

## Implementación de algoritmos en hardware: de la puerta NAND al bucle FOR

Javier Díaz Alonso

Departamento de Arquitectura y Tecnología de Computadores  
E.T.S.I.I.T., Universidad de Granada, España

[jda@ugr.es](mailto:jda@ugr.es)

**Resumen.** En este artículo se describe el trabajo a realizar en la asignatura de Implementación de algoritmos Hardware dentro de las titulaciones de grado en Ingeniería Informática (especialidad de Ingeniería de Computadores) y de grado en Ingeniería de Tecnologías de Telecomunicación de la Universidad de Granada. Motivaremos y contextualizaremos la asignatura, explicando a los estudiantes por qué este conocimiento es útil y conveniente como parte de su especialización. Describiremos los contenidos a impartir, mostrando que el diseño de sistemas hardware puede hacerse a diferentes niveles de abstracción y, particularmente, usando lenguajes de programación convencionales. Finalmente mostraremos la metodología didáctica y métodos de evaluación propuestos, completando con ello la descripción global de esta asignatura.

**Palabras Clave:** Docencia en Ingeniería de Computadores, diseño de Hardware, SoC, Aprendizaje basado en proyectos.

**Abstract.** This article describes the work to be done in the course Hardware Implementation of algorithms within undergraduate degrees in Computer Science (Computer Engineering specialization) and Telecommunications Technology Engineering at the University of Granada. We motivate and contextualize the subject, explaining to students why this knowledge is useful and appropriate as part of their specialization. We describe the contents to impart, showing that hardware system design can be made from different attraction levels and particularly, using conventional programming languages. Finally we show teaching the methodology and evaluation methods proposed, thereby completing the comprehensive description of this course

**Keywords:** Computer engineering teaching, hardware design, SoC, project-based learning

## 1 Introducción

*“El hardware es muy complicado. Requiere conocer muchos detalles, la mayoría de muy bajo nivel (voltajes e interfaces eléctricos, temporización de señales, principios físicos, etc.). Además nosotros no podemos fabricar chips (saldría muy caro) e incluso comprarlos para hacer un placa de circuito impreso requiere inversión o acceso a maquinaria difícil de obtener. Por todo ello la mejor opción es comprar hardware comercial. Este debe ser flexible y escalable en prestaciones de forma tal que nuestro objetivo sea su personalización mediante la programación del software asociado.”*

Las frases anteriores bien podrían describir la opinión de muchos estudiantes de las titulaciones de grado en Ingeniería Informática (especialidad de Ingeniería de Computadores) y de grado en Ingeniería de Tecnologías de Telecomunicación de la Universidad de Granada. El hardware se antoja como algo “feo y complejo”. Operaciones que hacemos todos los días como leer un valor del teclado o una memoria USB requieren horas sino semanas de trabajo para conseguir estos objetivos si pensamos en diseñar el hardware asociado. Si a esto le añadimos la confusión de que el termino hardware puede representar muchos conceptos diferentes (chips, tarjetas PCBs, elementos mecánicos, etc..) y que muchos autores de textos docentes no ayudan demasiado (ver por ejemplo [1] donde se dice que “los sistemas operativos ocultan el hardware feo con abstracciones hermosas”), se crea el clima perfecto para desmotivar al estudiante a profundizar en estos contenidos, situación que no es fácil de superar.

Pero, ¿qué ocurriría si el hardware se pudiera describir mediante programación en C/C++? ¿Y si este pudiera cambiarse, programarse, de la misma forma que se hace con el software convencional? ¿Y si la división, la frontera entre hardware y software hubiera, hoy en día, desaparecido? ¿Y si en vez de semanas pudiéramos tener sistemas hardware complejos en menos de 1 hora? Además, ¿qué hacemos cuando los miles de millones de transistores que traen los dispositivos de cómputo actuales no son suficientes para cumplir con los requisitos de nuestro proyecto?

