

1.- DATOS DE LA ASIGNATURA

Nombre de la asignatura: **Programación Lógica y Funcional**

Carrera: **Ingeniería en Sistemas Computacionales**

Clave de la asignatura: SCC-1019

(Créditos) SATCA¹ 2 – 2 – 4

2.- PRESENTACIÓN

Caracterización de la asignatura.

La asignatura de Programación Lógica y Funcional aporta al perfil del Ingeniero en Sistemas Computacionales la capacidad de desarrollar habilidades para la generación de soluciones automatizadas basadas en lenguajes de inteligencia artificial, considerando el entorno y la aplicación de diversas técnicas, herramientas y conocimientos.

Los programas para computadora actualmente son fundamentales en muchas áreas del ser humano, debido a que se usan para resolver diversos problemas en la ciencia, la industria y los negocios. Para cubrir estas necesidades, se han desarrollado lenguajes de programación dentro de la inteligencia artificial. El Ingeniero en Sistemas Computacionales contribuirá, aplicando estos conocimientos para la solución de problemas a través de la programación lógica y funcional, con una conciencia ética y de respeto al medio ambiente.

Programación Lógica y Funcional, es una asignatura que requiere tener conocimientos esenciales acerca de los lenguajes lógicos y funcionales de la inteligencia artificial, incluyendo la metodología y los aspectos relativos a la codificación, con el fin de ampliar el conocimiento de tecnologías alternativas para el desarrollo de sistemas automatizados y la implementación de agentes inteligentes.

Intención didáctica.

- Estimular el pensamiento creativo.
- Trabajar con metodologías participativas y de reflexión colectiva.
- Propiciar la solución de problemas concretos.
- Aprovechar los conocimientos que el estudiante adquirió en asignaturas anteriores.
- Propiciar en el estudiante la construcción e interiorización del conocimiento de los principios lógicos y funcionales de la programación para el diseño de agentes inteligentes.

¹ Sistema de asignación y transferencia de créditos académicos

Se denomina inteligencia artificial a la rama de la ciencia informática dedicada al desarrollo de agentes racionales no vivos (agentes inteligentes). Es la disciplina que se encarga de construir procesos que al ser ejecutados sobre una arquitectura física producen acciones o resultados que maximizan una medida de rendimiento determinada, basándose en la secuencia de entradas percibidas y en el conocimiento almacenado en tal arquitectura.

La inteligencia artificial incluye varios campos de desarrollo tales como: la robótica, usada principalmente en el campo industrial; comprensión de lenguajes y traducción; visión en máquinas que distinguen formas y que se usan en líneas de ensamblaje; reconocimiento de palabras y aprendizaje de máquinas; sistemas computacionales expertos.

Los sistemas expertos, que reproducen el comportamiento humano en un estrecho ámbito del conocimiento; son programas tan variados como los que diagnostican infecciones en la sangre e indican un tratamiento, los que interpretan datos sísmológicos en exploración geológica y los que configuran complejos equipos de alta tecnología.

La IA es un conjunto de métodos y filosofía, y por lo tanto no está atada a un lenguaje en particular. Los problemas de IA requieren que los programas manipulen conocimiento en lugar de números. Para lo anterior, se desarrollaron nuevos lenguajes para atacar estos problemas, como lo son: de programación simbólica y de programación lógica.

El temario está organizado en cuatro unidades y con una estructura lógica. En la primera unidad se inicia con un acercamiento a los conceptos fundamentales de la programación, con la idea de que el estudiante desarrolle una visión de conjunto de los lenguajes de alto nivel y sirva como marco de referencia a la metodología de los lenguajes de inteligencia artificial, como lo son la programación lógica y funcional, por lo que es recomendable una previa selección de materiales y lecturas de apoyo por parte del docente.

