

Programa del curso IC-6200

Inteligencia Artificial

Escuela de Computación
Carrera de Ingeniería de Computación, Plan 410.

I parte: Aspectos relativos al plan de estudios

1 Datos generales

Nombre del curso:	Inteligencia Artificial
Código:	IC-6200
Tipo de curso:	Teórico - Práctico
Nº de créditos:	4
Nº horas de clase por semana:	4
Nº horas extraclase por semana:	8
Ubicación en el plan de estudios:	Curso del 7º semestre de la carrera de Ingeniería en Computación
Requisitos:	IC6400 Investigación de Operaciones. IC5701 Compiladores e Intérpretes.
Correquisitos:	Ninguno
El curso es requisito de:	Ninguno
Asistencia:	Obligatoria
Suficiencia:	No
Posibilidad de reconocimiento:	No
Vigencia del programa:	I semestre 2012.

2 Descripción general

Curso teórico y formal que introduce al estudiante en el manejo de la representación del conocimiento, búsqueda, control y aprendizaje. Para su uso en la construcción de algoritmos para la solución de problemas de la inteligencia artificial.

3 Objetivos

Objetivo General

Al finalizar el curso el estudiante debe ser capaz de analizar y desarrollar algoritmos y programas para la solución de problemas que requieran búsqueda, control y aprendizaje.

Objetivos Específicos

- Analizar y desarrollar algoritmos para la solución de problemas de la inteligencia artificial.
- Construir y validar programas utilizando un lenguaje especializado de inteligencia artificial.
- Construir y analizar algoritmos para representación del conocimiento, búsqueda control y aprendizaje.

4 Contenidos

Introducción al Curso

- Definición de inteligencia
- Definición de inteligencia artificial
- Problemas que ataca la IA
- El modelo de Agentes

Introducción a la Programación para Inteligencia Artificial.

- Introducción a la programación simbólica
- Funciones como elementos de representación
- Procesamiento de Listas
- Procesamiento de Estructuras Complejas
- Procesamiento de Objetos
- Monitoreo y depuración de Programas

Solución de Problemas, búsqueda y control

- Problemas y espacios de búsqueda
- Definición de un problema como un espacio
- Profundidad primero, anchura primero, etc
- Métodos débiles
- Métodos de "hill-climbing"
- Algoritmos heurísticos
- Satisfacción de restricciones
- Sistemas de producción
- Razonamiento hacia adelante y hacia atrás
- "Match" y filtración

Representación del Conocimiento

- Lógica de predicados
- Representación del conocimiento mediante lógica
- Conversión a cláusulas
- Principio de unificación
- Resolución
- Lógicas no monotónicas
- Introducción al razonamiento no monotónico
- Introducción al razonamiento probabilístico
- Redes, redes semánticas y marcos

Redes semánticas
Marcos y "scripts"

Aprendizaje

Definición de Aprendizaje
Sistemas basados en conocimiento
Sistemas basados en casos
Redes Neurales

5 Metodología de enseñanza y aprendizaje

Clases magistrales, exámenes, tareas y quices.

El profesor asignará lecturas semanalmente, la asignación de la lectura incluirá una breve exposición para contextualizar al estudiante. Posteriormente a la lectura, el profesor desarrollará el tema en una clase magistral y el estudiante deberá ensayar posibles soluciones a preguntas de investigación derivadas tanto de la lectura como de la exposición.

La asimilación de los temas se reforzará con trabajo práctico, individual y en grupo, en forma de ejercicios, tareas cortas, quices, exámenes y proyectos.

6 Evaluación

Asistencia y participación en clase. Asignaciones, exposiciones y afines. Ensayos. Trabajo de investigación final. Exámenes

Examen parcial I	20%
Examen parcial II	20%

Proyectos programados	40%
Exámenes cortos, tareas	20%

Total: 100%

Cada lectura asignada será evaluada con un resumen, de máximo 700 palabras (aprox. 1 página), en el que el estudiante condensará, con sus propias palabras, los puntos más relevantes del texto y ocasionalmente deberá ensayar una propuesta de solución cuando el profesor asigne alguna(s) pregunta(s) de investigación relacionada(s) con dicha lectura. Los resúmenes corresponden a 10% del rubro de exámenes cortos y tareas.

Posterior a la finalización de cada tema se realizará un examen corto. La fecha de cada examen corto será avisada por el profesor con por lo menos una semana de antelación. Los exámenes cortos corresponden a 10% del rubro de exámenes cortos y tareas.

Durante el curso se asignarán tres proyectos programados: un proyecto de 10% posterior a la discusión del tema de solución de problemas, búsqueda y control; un proyecto de 10% posterior a la discusión del tema de representación del conocimiento; y finalmente un proyecto de 20% posterior al tema de aprendizaje.

Se realizarán dos exámenes parciales de 20% cada uno, el primero evaluará los temas de introducción al curso, introducción a la programación para inteligencia artificial y solución de problemas, búsqueda y control. El segundo examen evaluará los temas de representación del conocimiento y aprendizaje. Al no haber examen final, no hay posibilidad de eximirse.