En la asignatura de Implementación de Algoritmos en Hardware (IAH) tratamos de dar respuesta a estas preguntas y otras relacionadas. Centrándonos principalmente en el campo de los sistemas empotrados, trabajamos en el análisis y diseño hardware entendiendo como tal los elementos de computación y control existentes en un chip, típicamente programable como por ejemplo una FPGA. Mostramos que simplemente usando tarjetas para desarrollo de prototipos, podemos diseñar sistemas empotrados complejos sin más que reprogramar la funcionalidad de un único chip, diseñando/modificando los elementos software o hardware del sistema, eliminando la férrea distinción que existía entre ambos aspecto en el pasado. Como resultado, veremos que una buena división de los elementos hardware y software del sistema permite conseguir aumentos significativos de prestaciones o por ejemplo reducciones de consumo considerables, aspectos claves para el desarrollo de los sistemas empotrados actuales.

La asignatura de IAH parte de conocimientos generales del estudiante sobre sistemas digitales y la estructura/arquitectura de computadores. A partir de ellos veremos que es posible no sólo desarrollar nuevos sistemas empotrados de altas prestaciones para problemas de uso específico, sino que también, gracias a este conocimiento podemos programar los procesadores sacando máximo provecho de los potentes elementos hardware incluidos actualmente en ellos.

A lo largo de la asignatura utilizamos el concepto de System-on-Chip (SoC) y su presencia en la vida cotidiana en múltiples dispositivos como teléfonos, tabletas, routers de comunicaciones o cámaras de videovigilancia. Del análisis de estos dispositivos podemos identificar la gran variedad de elementos hardware que contienen los procesadores empotrados actuales, ya sean módulos de comunicaciones, coprocesadores multimedia, coprocesadores aceleradores o controladores de entrada/salida. Cada uno de ellos da respuesta a la solución de un problema específico y el conocimiento de éstos es vital para entender el desarrollo de los sistemas actuales.

Como conclusión, ¿qué aporta a los estudiantes cursar esta asignatura? ¿Para qué el esfuerzo? La realización de esta asignatura capacita a los alumnos a desarrollar eficientemente sistemas empotrados donde la división entre software y hardware sea realizada de forma óptima. Permite seleccionar la plataforma empotrada comercial más adecuada a nuestras especificaciones, de forma que minimicemos los costosos ciclos de rediseño debido a una mala selección de elementos del sistema. Finalmente, el conocimiento adquirido nos permite sacar el mejor provecho de los procesadores empotrados presentes en dispositivos multimedia (tabletas, teléfonos), en sensores inalámbricos y de red, en dispositivos de automoción, aviónica, aeroespacial, etc..., aumentando significativamente gracias a todos estos conocimientos nuestra especialización y valor como ingeniero en el campo de los sistemas empotrados.

## **2 Implementación de algoritmos en hardware en el contexto de la titulación**

La asignatura de IAH se imparte como optativa durante el 4º curso del grado en Ingeniería Informática (especialidad de Ingeniería de Computadores). Además, se ofrece como optativa de otras titulaciones del grado en Ingeniería de Tecnologías de Telecomunicación de la Universidad de Granada.

Tal y como se establece en la memoria Verifica del título de Ingeniería Informática de la Universidad de Granada [2], junto con la asignatura del Tecnologías emergentes pertenece a los Complementos de Sistemas de Cómputo para Aplicaciones Específicas y tiene una carga lectiva de 6 créditos ECTS.

Como se desprende de los objetivos formativos descritos en la tabla 1 y de las competencias descritas en la tabla 2, la asignatura pretende capacitar al estudiante en las tareas de análisis y diseño de elementos hardware de un sistema de computación, así como ser capaz de evaluar y seleccionar los elementos existentes. Es por ello, que el enfoque de la asignatura sigue un planteamiento global, evaluando los posibles soluciones de un problema dado de forma integral, seleccionando un particionamiento software/hardware que permita conseguir la solución de diseño óptima para un

problema teniendo en cuenta restricciones de prestaciones, potencia, coste, fiabilidad o tiempos de desarrollo.

Destaquemos que en los grupos de estudiantes de esta asignatura es posible la participación de alumnos de distintos grados, informática y telecomunicaciones. Lejos de ser un problema y como se expondrá en la sección 4, el enfoque metodológico variado, con grupos de trabajo y aprendizaje basado en proyectos puede/debe beneficiarse significativamente de esta diversidad de experiencias y permitir encontrar/resolver problemas de naturaleza más interdisciplinar.