La unidad dos integra la programación funcional, haciendo énfasis, al uso de nuevas metodologías para que la actividad del estudiante vaya más allá de la intuición y reflexión. Proporciona nuevas habilidades, distintas a las desarrolladas en los paradigmas convencionales de la programación. Son lenguajes de propósito general, procedural, basado en aritmética y manipulación de símbolos, utilizan datos de entrada y funciones específicas. La programación; en términos de relaciones entre símbolos, se denomina programación simbólica. Dado que el conocimiento consiste en símbolos y asociaciones entre ellos, estos lenguajes son convenientes en aplicaciones de inteligencia artificial. Esta unidad permite que el estudiante conozca las ventajas y puntos débiles de la programación funcional; diseñe, codifique, pruebe y depure programas funcionales; identifique los principales

ámbitos de aplicación de esta programación, para que las utilice como herramientas alternativas y complementarias en la solución de problemas a través de la primera característica esencial de estos lenguajes, que es la función de orden superior; es decir, funciones que tienen como argumento a otras funciones (dicho de otra forma, programas que tienen como argumentos otros programas).

La unidad tres está dirigida a la segunda característica esencial de los lenguajes funcionales, que es la evaluación perezosa, en la cual sólo se realiza un cálculo cuando otro cálculo posterior lo necesita, lo que permite la definición de estructuras infinitas de datos, así como de funciones y tipos de datos no estrictos. Este modo de evaluar expresiones tiene una serie de ventajas técnicas importantes, entre ellas, se evitan cálculos innecesarios. La evaluación perezosa consiste en utilizar paso por nombre y recordar los valores de los argumentos ya calculados para evitar recalcularlos. También se denomina estrategia de pasos de parámetros por necesidad.

La última unidad, esta dirigida a la programación lógica, enfocada a la utilización de metodologías, que permiten la relación de datos; y que proporcione al estudiante nuevas habilidades para establecer, si un dato es conocido o se deriva de una base de conocimientos; distintas a las desarrolladas en los paradigmas convencionales de la programación. La unidad permite que el alumno conozca las ventajas de la programación lógica; diseñe, codifique y pruebe programas lógicos; identifique los principales ámbitos de aplicación de la programación lógica como herramienta alternativa y complementaria de la programación, para tener una visión de los nuevos paradigmas de la programación diferentes a los que conoce, lo cual implica, una forma de resolver problemas de distinta manera y ser un complemento ideal a su formación como ingeniero en sistemas computacionales.

Para las unidades dos y tres, se realizarán prácticas en los lenguajes simbólicos y lógicos, implementando algoritmos de juegos para aplicar los conceptos aprendidos en cada tema. Se recomienda el desarrollo de un sistema experto básico para cada unidad, en la presentación de cada sistema es conveniente que el estudiante viva la aplicación del mismo en el aula, con sus exigencias y responda con profesionalismo y responsabilidad.

3.- COMPETENCIAS A DESARROLLAR

Competencias específicas:	Competencias genéricas
<ul style="list-style-type: none">• Conocer los principios lógicos y funcionales de la programación para identificarlos y aplicarlos en la resolución de problemas a través del diseño de agentes inteligentes.	<p>Competencias instrumentales</p> <ul style="list-style-type: none">• Capacidad de análisis y síntesis.• Capacidad de organizar y planificar.• Conocimientos básicos de la carrera• Comunicación oral y escrita.• Manejo de todos los recursos que proporciona la computadora.• Habilidad para buscar y analizar información proveniente de fuentes diversas.• Solución de problemas.• Toma de decisiones. <p>Competencias interpersonales</p> <ul style="list-style-type: none">• Capacidad crítica y autocrítica.• Trabajo en equipo.• Habilidades interpersonales. <p>Competencias sistémicas</p> <ul style="list-style-type: none">• Capacidad de aplicar los conocimientos en la práctica.• Habilidades de investigación.• Capacidad de aprender.• Capacidad de generar nuevas ideas (creatividad).• Habilidad para trabajar en forma autónoma.• Búsqueda del logro.

4.- HISTORIA DEL PROGRAMA

Lugar y fecha de elaboración o revisión	Participantes	Observaciones (cambios y justificación)
Instituto Tecnológico de Saltillo, del 5 al 9 de octubre de 2009	Representantes de los Institutos Tecnológicos de: San Luis Potosí, Celaya, Pinotepa, Superior de Libre, Tapachula.	Reunión Nacional de Diseño e Innovación Curricular para el Desarrollo de Competencias Profesionales de la Carrera de Ingeniería en Sistemas Computacionales.
Instituto Tecnológico de San Luis Potosí, del 12 de octubre de 2009 al 19 de febrero de 2010	Representante de la Academias de ISC de los Institutos Tecnológicos de San Luis Potosí, Celaya, Pinotepa, Superior de Libre, Tapachula, Toluca.	Análisis, enriquecimiento y elaboración del programa de estudio propuesto en la Reunión Nacional de Diseño e Innovación Curricular para el Desarrollo de Competencias Profesionales de la Carrera de Ingeniería en Sistemas Computacionales.
Instituto Tecnológico Superior de Poza Rica Fecha del 22 al 26 de febrero de 2010	Representantes de los Institutos Tecnológicos participantes en el diseño de la carrera de Ingeniería en Sistemas Computacionales: Institutos Tecnológicos de San Luis Potosí, Celaya, Pinotepa, Superior de Libre, Tapachula, Toluca.	Reunión Nacional de Consolidación Curricular de la carrera de Ingeniería en Sistemas Computacionales.