7 Bibliografía

+ : se usa en el curso
* : esta en la biblioteca
j : se puede comprar en librerías

El resto son libros de apoyo

++(Armstrong, 2007)

Armstrong, Joe. "Programming Erlang: software for a concurrent World". The Pragmatic Bookshelf. Raleigh, North Carolina – Dallas, Texas. 2007.

*(Bratko, 1986)

Bratko, Ivan. "Prolog Programming for Artificial Intelligence". Addison-Wesley, Reading, Massachusetts, 1986.

+(Buckland, 2005)

Auckland, Mat. "Programming Game AI by Example". Wordware game developers library, Wordware Publishing Inc. Plano, Texas, 2005.

(Cristianini, Shawe-Taylor, 2005)

Cristianini, Nello; Shawe-Taylor, John. "An Introduction to Support Vector Machines and other kernel-based methods". Cambridge University Press. Cambridge, UK. 2005.

*(Dean, Allen, Yiannis, 1995)

Dean, Thomas; Allen, James; Aloimonos, Yiannis. "Artificial Intelligence: Theory and Practice". Addison-Wesley Publishing Company, Menlo Park California, 1995.

(Duda et al, 2001)

Duda, Ricard O.; Hart, Peter E.; Stork, David G. "Pattern Classification, Second Edition" John Wiley & Sons Inc. Wiley Interscience Publication. 2001.

+(Freeman / Skapura, 1993)

Freeman, James A.; Skapura, David M. "Redes Neuronales: Algoritmos, aplicaciones y técnicas de programación". Addison-Wesley/Diaz de Santos. Delaware EE.UU. 1993.

(Kamimura, 2002)

Kamimura, Ryotaro. "Information Theoretic Neural Computation".
World Scientific. Singapore, Mainland Press. 2002.

+(Mitchel,1997)

Mitchell, Tom M. "Machine Learning". McGraw-Hill Series in Computer
Science / MIT Press. 1997.

(Nguyen/Walker, 2000)

Nguyen, Hung T.; Walker, Elbert A. " A First Course in Fuzzy Logic,
Second Edition". Chapman & Hall/CRC. 2000.

+(Norvig, 1992)

Norvig, Peter. "Artificial Intelligence Programming". Morgan Kaufmann.
San Francisco, California. 1992.

*(Rich, Knigth, 1995)

Rich, Elaine; Knight, Kevin. "Inteligencia Artificial". 2da Edición, Mc
Graw Hill Interamericana de España, Madrid, 1995.

(Riesbeck, Schank, 1989)

Riesbeck, Christopher K.; Schank, Roger C. "Inside Case-Based
Reasoning". Lawrence Erlbaum Associates, Puablishers, Hillsdale, New
Jersey, 1989.

+++!(Russell,Norvig,2002)

Russell, Stuart J.; Norvig, Peter; "Artificial Intelligence: A Modern
Approach" Prentice Hall, 2nd Edition, Saddle River, NJ, 2002.

+(Segaran, 2007)

Segaran, Toby. "Programming Collective Intelligence: building smart
web 2.0 Applications". O'Reilly, Sebastopol, California, 2007.

(Si et al, 2004)

Si, Jennie; Barto, Andrew G.; Powel, Warren B.; Wunsch II, Donald.
"Handbook of Learning and Approximate Dynamic Programming".

IEEE Press on Computational Inteligente. Piscataway NJ. 2004.

(Wasserman, 1989)

Wasserman, Philip D. "Neural Computing Theroy and Practice". Van Nostrand Reinhold, New York, 1989.

*(Wasserman, 1992)

Wasserman Philip D. "Advanced Methods in Neural Computing". Van Nostrand Reinhold, New York, 1993.

*(Winston, 1992) Winston,

Patrick. "Artificial Intelligence". 3era Edición, Addison-Wesley, Reading, Massachusetts, 1992

8 Profesor

Se le recomienda al estudiante conocer el reglamento de enseñanza y aprendizaje del ITCR, en caso de fraude, apelación o conflicto el curso se regirá por dicho reglamento.

Diego Munguía Molina tiene estudios de Ingeniería en Computación (ITCR) y es egresado de la maestría en Ciencias Cognoscitivas (UCR), actualmente se encuentra desarrollando su tesis de graduación. Profesionalmente se ha desarrollado como arquitecto de software, acumulando experiencia en ingeniería de software, computación de alto rendimiento e integración de sistemas. Ha laborado como docente para la institución desde el 2012 en la Escuela de Computación impartiendo los cursos de Introducción a la Programación, Taller de Programación, Lenguajes de Programación, Compiladores e Intérpretes, Programación Orientada a Objetos, Diseño de Software e Inteligencia Artificial.

Medio oficial electrónico:

<http://www.ic-itcr.ac.cr/~dmunguia/pages/ic-6200.html>

Correos electrónicos: dmunguia@itcr.ac.cr

Entrega de proyectos: <http://tec.siuu.ac.cr/gitlab/>

Oficina de Ingeniería en Computación, SIUA

Horario de consulta: K 11:00 a 12:00 y de 14:00 a 15:00