades didácticas necesarias para su consecución, incluidas, como es natural, las famosas clases de teoría clásicas, y todas aquellas actividades complementarias que nos aconsejen las metodologías más novedosas.

El planteamiento de eje (o ejes) de actividad y ejes parciales tiene un cierto paralelismo con PBL, aunque a pequeña escala. Naturalmente, se diferencia de éste último en que los alumnos no eligen el proyecto. Es, sin embargo, mucho más adecuado para alumnos de primeros cursos que precisan mayor guía y, en cambio, mantiene casi todas las ventajas de aquel, incluso mayores, si admitimos que los ejes de actividad parcial se pueden estructurar mejor y dar mayor cobertura a los objetivos didácticos.

Además, hay que añadir que, lógicamente, la integración del aprendizaje cooperativo en este

marco de eje de actividad y ejes parciales, es inmediata.

2.3. ¿Cómo identificamos los ejes de actividad parciales?

Hemos admitido que no se trata de tirar por los suelos nuestro trabajo y experiencias de los últimos años. Por el contrario, se trataría de aprovechar ambos con un nuevo planteamiento orientado a enseñar no sólo a “saber” sino a “saber hacer” y/o “saber resolver” y/o “saber diseñar”. ¿De dónde podemos sacar esta información?....

Si somos coherentes con nosotros mismos, deberíamos aceptar que sería lógico encontrarlo en los exámenes. Este ya era el criterio básico contenido en [13], donde se propone realizar un análisis de los exámenes, como inestimable herramienta metodológica de ayuda para dar soporte a la reformulación de los objetivos didácticos de una determinada materia.

3. A modo de desenlace : aplicación al caso de Fundamentos de Informática para ingenierías no informáticas

Cada uno deberá hacer su propio proceso. En mi caso, a modo de guía, he hecho este ejercicio ampliando los contenidos de los exámenes con algunos problemas típicos que me han parecido relevantes, y he obtenido la siguiente lista (a modo de “*check-list*” preliminar):

3.1. Elementos generales y de cultura

TERMINOLOGÍA INFORMÁTICA

- Conocer la terminología informática básica. Ser capaz de explicar cualquiera de estos términos y conceptos asociados.

ESTRUCTURA, CONFIGURACIONES y CAPACIDADES.

- Conocer los elementos de la estructura de un ordenador (en especial CPU y memorias), sus posibles configuraciones, capacidades y prestaciones típicas, así como su evolución en el tiempo.

3.2. Representación interna

- Conocer los fundamentos de la representación interna. El concepto de la representación de instrucciones pero, sobre todo, la representación de los datos. En particular, los

datos numéricos: naturales y enteros, así como valores con decimales. Se incluye, también, la representación de datos alfanuméricos. Además, los propios alumnos pueden hacer (o usar) programas simples de simulación para “visualizar” esta representación.

- Conocimiento de la problemática de la limitación en la “capacidad de representación” de los naturales y enteros (*overflow*). Problemática de la precisión y capacidad de representación (*overflow* y *underflow*) de los datos con decimales.
- Conocer los fundamentos de la representación de imágenes (en pantalla y estándares de imagen como JPG).

3.3. Transmisión de la información

- Conocer los fundamentos y alternativas de transmisión de la información entre subsistemas y evaluar su importancia.
- Conocer el concepto de códigos detectores y correctores de errores..

3.4. Conocimientos complementarios y sistemas operativos

- Características básicas y fundamentos tecnológicos de algunos dispositivos externos.
- Conocer los diferentes tipos de arquitecturas de sistemas informáticos.
- Conocer las diferentes formas de explotación.
- Conocer el concepto de lenguaje máquina y necesidad del concepto de lenguaje de programación de alto nivel.
- Conocer el concepto de máquina virtual como soporte a la disponibilidad de lenguajes de programación en un ordenador básico.
- Aplicación del concepto de máquina virtual a la comunicación hombre-máquina. Conocer el concepto de máquina “ampliada” como estrategia de estructuración multinivel de las diferentes máquinas virtuales.
- Entender el concepto de Sistema operativo como fundamento de la máquina ampliada. Visión “multi-nivel” de un sistema operativo.
- Conocer las funciones y elementos de un sistema operativo.
- Entender la importancia del SO como gestor de recursos y los inconvenientes de no disponer de algunos de sus servicios.

- Entender la importancia de un sistema operativo multitarea en control industrial.

