



f2

**Facultad de Ingeniería  
Comisión Académica de Posgrado**

---

**Formulario de Aprobación Curso de Posgrado 2013**

**Asignatura:**

Introducción a la Filosofía de la Ciencia de la Computación.

---

**Profesor de la asignatura <sup>1</sup>:**

Msc. Luis Sierra, Prof. Adjunto (Gr. 3), InCo

**Profesor Responsable Local <sup>1</sup>:**

Dra. Sylvia da Rosa, Prof. Agregada (Gr. 4), InCo

**Otros docentes de la Facultad:**

(título, nombre, grado, Instituto)

**Docentes fuera de Facultad:**

**Instituto ó Unidad:**

**Departamento ó Area:**

<sup>1</sup> Agregar CV si el curso se dicta por primera vez.

(Si el profesor de la asignatura no es docente de la Facultad se deberá designar un responsable local)

---

**Fecha de inicio y finalización:** 5 de agosto al 1 de noviembre.

**Horario y Salón:**

**Horas Presenciales:** 30

**Nº de Créditos:** 6

(de acuerdo a la definición de la UdelaR, un crédito equivale a 15 horas de dedicación del estudiante según se detalla en el ítem metodología de la enseñanza)

**Público objetivo y Cupos:**

Cupo máximo 20 personas (ver justificación adjunta).

El curso se dicta en el marco de las actividades del Núcleo Interdisciplinario *Filosofía de la Ciencia de la Computación*, y se ofrecerá a:

Estudiantes de posgrado o grado de Ingeniería en Computación.

Estudiantes de posgrado o grado de Filosofía.

Estudiantes y profesores del profesorado de informática.

A estudiantes y profesores vinculados al Espacio Interdisciplinario de la UDELAR.

---

**Objetivos:**



## Facultad de Ingeniería Comisión Académica de Posgrado

---

Introducir al estudiante en los problemas de la filosofía de la ciencia de la computación, desde su dimensión ontológica, epistemológica, y metodológica. Los problemas, en la medida en que sea posible, se abordarán desde una perspectiva interdisciplinaria.

---

**Conocimientos previos exigidos:** Ninguno.

**Conocimientos previos recomendados:** Un curso básico de lógica. Conocimientos básicos de programación. Lectura de inglés.

---

**Metodología de enseñanza:**

(comprende una descripción de las horas dedicadas por el estudiante a la asignatura y su distribución en horas presenciales -de clase práctica, teórico, laboratorio, consulta, etc.- y no presenciales de trabajo personal del estudiante)

La metodología comprende instancias presenciales y a distancia usando la plataforma moodle, según se detalla abajo. Se cuenta con la colaboración de los docentes Alejandro Chmiel, Asistente (Gr. 2) de la Facultad de Humanidades, FHCE, que se desempeña en la asignatura de Lógica del Instituto de Filosofía y el Profesor de Filosofía, Guillermo Nigro, Asistente (Gr. 2), docente del NI FCC.

- Horas clase (teórico): 24
- Horas clase (práctico): ---
- Horas clase (laboratorio): ---
- Horas consulta: ---
- Horas evaluación: 6
  - Subtotal horas presenciales: 30
- Horas estudio: 36
- Horas resolución ejercicios/prácticos: 16
- Horas proyecto final/monografía: 8
  - Total de horas de dedicación del estudiante: 90

---

**Forma de evaluación:**

Los participantes deberán responder una serie de preguntas y ejercicios, por cada unidad. También se espera que elaboraren una presentación oral hacia el final del curso.

---

**Temario:**

**Unidad 1. Introducción a los problemas.**

1. ¿Qué es la computación? Perspectivas generales.
2. La noción de programa.

3. Presentación de los tres principales paradigmas desde los cuales se interpreta la disciplina: racionalista, científico y tecnológico.

**Unidad 2. Introducción a los problemas ontológicos.**

- 1 ¿Qué es un programa? Nivel físico y nivel simbólico.
- 2 Compromiso ontológico.
- 3 Breve desarrollo histórico de los lenguajes de programación.
- 4 Distinción entre distintos tipos de lenguajes de programación.
- 5 Lenguajes de programación y sus compromisos ontológicos.
- 6 Relación entre el programa y la máquina (implementación física).
- 7 Formalismo y platonismo.

