

Formulario de Aprobación Curso de **Posgrado** 2012

Asignatura: Herramientas de programación para procesamiento de señales.

Profesor de la asignatura ¹:

Dr. Pablo Musé, Prof. Adjunto gr. 3, Instituto de Ingeniería Eléctrica.
Msc. Juan Cardelino, Asistente gr. 2, Instituto de Ingeniería Eléctrica.

Profesor Responsable Local ¹:

Otros docentes de la Facultad:

Ing. Haldó Spontón, Ayudante gr. 1, Instituto de Ingeniería Eléctrica.

Docentes fuera de Facultad:

Instituto ó Unidad: Instituto de Ingeniería Eléctrica.
Departamento ó Area: Departamento de procesamiento de señales.

Fecha de inicio y finalización:

Curso semestral, que se dicta en ambos semestres. Su comienzo coincidirá con el principio de cada semestre.

Horario y Salón:

Horarios de consulta a convenir con los docentes.

Horas Presenciales: 25

Nº de Créditos: 6

Público objetivo y Cupos:

El curso está orientado fundamentalmente a estudiantes a punto de comenzar o en la primera etapa de una maestría o doctorado en el área de procesamiento de señales. También puede ser de interés para investigadores interesados en profundizar sus conocimientos de programación científica.

El cupo máximo es de 8 estudiantes y el mínimo de 4.

Objetivos:

Familiarizar a los participantes con herramientas de programación avanzadas, aplicadas al procesamiento de señales en general y de imágenes en particular.
Estudiar en profundidad de algoritmos avanzados de procesamiento de señales e imágenes.

Al aprobar el curso se espera que los estudiantes hayan desarrollado las siguientes habilidades:

- * profundizar los conocimientos de programación en lenguaje C.
- * uso fluido en el uso de herramientas de programación: debuggers, profilers, verificadores de memoria.
- * conocimientos de los algoritmos básicos utilizados en el procesamiento de señales y de las estructuras de datos básicas necesarias para implementar dichos algoritmos.
- * capacidad de afrontar la implementación de un algoritmo de complejidad media/alta tanto desde el punto de vista teórico, como del de la eficiencia computacional.

Desde el punto de vista docente se buscará fomentar además diversas habilidades transversales en los estudiantes, como el estudio autónomo de herramientas, algoritmos y lenguajes de programación. También se hará especial énfasis en la lectura y redacción de publicaciones científicas/técnicas.

Se incentivará el uso y desarrollo de herramientas de software libre y el uso de licencias abiertas para el contenido generado.

Conocimientos previos exigidos:

Programación estructurada, muestreo y procesamiento digital, series y transformadas de Fourier. Nociones básicas de procesamiento de señales. Nociones básicas de procesamiento de imágenes. Nociones básicas de estructuras de datos.

Conocimientos previos recomendados:

Programación orientada a objetos.

Metodología de enseñanza:

Este curso se dicta en modalidad de taller, con una carga importante de trabajo personal fuera de clase, en relación a las horas de laboratorio. Los estudiantes tendrán una reunión semanal con los docentes, la cual tendrá una dinámica estilo laboratorio, en la cual se presentan brevemente los conceptos a desarrollar durante la semana, y donde se marca una serie de actividades dirigidas a profundizar esos conceptos. Estas actividades constan del estudio autónomo de los temas planteados y de la resolución de ejercicios teóricos y de programación. Asimismo, se prevén consultas periódicas con los docentes de la asignatura para evacuar las dudas surgidas en la semana.

Una vez concluidas las actividades programadas, se presentará por parte de los docentes un algoritmo clásico de señales/imágenes que presente dificultades considerables, tanto desde el punto de vista teórico como práctico.

Se dará una guía sobre como implementar dicho algoritmo y se seguirá el desarrollo del mismo por parte de los estudiantes. Los estudiantes deberán presentar un informe escrito donde documenten la implementación del algoritmo.

Finalmente, los estudiantes deberán implementar un algoritmo de un tema a elección dentro del área de interés de los estudiantes. El algoritmo a elegir deberá estar comprendido en alguna de las siguientes áreas: procesamiento de imágenes, audio, señales biomédicas o algoritmos generales de señales. La propuesta de los estudiantes deberá ser validada por los docentes de la asignatura.