En lo referente a su relación con otras asignaturas, un diagrama de bloques que muestra su relación con otras asignaturas de la memoria del grado en Ingeniería Informática es mostrado en la Figura 1.

**Tabla 1. Objetivos formativos de IAH.**

- Describir y aplicar metodologías de diseño de circuitos digitales para sistemas de uso específico.
- Aplicar técnicas de análisis de prestaciones (precisión, recursos y potencia de cómputo) para la identificación de elementos que limitan el sistema.
- Identificar las herramientas y metodologías de diseño más adecuadas según la aplicación y sus especificaciones.
- Conocer técnicas de diseño de módulos hardware de altas prestaciones (coprocesadores o periféricos) para sistemas integrados.
- Aplicar técnicas de máquinas de estado algorítmicas.
- Conocer técnicas de diseño de circuitos de operaciones matemáticas complejas. Elegir el tipo de representación de los datos acorde a las especificaciones de precisión, recursos y prestaciones.
- Identificar y reconocer las necesidades funcionales que justifican el desarrollo de hardware de propósito específico en distintos campos de aplicación.
- Aplicar metodologías de adaptación de algoritmos para su implementación en hardware de propósito específico.
- Aplicar técnicas avanzadas de compartición de recursos y optimización de prestaciones (segmentación de cauce, diseño superescalar, etc.).
- Evaluar costes y prestaciones de módulos hardware para aplicaciones específicas.
- Identificar dispositivos y plataformas integradas para aplicaciones específicas, sus restricciones y campos de aplicación

**Tabla 2. Competencias de la asignatura IAH**

- **IC1.** Capacidad de **diseñar y construir sistemas digitales**, incluyendo computadores, sistemas basados en microprocesador y sistemas de comunicaciones.
- **IC2.** Capacidad de desarrollar **procesadores específicos y sistemas empotrados**, así como desarrollar y **optimizar el software** de dichos sistemas.
- **IC5.** Capacidad de analizar, evaluar y seleccionar las plataformas hardware y software más adecuadas para el soporte de aplicaciones empotradas y de tiempo real.
- **E4.** Capacidad para **definir, evaluar y seleccionar plataformas hardware** y software para el desarrollo y la ejecución de sistemas, servicios y aplicaciones informáticas.
- **E9.** Capacidad para **resolver problemas con iniciativa, toma de decisiones, autonomía y creatividad**. Capacidad para saber comunicar y transmitir los conocimientos, habilidades y destrezas de la profesión de Ingeniero Técnico en Informática.

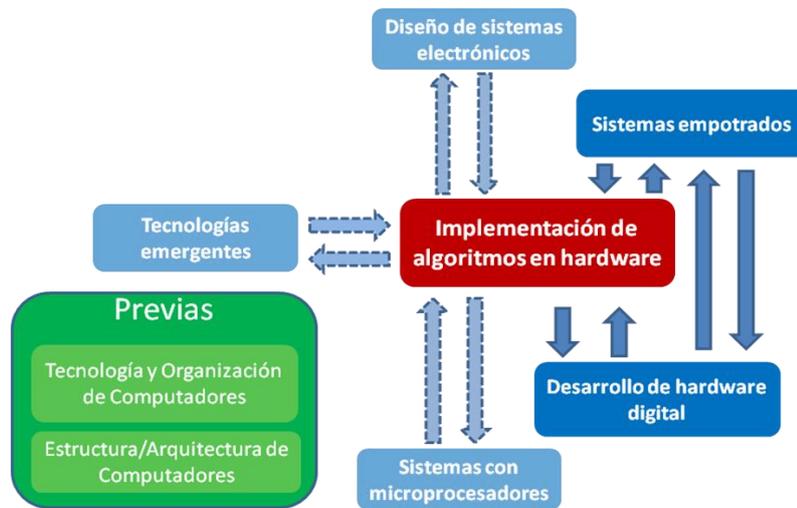
La asignatura requiere conocimientos básicos sobre sistemas digitales y estructura de los sistemas de computación. Estos conocimientos son parte los contenidos tratados en la asignaturas como Tecnología y Organización de Computadores, Estructura de Computadores y Arquitectura de Computadores

Ya en el contexto de la especialidad en Ingeniería de Computadores, la asignatura tiene una fuerte relación con las obligatorias de la especialidad de Sistemas empotrados y Desarrollo de hardware digital. La primera supone un complemento natural a los contenidos aquí expuestos al tratar, entre otros, los contenidos necesarios para desarrollar un los elementos software de programación de los procesadores desde la programación de bajo nivel hasta el nivel de sistema operativo. En lo referente a la asignatura de Desarrollo de hardware digital, ésta proporciona un puente entre los conocimientos de tecnología de computadores básicos y los conocimientos abordados en esta asignatura. De esta forma la conexión entre los distintos niveles de descripción de un sistema digital está completa.