**Unidad 3. Introducción a los problemas epistemológicos.**

1. Introducción al problema del conocimiento.
2. ¿Qué podemos conocer acerca de un programa?
3. ¿Qué significa que un programa es correcto?
4. Problemas de complejidad vinculados a la evaluación de la corrección.

**Unidad 4. Tópicos especiales.**

1. Problemas filosóficos en torno al concepto de *máquina*.
2. Problemas filosóficos de la inteligencia artificial.

---

**Bibliografía:**

Amnon H Eden. "Three Paradigms of Computer Science." *Minds and Machines*, Special issue on the Philosophy of Computer Science, Vol. 17, No. 2 (Jul. 2007), pp. 135–167. London: Springer.

Amnon H. Eden, Raymond Turner. "Problems in the ontology of computer programs." *Applied Ontology* Vol. 2, No. 1 (2007), pp. 13–36. Amsterdam: IOS Press.

DeMillo, R.A., Lipton, R.J. and Perlis, A.J., 1979, "Social Processes and Proofs of Theorems and Programs", *Communications of the ACM* 22(5): 271–280.

Fetzer, J.H., 1988, "Program Verification: The Very Idea", *Communications of the ACM* 31(9): 1048–1063.

Chalmers, D. *Does a Rock Implement Every Finite State Automaton?* (1996)

Colburn, T., 2004, "Methodology of Computer Science", *The Blackwell Guide to the Philosophy of Computing and Information*, Luciano Floridi (ed.), Malden: Blackwell, pp. 318–326.

Colburn, T., and Shute, G., 2007, "Abstraction in Computer Science", *Minds and Machines* 17(2): 169–184.

Copeland, B. Jack, "The Modern History of Computing", *The Stanford Encyclopedia of Philosophy* (Fall 2008 Edition), Edward N. Zalta (ed.), URL = <<http://plato.stanford.edu/archives/fall2008/entries/computing-history/>>.

Copeland, B. Jack, 2008, "The Church-Turing Thesis", *The Stanford Encyclopedia of Philosophy* (Fall 2008 Edition), Edward N. Zalta (ed.), URL = <<http://plato.stanford.edu/archives/fall2008/entries/church-turing/>>.

Copeland, B. Jack, 2004, "Computation", *The Blackwell Guide to the Philosophy of Computing and Information*, Luciano Floridi (ed.), Malden: Blackwell, pp. 3–17.

sigue en f. 5



## Facultad de Ingeniería Comisión Académica de Posgrado

---

- Fetzer, J.H., 1988, "Program Verification: The Very Idea", *Communications of the ACM* 31(9): 1048-1063.
- Floridi, Luciano, 2004. "Information", *The Blackwell Guide to the Philosophy of Computing and Information*, Luciano Floridi (ed.), Malden: Blackwell, pp. 40-62.
- Floridi, Luciano 2007, "Semantic Conceptions of Information", *The Stanford Encyclopedia of Philosophy* (Spring 2007 Edition), Edward N. Zalta (ed.), URL = <<http://plato.stanford.edu/archives/spr2007/entries/information-semantic/>>.
- Gandy, R. (1980) Church's Thesis and principles for mechanisms; in: *The Kleene Symposium* (edited by J. Barwise, H.J. Keisler and K. Kunen), NorthHolland, 123-148.
- Hoare, C.A.R., 1969, "An axiomatic basis for computer programming". *Communications of the ACM* 12(10):576-585.
- Moor, J.H., 1978, "Three Myths of Computer Science", *The British Journal for the Philosophy of Science* 29(3): 213-222.
- Rapaport, W.J., 2005b, "Implementation is Semantic Interpretation: Further Thoughts." *Journal of Experimental and Theoretical Artificial Intelligence* 17(4): 385-417.
- Sieg, W. (1994). *Mechanical Procedures and Mathematical Experience*. Mathematics and Mind. A. George. New York, Oxford University Press: 71-117.
- Smith, B.C., 1996, "Limits of Correctness in Computers", *Computerization and Controversy*, Kling, R. (ed.), Morgan Kaufman, pp. 810-825.
- Turner, Raymond, 2007, "Understanding Programming Languages". *Minds and Machines* 17(2): 129-133.
- Turner Raymond and Eden, Amnon. "Philosophy of computer science." In *The Stanford Encyclopedia of Philosophy* (Winter 2008 edition), Edward N. Zalta (ed.).
- Richard L. Wexelblat (ed.), 1981, *History of Programming Languages*, Academic Press.