El cronograma del curso es el siguiente: los primeros 5 módulos de ejercicios se realizarán a razón de 1 por semana, al finalizar los mismos se dedicarán 3 semanas al estudio e implementación del algoritmo guiado y finalmente se dedicarán 5 semanas a la implementación del proyecto final.

Las primeras 7 semanas de clase se realizarán sesiones de laboratorio de 2hs cada una, donde se realizarán las actividades anteriormente descritas, correspondientes al los primeros 5 módulos y el algoritmo guiado.

Durante las 7 semanas finales, los estudiantes trabajan de forma autónoma en el proyecto final, teniendo 1 hora de consulta semanal para evacuar dudas.

Las horas de clase del curso se distribuyen de la siguiente manera:

- Horas clase (teórico): 0
- Horas clase (práctico): 0
- Horas clase (laboratorio): 14
- Horas consulta: 7
- Horas evaluación: 4
 - Subtotal horas presenciales: 25
- Horas estudio: 30
- Horas resolución ejercicios/prácticos: 0
- Horas proyecto final/monografía: 35

o Total de horas de dedicación del estudiante: 90

Forma de evaluación:

Se realizará un informe escrito del algoritmo guiado desarrollado y un proyecto final de aplicación libre que involucra la implementación de un algoritmo a elección. Se deberá entregar un informe escrito del algoritmo desarrollado y realizar una presentación oral del mismo.

El trabajo final se realizará en grupos de 2 estudiantes (excepcionalmente 3).

El curso no tiene examen ni parciales ni ganancia del curso, se aprueba o se reprueba.

La nota de aprobación del curso será un promedio ponderado del informe del algoritmo guiado, el informe del algoritmo libre y la presentación oral.

Temario:

- Parte 1: Nociones generales de programación
 - Módulo 1: Introducción al lenguaje C.
 - Módulo 2: Punteros, arreglos y estructuras. Memoria dinámica.
 - Módulo 3: Diseño de un filtro FIR.
 - Módulo 4: Herramientas de programación: Entornos de desarrollo. Sistemas automáticos de compilación, Debugger, Profiler, Documentación automática, Compilación multiplataforma, testing.
 - Parte 2: Aplicaciones al procesamiento de señales/imágenes
 - Módulo 5: Lenguaje C aplicado al procesamiento de imágenes.
 - Módulo 6: Implementación guiada de un algoritmo.
 - Módulo 7: Proyecto final de implementación libre de un algoritmo.
-

Bibliografía:

General

•wiki del curso: <http://ie.fing.edu.uy/rs/wiki/ipol:main>

Programación en C

- ▲ Deitel & Deitel. How to Program in C/C++.
- ▲ C Programming Language (2nd Edition). Brian W. Kernighan, Dennis M. Ritchie.
- ▲ Programación en C (Wikibook). [http://es.wikibooks.org/wiki/Programación en C](http://es.wikibooks.org/wiki/Programación_en_C). (visto 11.2011)
- ▲ C Programming (Wikibook). [http://en.wikibooks.org/wiki/C Programming](http://en.wikibooks.org/wiki/C_Programming) (visto 11.2011)
- ▲ A Little C Primer (Wikibook). [http://en.wikibooks.org/wiki/A Little C Primer](http://en.wikibooks.org/wiki/A_Little_C_Primer) (visto 11.2011)

C y Procesamiento de Imágenes

- ▲ D. Phillips; Image Processing in C: Analyzing and Enhancing Digital Images, RandD Publications, 1994. Has source code, Windows, ~2004: <http://homepages.inf.ed.ac.uk/rbf/BOOKS/PHILLIPS/>

Procesamiento de imágenes

- ▲ Digital Image Processing. Rafael Gonzalez & Richard Woods.
 - ▲ Computer and Robot Vision. Robert Haralick y Linda Shapiro.
 - ▲ Computer Vision: A Modern Approach. David Forsyth y Jean Ponce.
 - ▲ David A. Forsyth and Jean Ponce (2003). Computer Vision, A Modern Approach. Prentice Hall. ISBN 0-12-379777-2.
 - ▲ Multiple View Geometry in Computer Vision. Andrew Zisserman y Robert Hartley.
 - ▲ Richard Hartley and Andrew Zisserman (2003). Multiple View Geometry in computer vision. Cambridge University Press. ISBN 0-521-54051-8
 - ▲ Variational Methods in Image Segmentation. J.M. Morel y S. Solimini.
-