Cabe por último destacar otras asignatura de la especialidad de Ingeniería de Computadores que, si bien no tienen una conexión tan directa, se benefician/complementan los contenidos aquí expuestos. Por ejemplo en la asignatura de Sistemas con microprocesadores se centra en la descripción y programación de un subconjunto de los elementos que veremos en IAH. Tecnologías emergentes describe numerosos sistemas de aplicaciones específicas donde la se hace vital el desarrollo de elementos hardware a medida tal y como los descritos en esta asignatura. Finalmente, en la asignatura de diseño de sistemas electrónicos se describen a bajo nivel los componentes electrónicos utilizados para el diseño de los bloques descritos en IAH.

De forma similar a lo descrito para el grado de Ingeniería Informática, en el grado en Ingeniería de Tecnologías de Telecomunicación, de forma destacada en el marco

de la especialidad de sistemas electrónicos, la asignatura IAH aporta una visión de alto nivel y aplicada de los bloques básicos analizados en otras asignaturas de la especialidad. De esta forma la asignatura proporciona en este grado una visión aplicada y funcional de los elementos hardware que se utilizan en el diseño de un sistema empotrado. Como se explica en la sección 3, la utilización de lenguajes de descripción hardware (HDLs) y de dispositivos reconfigurables (FPGAs) proporcionan una notable flexibilidad para el diseño rápido de sistemas electrónicos y, en concreto, para la implementación de algoritmos en hardware de propósito específico.



**Figura 1.** Relación de IAH con otras asignaturas del grado de Ingeniería Informática.

### 3 Contenidos de la asignatura Implementación de algoritmos en hardware

Los objetivos y competencias descritos en la tablas 1 y 2 toman forma con una selección de contenidos que es la descrita en la tabla 3. En lo referente a los contenidos de teoría, la asignatura comienza con un tema de contenidos generales para luego tener los temas 2 y 3 dedicados a los contenidos más propiamente relacionados con el cuerpo técnico de la materia. El 4 es un tema de carácter más aplicado y divulgador y que pretender conectar los contenidos más teóricos descritos en los temas anteriores con contenidos de carácter práctico de la asignatura, analizando para ello sistemas comerciales o de investigación existentes y/o en desarrollo actualmente.

**Tabla 3. Contenidos de la asignatura Implementación de algoritmos en hardware**

**TEORÍA:**

- **Tema 1: Introducción y conceptos generales**
  - 1.1 Motivación y objetivos de la implementación de algoritmos en hardware.
  - 1.2 Tecnologías y dispositivos.
  - 1.3 Conceptos generales (niveles de abstracción, síntesis automática, módulos IP, SoC, NoC, codiseño hardware-software).
  - 1.4 Lenguajes de descripción y herramientas de diseño.
- **Tema 2: Metodologías de diseño e implementación de algoritmos en hardware**
  - 2.1 Sistemas autónomos, periféricos y coprocesadores.
  - 2.2 Diseño a nivel de sistema. SoC y particionamiento hardware software. Metodología de diseño basada en plataformas..
  - 2.3 Diseño RTL. Diseño basado en máquinas algorítmicas.
  - 2.4 Tipos de aritméticas para implementación de algoritmos.
  - 2.5 Compromisos de diseño: consumo de recursos, prestaciones, precisión, potencia, coste, tiempo de diseño.
  - 2.6 Técnicas de diseño y optimización: compartición de recursos, planificación temporal, segmentación de cauce, diseño superescalar, diseño para bajo consumo.
- **Tema 3: Hardware para operaciones específicas**
  - 3.1. Técnicas de diseño de funciones aritméticas y trigonométricas.
  - 3.2. Controladores de memoria, buses, E/S.
  - 3.3. Arquitecturas de procesamiento SIMD/MIMU. Arquitecturas basadas en multicores/NoCs.
  - 3.4. Otras arquitecturas de interés (registros LFSR, procesadores sistólicos, etc..).
- **Tema 4: Ejemplos de sistemas y aplicaciones.**
  - 4.1 Identificación de requisitos de hardware de propósito específico en aplicaciones de comunicaciones, procesamiento de imágenes y video, audio, control y criptografía. 5.1 Aplicaciones para procesamiento de audio.
  - 4.2. Ejemplos para procesamiento de imágenes y video.
  - 4.3 Ejemplos para comunicaciones.
  - 4.4 Ejemplos para control.
  - 4.5 Ejemplos para criptografía.
  - 4.6. Ejemplos para supercomputación y aceleración de procesos.