5.- OBJETIVO GENERAL DEL CURSO

Conocer los principios lógicos y funcionales de la programación para identificarlos y aplicarlos en la resolución de problemas a través del diseño de agentes inteligentes.

6.- COMPETENCIAS PREVIAS

- Conocimiento y aplicación de programación orientada a objetos para resolver situaciones reales.

7.- TEMARIO

Unidad	Temas	Subtemas
1	Conceptos Fundamentales.	1.1. Estilos de programación. 1.2. Evaluación de expresiones. 1.3. Definición de funciones. 1.4. Disciplina de tipos. 1.5. Tipos de datos.
2	Programación Funcional.	2.1. El tipo de datos. 2.2. Funciones. 2.3. Intervalos. 2.4. Operadores. 2.5. Aplicaciones de las listas. 2.6. Árboles.
3	Evaluación perezosa.	3.1. La estrategia de evaluación perezosa. 3.2. Técnicas de programación funcional perezosa.
4	Fundamentos de la programación lógica.	4.1. Repaso de la lógica de primer orden. 4.2. Unificación y resolución. 4.3. Cláusulas de Horn. Resolución SLD. 4.4. Programación lógica con cláusulas de Horn. 4.5. Semántica de los programas lógicos. 4.6. Representación clausada del conocimiento. 4.7. Consulta de una base de cláusulas 4.8. Espacios de búsqueda. 4.9. Programación lógica con números, listas y árboles. 4.10. Control de búsqueda en programas lógicos 4.11. Manipulación de términos. Predicados metalógicos.

8.- SUGERENCIAS DIDÁCTICAS

- Que la materia sea eminentemente práctica y participativa, combinando elementos de aprendizaje con dinámicas de grupo, ejercicios de reflexión individual y grupal que faciliten el desarrollo de competencias.
- Se busca potenciar las cualidades creativas del estudiante, estimulando permanentemente el diálogo de saberes, la reflexión y la acción.
- Este enfoque plantea que el conocimiento no se adquiere simplemente, ni se recibe, ni es una copia de la realidad, sino que es una construcción de la persona a partir de la percepción e interpretación de la realidad.
- Usar un portal de Internet para apoyo didáctico de la materia, el cual cuente por lo menos con un foro, preguntas frecuentes, material de apoyo y correo electrónico.
- Utilizar herramientas de docencia no presencial, como soporte de los contenidos teóricos y prácticos, incorporando documentación adicional, guiones de prácticas y herramientas de apoyo.
- Trabajar en grupos pequeños, para sintetizar y construir el conocimiento necesario para resolver problemas relacionados con situaciones reales.
- Solicitar al estudiante propuestas de problemas a resolver y que sean significativas para él.
- Propiciar que el estudiante experimente con diferentes programas encontrados en revistas, Internet y libros de la especialidad, que lo lleven a descubrir nuevos conocimientos.
- Elaborar de manera conjunta con el estudiante una guía de ejercicios para actividades extra clase.
- Plantear problemas reales para que ellos diseñen soluciones utilizando los conceptos de la IA.
- Desarrollo de un proyecto con aplicación real basado en lenguaje lógico.
- En la solución de problemas, motivar que seleccione la metodología que permita que la aplicación sea pertinente y viable.
- Propiciar en el estudiante, el desarrollo de actividades intelectuales de inducción-deducción y análisis-síntesis, las cuales lo encaminan hacia la investigación, la aplicación de conocimientos y la solución de problemas.
- Llevar a cabo actividades prácticas que promuevan el desarrollo de habilidades para la experimentación, tales como: observación, identificación, manejo y control de variables, datos relevantes, planteamiento de hipótesis y trabajo en equipo.
- Desarrollar actividades de aprendizaje que propicien la aplicación de los conceptos, modelos y metodologías aprendidas en el desarrollo de la asignatura a través de diferentes técnicas, como pueden ser: mapas conceptuales o mentales.
- Propiciar el uso adecuado de conceptos y de terminología científico tecnológica.