3.5. Ejes de actividad parciales en diseño de sistemas y programación

INTRODUCCIÓN

- Conocer el concepto de programación estructurada y estructuras privilegiadas así como el núcleo de “pseudocódigo asociado”. Saber justificar sus ventajas.
- Como aplicación, conocer los elementos de la estructura de un lenguaje concreto. En particular, tipos de variables.
- Saber diseñar mediante pseudocódigo algoritmos secuenciales, de selección e iteración.

ESTRUCTURA SECUENCIAL

- Saber escribir un programa secuencial elemental para un cálculo muy simple que incluya uso de funciones matemáticas estándar. Saber utilizar los criterios de prioridad de evaluación de expresiones.
- Saber escribir un programa secuencial sencillo que utilice variables lógicas.
- Saber escribir programas sencillos que utilicen funciones de tratamiento de textos.
- Saber escribir programas sencillos que utilicen funciones de conversión numérico-enumerado (en particular, caracteres).
- Saber escribir un programa sencillo de salida, con diferentes formatos (contenido interno vs. visión externa).
- Escribir programas sencillos para visualizar pérdida de precisión interna. Escribir programas sencillos para visualizar “*overflow*” de enteros. Saber justificarlo.
- Escribir programas sencillos para visualizar *Underflow/Overflow* de datos con decimales. Comentar implicaciones en la práctica.
- Entender la entrada/salida como elemento independiente del tratamiento (visión secuencial y “*event-oriented*” o programación visual).

ESTRUCTURAS de SELECCIÓN

- Saber escribir un programa sencillo que implique una decisión (estructura alternativa o selección simple). Por ejemplo: ver si una persona tienen una edad superior a un mínimo.

- Saber escribir programas sencillos que impliquen una decisión doble. Por ejemplo, edad en un intervalo, etc.). Saber hacerlo con expresiones lógicas y con variables lógicas.
- Saber utilizar estructuras anidadas excluyentes y no excluyentes. Conocer y saber aplicar la estructura de selección múltiple, en opción.
- Saber resolver un problema sencillo pero “completo” y real con estructura alternativa. Por ejemplo, ecuación de segundo grado o ¿En qué cuadrante está una coordenada?.
- Ser capaz de utilizar el concepto de diseño por refinamientos en casos algo más complejos.

ESTRUCTURAS ITERATIVAS

- Casos como calcular el cero de una función en un intervalo, o calcular suma de n naturales o Imprimir la serie de Fibonacci.

PROGRAMACIÓN por AUTÓMATAS

- Validación de un entero, de una fecha o un *float*.

TIPOS ENUMERADOS

- Conocer el concepto de tipo enumerado.
- Entrada/salida de tipos enumerados.

EJERCICIOS de CONSOLIDACIÓN

- Resolver ejercicios de consolidación. Especial énfasis en la necesidad de tratamiento de errores, aunque no se profundice (o apenas se cite) en el tema de gestión de excepciones.

CONOCER el CONCEPTO de ACCIÓN (PROCEDURE y/o FUNCIÓN).

- Entender su justificación inicial como “secretario” y/o minimización de código.
- Saber realizar especificación de acciones con precondiciones y post-condiciones, siquiera a nivel intuitivo.
- Saber resolver ejemplos simples como área corona circular, convertir (x,y) a (m,arg) , convertir (m,arg) a (x,y) , validación edad, validación como real, etc.
- Entender la importancia para la ingeniería del software de la reutilización de acciones “preprogramadas” (método y función).

VARIABLES ESTRUCTURADAS

- Modelización de estructuras de datos más complejas (tipos estructurados homogéneos y heterogéneos) en “papel” sin relación con la informática.

- Diseño de una estructura de datos homogénea sencilla y resolución de un problema con ordenador. Por ejemplo, en el caso de notas de un examen, cálculo de valor medio, máximo y mínimo. Recorrido de secuencias.
- Diseño de representación de un polinomio. Grado máximo y grado conocido. Aplicación a una operación (como suma de polinomios).
- Diseñar la estructura de datos para representación de un sólido con “ n ” masas puntuales. Calcular el c.d.g del sólido.

ACCIONES con VARIABLES ESTRUCTURADAS

- Definir ACCIONES con estructuras de datos complejas. Comenzar con algún ejercicio como eje parcial y pasar a tratar el caso de eje de actividad global.

ESTRUCTURAS EXTERNAS

- Plantear el problema del eje de actividad sobre estructuras externas con accesos secuencial y directo.
- Replantear el eje de actividad global a la vista de la disponibilidad de estas estructuras externas. Ventajas. Necesidad de seguridades.