**SEMINARIOS:**

- **Seminario práctico 1:** Introducción al entorno de desarrollo y lenguaje VHDL.
- **Seminario práctico 2:** Implementación de algoritmos en aritmética en punto fijo: recursos, precisión y prestaciones.
- **Seminario práctico 3-4:** Introducción a los lenguajes de descripción de hardware basados en C/C++.
- **Seminarios prácticos 5:** Ejemplos de arquitecturas para comunicaciones, procesamiento de audio/video y criptografía.

## PRÁCTICAS

- **Practica 1:** Diseño e instanciación de módulos IP para aceleración de operaciones matemáticas/procesamiento digital de señales.
- **Practica 2:** Compresión JPEG. Análisis de arquitecturas hardware. Estudio de las diferentes arquitecturas, estrategias de diseño y su impacto en el diseño resultante.
- **Practica 3.** Implementación en hardware de un algoritmo para aplicaciones específicas.

Respecto a la bibliografía para estos contenidos de teoría, dado que la asignatura requiere una actualización continua de los mismos para reflejar las últimas tendencias existentes en la industria y la academia, buena parte de ellos serán obtenidos de páginas webs de compañías y revistas de carácter industrial. No obstante, para una revisión de los contenidos más clásicos o de índole más teórica, recomendamos la consulta de los trabajos siguientes [4, 5, 6, 8].

La bibliografía de contenidos de índole más aplicada, seminarios y prácticas, será obtenida principalmente de sitios web de proveedores de herramientas, chips, tarjetas, etc.. Destacamos para ello las referencias [9, 10, 11, 12, 13, 14] de donde pueden obtenerse buena parte del material que utilizamos así como información actualizada de las últimas tendencias en la temática.

Como resultado de este enfoque bibliográfico actualizado y de la metodología docente descrita en la sección 4, aproximadamente un 50-60% de los contenidos impartidos son fijos, siendo un 10-20% dependiente de las tendencias actuales de la tecnología y aproximadamente un 30% se determinan cada año en base a los intereses del alumnado matriculado en ese curso.

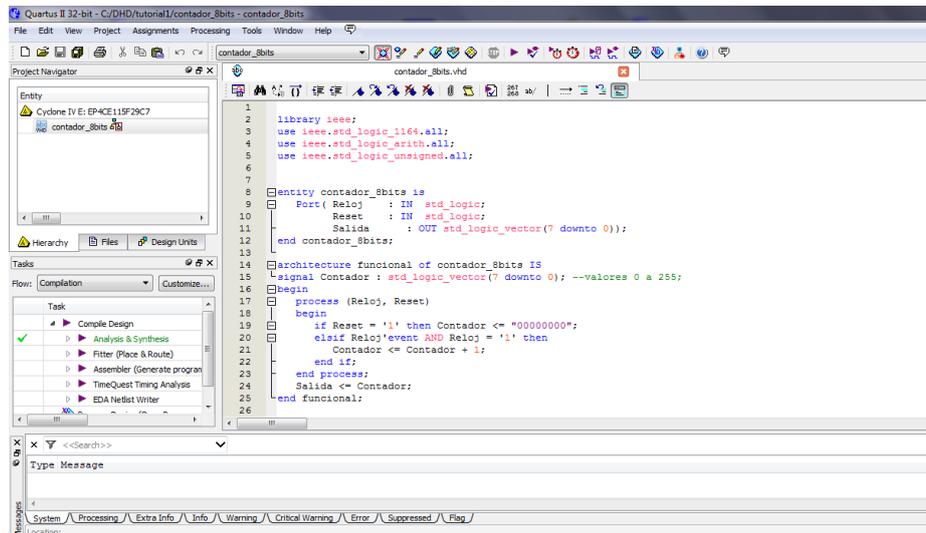


Figura 2. Entorno de trabajo Quartus II de Altera.