- Proponer problemas que permitan al estudiante integrar y relacionar los contenidos de esta asignatura con otras, para su análisis y solución.
- Observar y analizar fenómenos y problemáticas propias del campo ocupacional.
- Se recomienda utilizar en programación funcional, Lisp y para programación lógica, Prolog.

9.- SUGERENCIAS DE EVALUACIÓN

La evaluación de la asignatura será continua y cotidiana, por lo que se debe considerar el desempeño en cada una de las actividades de aprendizaje, haciendo especial énfasis en:

- Evaluación continua.
- Ponderación de tareas.
- Participación y desempeño en el aula y el laboratorio, a través de dinámicas grupales, trabajo individual o en equipo.
- Dar seguimiento al desempeño en el desarrollo del programa (dominio de los conceptos, capacidad de la aplicación de los conocimientos en problemas reales, transferencia del conocimiento).
- Desarrollo de un proyecto final basado en programación lógica y que integre además todas las unidades de aprendizaje.
- Cumplimiento de los objetivos y desempeño en las prácticas de cada tema.
- Información obtenida durante las investigaciones solicitadas en documentos escritos.
- Exámenes escritos para comprobar el manejo de aspectos teóricos y declarativos.

10.- UNIDADES DE APRENDIZAJE

Unidad 1: Conceptos Fundamentales

Competencia específica a desarrollar	Actividades de Aprendizaje
Identificar los paradigmas y lenguajes de programación representativa.	<ul style="list-style-type: none">• Visualizar los diversos estilos de la programación.• Identificar los conceptos básicos de la programación representativa.• Reconocer las características de la programación representativa.• Investigar, al menos, un lenguaje de programación representativa.• Realizar mapa conceptual de los paradigmas y lenguajes de la programación representativa.

Unidad 2: Programación funcional

Competencia específica a desarrollar	Actividades de Aprendizaje
<p>Conocer los principales puntos fuertes y debilidades del paradigma de programación funcional</p> <p>Identificar los elementos de la programación funcional.</p> <p>Aplicar la programación funcional en la resolución de problemas reales.</p> <p>Diseñar herramientas alternativas y</p>	<ul style="list-style-type: none">• Identificar los conceptos básicos de la programación funcional.• Describir las características de la programación funcional.• Reconocer la estructura y elementos de la programación funcional.• Investigar; al menos, un lenguaje de programación representativa diferente al establecido para la materia.• Realizar mapa conceptual de los lenguajes de la programación funcional vistos en la materia.

complementarias de programación.	<ul style="list-style-type: none"> • Aplicar los conceptos de la programación funcional para resolver un problema real, diseñando un programa sencillo.
----------------------------------	--

Unidad 3: Evaluación perezosa

Competencia específica a desarrollar	Actividades de Aprendizaje
<p>Conocer la evaluación perezosa.</p> <p>Identificar la evaluación perezosa como una de las funciones de la programación representativa.</p> <p>Aplicar la evaluación perezosa en la resolución de problemas.</p> <p>Diseñar programación con modularidad.</p>	<ul style="list-style-type: none"> • Identificar los conceptos básicos de la evaluación perezosa. • Describir las técnicas de la programación funcional perezosa. • Investigar, al menos, una técnica de programación funcional perezosa vista en clase. • Realizar mapa conceptual de la evaluación perezosa • Aplicar una técnica de la programación funcional perezosa, para resolver un problema real a través de la modularidad en una situación sencilla.