3.6. Eje de actividad global y estructura del curso

Supondremos el curso organizado en 15 semanas. Esto sería mucho suponer con el marco actual pero debemos considerar que hablamos de dedicación de alumno (no de clase). Considerando una media de 8 horas semana de dedicación del alumno, obtendríamos las 120 horas que corresponden, aproximadamente, a los 4,5 créditos ECTS asignados por los libros blancos que hemos tomado de referencia.

En estas semanas la dedicación no tiene porque ser homogénea pero pensamos que es recomendable que haya unas horas en horario fijo (por ejemplo 2 de teoría y 2 de laboratorio) y que se reserven, asimismo, un bloque de horas (idealmente 2) para asegurar la viabilidad de las actividades de trabajo en grupo.

El eje de actividad global es un elemento esencial en el diseño de la estructura del curso.

En la Figura 1 se incluye una tabla con una representación del resultado de la propuesta.

El eje de actividad, lógicamente, sería un ejercicio “integrador” de los ejes parciales.

S1	Adquirir conocimientos Generales sobre informática (introducción y aspectos culturales)		
S2	Conocer alternativas de representación de la información, saber aplicarlas y sacar conclusiones (costes de capacidad, tiempo de ejecución, problemas de diseño, etc.)		
S3	Conocer el concepto de ALGORÍTMICA, programación estructurada y pseudocódigo estructurado. Diseño de algoritmos con pseudocódigo estructurado. Conocer y saber aplicar el concepto de diseño por refinamientos.	Trabajos de "verificación" en laboratorio de los ejes de actividad parciales Trabajo cooperativo EJE de actividad global, como referencia	
S4	Resolución de mini-problemas secuenciales con un ordenador. Exigirá conocer características básicas y sintaxis de un lenguaje de programación <ul style="list-style-type: none"> - Énfasis en prioridades de evaluación de expresiones - Énfasis en "expresiones" de resultado "lógico" - Énfasis en funciones estándar. En particular, aparte la matemáticas, incidir en funciones de tratamiento de cadenas y de conversión de tipo. - Énfasis en presentar la <i>interface</i> hombre-máquina como una acción independiente del núcleo del algoritmo. Necesidad de validación. - Utilización de módulos simples prediseñados (funciones estándar y módulos "ad-hoc") - Aprender a modificar alguno de los módulos "ad-hoc" simples, en forma intuitiva. 		
S5	Resolución de mini-problemas que incluyan estructuras de selección con un ordenador. Selección múltiple.		
S6	Resolución de mini-problemas de tipo iterativo con un ordenador.		
S7	Resolución de un problema completo con programación modular. Saber justificar su necesidad, conocer la evolución histórica. Saber especificar y diseñar módulos. Entender el concepto de ocultación de información como criterio-base del diseño con módulos. Montaje de un conjunto simple en forma cooperativa, utilizando unidades de compilación separada. Entender la importancia de la reutilización en ingeniería del software.		
S8	Saber modelizar la realidad mediante el diseño de estructuras de datos homogéneas (por ejemplo, notas de un examen, o un polinomio) y aplicarlo a la solución de problemas con ordenador.		
S9	Utilizar programación modular en la resolución de los problemas anteriores que incluyen modelización de datos con variables estructuradas homogéneas.		
S10	Saber ampliar la modelización al caso de estructuras heterogéneas y mixtas de varios niveles (versión mejorada de un polinomio, o un sólido representado con masas concentradas u ocupación de un aparcamiento) y aplicarlo a la resolución de un problema con ordenador.		
S11	Saber aplicar al eje de actividad los conceptos de especificación y DISEÑO de SISTEMAS. Saber tomar partido en discusión entre diseño modular y diseño por refinamientos. Reutilización, la palabra clave.		
S12	Resolución de problemas con utilización de estructuras secuenciales externas persistentes (Ficheros I)		
S13	Resolución de problemas con utilización de estructuras secuenciales, y/o acceso directo, externas persistentes (Ficheros II)		
S14	Ampliación conocimientos (básicamente SO)		Completar EJE de ACTIVIDAD GLOBAL
S15	Ampliación conocimientos(básicamente SO)		Completar EJE de ACTIVIDAD GLOBAL

Figura 1. Estructura cronológica del curso por semanas con indicación de los ejes de actividad parciales

4. A modo de conclusiones

Hemos dedicado el trabajo a compartir nuestra preocupación en relación a qué hacer para afrontar el proceso de diseño curricular en el marco EEES.