Finalmente, en lo referente a los elementos a utilizar en clase, destacar el uso del entorno de programación Quartus II de Altera y la tarjeta DE2-115 [9]. Las imágenes de estos elementos se muestran en las figuras 2 y 3. Como se aprecia en la Figura 3, la tarjeta FPGA utilizada consta de numerosas interfaces y periféricos lo que nos permite desarrollar una gran cantidad de proyectos usando la misma tarjeta.



**Figura 3.** Tarjeta de desarrollo de prototipos, DE2-115 de Altera.

#### **4 Secuenciación temporal, metodología y evaluación**

En los apartados anteriores hemos mostrado los objetivos, competencias y contenidos de la asignatura IAH. De forma simplificada y como resumen podemos decir que la asignatura persigue tres objetivos fundamentales que son:

a) Repasar, dar una visión homogénea a los estudiantes de conceptos básicos relacionados con IAH. Esto se consigue mediante los el tema 1 de teoría y el seminario y práctica 1.

b) Desarrollar los objetivos y competencias de la asignatura. Esto es tratado en los temas 2 y 3, en los seminarios 2-4 y en la práctica 2.

c) Conectar el conocimiento adquirido con los sistemas disponibles actualmente para capacitar al ingeniero en formación en el desarrollo autónomo de los sistemas estudiados y/o en la selección de los mismos. Estos apartados son cubiertos en el tema 4 de teoría, en el seminario 5 y en la práctica 3.

El primer objetivo tiene una duración temporal de 2-3 semanas y se desarrolla con una metodología clásica, con lecciones magistrales pero participativas y con ejercicios muy concretos y específicos.

El segundo objetivo utiliza una metodología mixta basada en lecciones magistrales, aprendizaje basado en proyectos, metodología basad en discusión y metodología

basada en aprendizaje autónomo usando las guías de trabajo autónomo. Tiene una duración temporal de aproximadamente 7-8 semanas.

Finalmente, el último bloque tiene una duración de 4-5 semanas y dota de gran autonomía y poder de decisión a los alumnos. Se centra en las metodologías de aprendizaje basado en proyectos y en el trabajo autónomo del alumno. Esto se materializa mediante la creación de grupos de trabajo que resuelvan en la práctica 3 un problema concreto propuesto por los alumnos y aceptado por el profesor. Además, en los contenidos de teoría se fomentará la participación del alumnado en forma de exposiciones orales y discusiones/debates de artículos de forma que las competencias como IC5 o E9 queden cubiertas. De esta forma, la última parte de la asignatura vendrá determinada por los intereses de los alumnos matriculados y producirá que, como se ha descrito en la sección 3, aproximadamente un 30% de los contenidos sean variables acorde a los intereses de los alumnos.

Esta diversidad de metodologías persigue tanto desarrollar los conocimientos del alumno como sus competencias de carácter práctico. Se complementan con el uso de las tutorías, individuales o grupales como un recurso didáctico más que es planificado para permitir conseguir un correcto resultado de aprendizaje y una buena comunicación profesor-alumno.

Acorde a la diversidad de métodos docentes utilizados, la evaluación docente es también rica en los métodos a utilizar. Se fomenta la evaluación continua de los alumnos. Exámenes, cuestiones en clase, trabajos de exposición oral, realización de prácticas de laboratorio, trabajos libres de carácter práctico, etc.. Muchos métodos son posibles y parte de ellos se concretan dependiendo de los intereses de los alumnos matriculados y previo pacto profesor-alumno. De igual manera, en peso en la evaluación de los diferentes objetos calificables también es pactado entre los alumnos y el profesor. Recordemos que el objetivo de estas evaluaciones es ayudar al proceso de aprendizaje y no el conseguir poner un nota. Por ello, se trata que los estudiantes reciban pronto las valoraciones y que el número de pruebas sea suficiente para informarles del progreso de su aprendizaje.