Unidad 4: Fundamentos de la programación lógica

Competencia específica a desarrollar	Actividades de Aprendizaje
<p>Conocer las ventajas y desventajas del paradigma de programación lógica.</p> <p>Identificar los elementos de la programación lógica.</p> <p>Aplicar la programación lógica en la resolución de problemas reales.</p> <p>Diseñar sistemas expertos con programación lógica.</p>	<ul style="list-style-type: none"> • Identificar los conceptos básicos de la programación lógica. • Describir las clausulas de Horn y resolución SLD, para identificar reglas de inferencia lógica y emplearlas en la representación del conocimiento. • Reconocer los elementos de la semántica de la programación lógica para interpretar el conocimiento y aplicarlo en su representación. • Investigar, al menos, un lenguaje de programación lógica diferente al establecido para la materia. • Realizar mapa conceptual de la

	<p>programación lógica.</p> <ul style="list-style-type: none"> • Aplicar la programación lógica para resolver un problema real, diseñando un sistema experto basado en el control de búsqueda lógica.
--	--

11.- FUENTES DE INFORMACIÓN

1. NILSSON, N. J. *Inteligencia Artificial. Una nueva síntesis*. Mc Graw Hill. 2001.
2. POOLE, D., Mackworth, A. y Goebel, R. *Computational Intelligence (A Logical Approach)*. Oxford University Press. 1998.
3. BRATKO, I. *Prolog Programming for Artificial Intelligence (2nd ed.)*. Addison Wesley. 1990.
4. MITCHELL, T. M. *Machine Learning*. Mc Graw Hill. 1997.
5. FLACH, P. *Simply Logical (Intelligent Reasoning by Example)*. John Wiley. 1994.
6. BIRD, Richard. *Introducción a la Programación Funcional con Haskell*. Segunda Ed. Prentice Hall. 2000.
7. FOKKER, Jeroen. *Programación Funcional*. Universidad de Utrecht, Departamento de Informática. 1995.
8. JULIAN, P., Alpuente, M. *Programación Lógica. Teoría y Práctica*. Pearson Prentice Hall. 2007.
9. HOGGER, C. *Essentials of Logic Programming*. Clarendon Press, Oxford. 1990.
10. BRATKO. *Prolog Programming for Artificial Intelligence*. Segunda Edición). Addison Wesley. 1991.
11. STERLING & Shapiro. *The art de Prolog*. MIT. 1994.
12. LUCAS, P. y Gaag, L.v.d. *Principles of Expert Systems*. (Addison–Wesley. 1991.

Fuentes electrónicas:

1. Programación Funcional. Clase 1. En línea en:
<http://funcional.fciencias.unam.mx/pdf/class1.pdf>
 Programación Funcional. Clase 2 En línea en:
<http://funcional.fciencias.unam.mx/pdf/class2.pdf>
 Programación Funcional. Clase 3 En línea en:
<http://funcional.fciencias.unam.mx/pdf/class3.pdf>

Programación Funcional. Clase 4 En línea en:

<http://funcional.fciencias.unam.mx/pdf/class4.pdf>

2. Programación Funcional. En línea en:

http://es.wikipedia.org/wiki/Programaci%C3%B3n_funcional

3. Programación Lógica. En línea en:

[http://es.wikipedia.org/wiki/Programación_lógica](http://es.wikipedia.org/wiki/Programaci%C3%B3n_l%C3%B3gica)

4. Programación lógica. En línea en:

www.cs.cinvestav.mx/PaginaAntigua/SC/publica/chapa/intro_lm/node42.html

5. Programación lógica. En línea en:

http://expo.itchihuahua.edu.mx/view.php?f=prog_46

6. Análisis lógico 2008-2, nota de clase 8. Fundamentos de programación lógica I.

Resolución binaria. En línea en:

<http://abulafia.fciencias.unam.mx/~faviu/cursos/al82/notas/al82n8.pdf>

7. Sistema Experto. En línea en: http://es.wikipedia.org/wiki/Sistema_experto

12.- PRÁCTICAS PROPUESTAS

- Elaboración de mapas conceptuales y/o mentales de los temas vistos.
- Elaboración de glosario por unidad de conceptos clave de cada tema.
- Desarrollo de programas funcionales con un grado creciente de complejidad, utilizando herramientas de programación funcional, que den solución a problemas reales.
- Diseñar y construir una base de conocimiento a través de programación funcional.
- Desarrollo de programas lógicos con un grado creciente de complejidad, utilizando herramientas de programación lógica, que den solución a problemas reales.
- A partir de una situación real, diseñar y construir una base de conocimiento a través de herramientas de sistemas expertos basado en programación lógica.
- Construir un sistema experto a partir de la base de conocimiento creada en programación lógica.