El marco de Bolonia es un crisol en el que confluyen diferentes conceptos y elementos. Pero para que la pieza fundida disponga de suficiente calidad es necesario manejar con esmero cada componente. Casi siempre innovar es, sin duda, mejor que no hacerlo, pero apuntarse a una moda, sin más, puede ser incluso peor que el inmovilismo. Además, los criterios a aplicar han de ser simples: si no, ¡no se usarán!

Hemos querido buscar elementos comunes en nuestras experiencias pasadas para poder construir de manera novedosa, pero simple, sólida y coherente, el proyecto de futuro.

Hemos detectado un elemento que pensamos puede actuar de catalizador del éxito del nuevo marco: se trata del eje de actividad y los ejes de actividad parcial. También hemos recogido una metodología para identificarlos.

Finalmente, hemos aplicado los conceptos a una propuesta de estructura curricular para Fundamentos de Informática, en los nuevos títulos de grado en ingenierías no informáticas (en particular, las industriales). Su generalización a otras materias y titulaciones nos parece inmediata.

Referencias

- [1] ANECA. http://www.aneca.es/modal_eval/convener_docs_titulos.html <http://www.aneca.es/publicaciones/publicaciones.html>
- [2] ANECA. *CD con los capítulos del libro blanco del grado de diferentes ingenierías.*
- [3] Bloom, 1956]. Bloom, B.S. et al., 1956. *Taxonomy of educational objectives*. Vol 1: The cognitive domain. New York. Mc Kay
- [4] [Bolonia process] *Main Documents of the Bologna Process*. http://www.bologna-bergen2005.no/EN/MAIN_DOC/MAIN1.HTM.
- [5] *Bologna Secretariat Website* <http://www.dfes.gov.uk/bologna/>
- [6] [Buendía et al., 2004] Félix Buendía, Juan Carlos Cano, Julio Sahuquillo, Jean-Luis Posadas, Juan-Miquel Martínez, José-Vicente Benlloch. *Un modelo de diseño curricular de Informática orientado a la obtención de competencias*. Libro de actas JENUI 2004. Ed. Thomson. Julio 2004. Pag. 89 a 94.
- [7] Fontela Romero, Oscar; Hernández Pereira, Elena M. *Adaptación de la asignatura Fundamentos de Informática de la Ingeniería Técnica Industrial al espacio Europeo de Educación superior*. Libro de actas Jenui 2005. Thomson, 2005 (pag 463 a 470).
- [8] García García, María Jose; Lara Bercial, Pedro J.; Gaya López, María Cruz. *Una asignatura "a la Boloñesa"*. Libro de actas Jenui 2005. Thomson, 2005 (pag 479 a 485).
- [9] [Ibáñez et al., 2001] Jesús Ibáñez, Julián Gutierrez, Jon Ander Elorriaga y Alfredo Goñi. *El Congreso de alumnos como Recurso didáctico*. VII Jornadas de la enseñanza universitaria de la informática JENUI'2001. Palma de Mallorca, julio 2001. Pag 126 a 131
- [10] [Kolb, 1971]. Kolb, David A. *Individual learning styles & the learning process*. Sloan School of management. WP535-71
- [11] [Navarro et al. 2000]. Juan J. Navarro, Miguel Valero-García, Fermín Sanchez y Jordi Tubella. *Formulación de los objetivos de una asignatura en tres niveles jerárquicos*. VI Jornadas de la enseñanza universitaria de la informática JENUI'2000. Alcalá de Henares, Septiembre 2000. Pag 457 a 462.
- [12] [Piña, 2004]. Bartolomé Piña. *Blended learning*. Conceptos básicos. <http://www.sav.us/es/pixelbit/articulos/n23/n23art/art2301.htm>
- [13] [Valero & Navarro, 2001] Miguel Valero-García y Juan J. Navarro. *Niveles de competencia de los objetivos formativos en las ingenierías*. VII Jornadas de la enseñanza universitaria de la informática JENUI'2001. Palma de Mallorca, julio 2001. Pag 149 a 154
- [14] [Virgós 2004-1] Ferran Virgós Bel. *La función TSI en las organizaciones: una evaluación formativa para la detección de "gaps" críticos*. Actas de JENUI 2004. Alicante, julio 2004. Pag. 79 a 87
- [15] [Virgós, 2004-2]. Ferran Virgós Bel. *Algunas técnicas didácticas a considerar para el proceso enseñanza-aprendizaje en el marco de Bolonia*. Actas CUIEET. Julio 2004
- [16] [Virgós, 2005] Ferran Virgós Bel. *Elementos a considerar en el diseño curricular del nuevo grado en informática*. Actas de JENUI 2005 Thomson, pag. 155 a 162.