Con carácter orientativo, un posible esquema de la distribución de la nota en la evaluación podría ser el siguiente:

- Trabajo autónomo y en aula (individual y en grupo) → 30%
- Prácticas guiadas y libres → 40 %
- Pruebas/exámenes → 30%

Se pretende que cada año los alumnos sean parte activa de la evaluación mediante utilización de técnicas de autoevaluación o de evaluación entre iguales. Su uso más o menos intensivo depende del interés de los alumnos y del número de matriculados (si el número supera los 20-25 se hace imprescindible para, sin eliminar pruebas de evaluación, mantener una carga de trabajo del profesor razonable).

Por último destacar que aunque se trata de utilizar una evaluación continua del alumnado, ya sea porque estos no han podido asistir o bien porque no han superado las pruebas realizadas, existe una última prueba consistente en un examen teórico-práctico global para permitir superar la asignatura a aquellos que así lo requieran.

## 5 Conclusiones

Este trabajo ha tratado de describir el trabajo que se desarrolla en la asignatura de Implementación de algoritmos en hardware. Hemos visto una motivación general en la sección 1, los objetivos y competencias en la sección 2, los contenidos en la sección 3 y la secuenciación de los mismos, metodología didáctica y evaluación en la sección 4. Con ello se ha descrito de forma global la asignatura y su forma de impartición.

Es importante destacar que esta asignatura tiene un carácter eminentemente práctico. La utilización de lenguajes de descripción de hardware como VHDL y sobre todo los basados en lenguajes de alto nivel como C/C++ son clave para acercar a los alumnos a esta materia y base de la productividad de los diseños a realizar.

La realización de esta asignatura ayuda a mejorar la capacitación de los estudiantes en un campo en continua expansión como es el de los sistemas empotrados. No sólo es gracias a la mejora de sus capacidades de diseño y análisis de estos sistemas, sino también porque este conocimiento permite una selección adecuada de las plataformas empotradas a utilizar y su programación optimizada, capacidades claves para su futuro ejercicio profesional.

Finalmente concluir incidiendo en a gran libertad de decisión que tienen los alumnos en la asignatura, permitiendo metodologías de trabajo en grupo muy similares a las que utilizarán en la industria durante el ejercicio de su carrera profesional y con posibilidad de seleccionar parte de los contenidos de la asignatura en base a sus propios intereses e inquietudes.

## Referencias

1. Tanenbaum, Andrew S.: Sistemas operativos modernos. 3ed. Prentice Hall, pag. 5 (2003)
2. Memoria Verifica del título de Ingeniería Informática de la Universidad de Granada. Disponible online en el siguiente enlace:  
<http://grados.ugr.es/informatica/pages/infoacademica/archivos/verificaingenieriainformatica/%21>
3. Steve Kilts, “Advanced FPGA Design: Architecture, Implementation, and Optimization”. John Wiley and Sons, 2007.
4. Clive Maxfield, “The design warrior’s guide to FPGAs”, Elsevier 2004.
5. A. Rushton: "VHDL for Logic Synthesis", John Wiley and Sons, 2001.
6. D.L. Perry, "VHDL programming by example", McGraw-Hill, 2002. (Recurso electrónico).
7. J.O. Hamblen, T. S. Hall, M. O. Furman: “Rapid Prototyping of Digital Systems : SOPC Edition”, Springer 2008.
8. J. Bhasker, “A SystemC Primer”, Star Galaxy Publishing (2002).
9. Altera, fabricantes de dispositivos reconfigurables. Online en: <http://www.altera.com>
10. Xilinx, fabricantes de dispositivos reconfigurables. Online en: <http://www.xilinx.com>
11. Principal portal de descargas de código HDL Opencores. Online:  
<http://www.opencores.com>
12. Open hardware repository. Online: <http://www.ohwr.org>
13. Promotores iniciativa SystemC. Online: <http://www.accellera.org/home/>
14. Sitio web sobre empresas, foros, eventos, etc.. Online: <http://www.fpgacentral.com/>