



UAEM | Universidad Autónoma
del Estado de México

SD
Secretaría de Docencia



Universidad Autónoma del Estado de México • Secretaría de Docencia • Dirección de Estudios Profesionales

Universidad Autónoma del Estado de México

Licenciatura de Ingeniería en Sistemas Inteligentes 2007

Programa de estudios de la unidad de aprendizaje:

Fundamentos de programación



I. Datos de identificación

Licenciatura

Unidad de aprendizaje Clave

Carga académica

Horas teóricas Horas prácticas Total de horas Créditos

Período escolar en que se ubica 1 2 3 4 5 6 7 8 9 10

Seriación

UA Antecedente UA Consecuente

Tipo de Unidad de Aprendizaje

Curso Curso-taller

Seminario Taller

Laboratorio Práctica profesional

Otro tipo (especificar)

Modalidad educativa

Escolarizada. Sistema rígido No escolarizada. Sistema virtual

Escolarizada. Sistema flexible No escolarizada. Sistema a distancia

No escolarizada. Sistema abierto Mixta (especificar)

Formación común

Unidad de Aprendizaje

Formación equivalente

Unidad de Aprendizaje



II. Presentación del programa

--

III. Ubicación de la unidad de aprendizaje en el mapa curricular

Núcleo de formación:	Sustantivo
Área Curricular:	Programación e ingeniería de software de base
Carácter de la UA:	Obligatoria

IV. Objetivos de la formación profesional

Objetivos del programa educativo:

Formar Ingenieros en Sistemas Inteligentes que contribuyen al progreso social, económico y cultural del país; y desarrollar en el alumno los aprendizajes y competencias para:

- Comprender los fundamentos científicos y tecnológicos de la ingeniería en computación, así como de sus áreas de desarrollo
- Comprender y aplicar los conocimientos, técnicas y herramientas de la inteligencia artificial y de la minería de datos, para el desarrollo de sistemas inteligentes
- Desarrollar sistemas computacionales, mediante métodos y técnicas de inteligencia artificial, para el tratamiento de información, toma de decisiones y solución de problemas
- Utilizar de manera ética, económica y eficiente, los datos e información que mejoren la forma de decisiones sobre la gestión y el control de procesos
- Comprender los sistemas sociales y económicos, y sus efectos en el desarrollo de mejores soluciones tecnológicas



- Desarrollar investigación competitiva en el área de Ingeniería en Sistemas Inteligentes
- Innovar, asimilar, incorporar y aprovechar las tecnologías de la información

Objetivos del núcleo de formación: Sustantivo

El núcleo sustantivo desarrollará en el alumno el dominio teórico, metodológico y axiológico del campo de conocimiento donde se inserta la profesión.

Comprenderá unidades de aprendizaje sobre los conocimientos, habilidades y actitudes necesarias para dominar los procesos, métodos y técnicas de trabajo; los principios disciplinares y metodológicos subyacentes; y la elaboración o preparación del trabajo que permita la presentación de la evaluación profesional.

Este núcleo podrá comprender unidades de aprendizaje comunes o equivalentes entre dos o más estudios profesionales que imparta la Universidad, lo que permitirá que se cursen y acrediten en un plan de estudios y Organismo Académico, Centro Universitario o Dependencia Académica, diferentes al origen de la inscripción del alumno.

Objetivos del área curricular o disciplinaria:

- Estudiar la naturaleza de los lenguajes de programación considerando la filosofía que emplean para describir elementos de la realidad.
- Estudiar formas y características de implantación de los procesadores de los lenguajes.
- Analizar la evolución de los lenguajes de programación, así como presentar y discutir las tendencias futuras de su desarrollo.



V. Objetivos de la unidad de aprendizaje

El alumno desarrollará programas en un Lenguaje estructurado de propósito general, que sirva como base para cursos posteriores donde se desarrollen sistemas de computación.

VI. Contenidos de la unidad de aprendizaje, y su organización

1. Conceptos básicos del modelo orientado a objetos

- 1.1 Reconocimiento de objetos y clases en el mundo real y la interacción entre ellos.
- 1.2 La abstracción y el encapsulamiento como un proceso natural.
- 1.3 La POO y la complejidad del software.
- 1.4 Conceptos del ciclo de vida del software.
 - 1.4.1 Especificaciones de requerimientos.
 - 1.4.2 Análisis Orientado a Objetos.
 - 1.4.3 Diseño Orientado a Objetos.
 - 1.4.4 Programación Orientada a Objetos, conceptos y características.
- 1.5 Elementos primordiales en el modelo de objetos.
 - 1.5.1 Abstracción.
 - 1.5.2 Encapsulamiento.
 - 1.5.3 Modularidad.
 - 1.5.4 Jerarquía y herencia.
 - 1.5.5 Polimorfismo.
- 1.6 Historia de los paradigmas en el desarrollo del software.
- 1.7 Beneficios del modelo de objetos y de la POO sobre otros paradigmas.



2. Técnicas básicas de modelado de objetos

- 2.1 Definición de clases, atributos, métodos y objetos.
- 2.2 El Modelo como resultado de la abstracción.
- 2.3 El UML como una herramienta de modelado de objetos.
- 2.4 Planteamiento del problema.
 - 2.4.1 Analizar el enunciado del problema.
 - 2.4.2 Identificar funciones del sistema.
- 2.5 Análisis.
 - 2.5.1 Descubrir objetos en el dominio del problema.
 - 2.5.2 Identificar atributos de los objetos.
 - 2.5.3 Identificar métodos en los objetos.
- 2.6 Introducción al diseño de la solución.
 - 2.6.1 Representación gráfica de una clase.
 - 2.6.2 Diagramas de interacción entre la aplicación y una clase.
 - 2.6.3 Diagramas de estado de una clase.

3. Técnicas de diseño detallado.

- 3.1 Diseño algorítmico.
 - 3.1.1 Elementos y reglas de la representación gráfica de los algoritmos.
 - 3.1.2 Implementación de algoritmos secuenciales (utilizando notación algebraica).
- 3.2 Diseño algorítmico de las funciones.

4. Introducción a la programación.

- 4.1 Clasificación del software.
 - 4.1.1 Software del sistema.
 - 4.1.2 Software de aplicación.
- 4.2 Conceptos de la programación.
 - 4.2.1 Definición de programa.
 - 4.2.2 Definición de programación.
 - 4.2.3 Definición de lenguaje de programación.



4.3 Datos.

4.3.1 Definición.

4.3.2 Tipos de datos.

4.3.3 Identificadores.

4.3.4 Almacenamiento, direccionamiento y representación en memoria.

4.3.5 Sistema de numeración binaria y hexadecimal.

4.4 Operadores, operandos y expresiones.

4.5 Prioridad de operadores, evaluación de expresiones.

4.6 Estructura básica de un programa.

4.7 Proceso de creación de un ejecutable

5. Implementación de la clase.

5.1 Modificadores de acceso (Public, Private).

5.2 Encapsulamiento de la clase.

5.3 El método como elemento de la comunicación.

5.3.1 Sintaxis.

5.3.2 Concepto de parámetro.

5.3.3 Parámetros de salida y de entrada.

5.3.4 El constructor.

5.3.5 El destructor.

6. Estructuras secuenciales y selectivas.

6.1 Modificadores de acceso (Public, Private).

6.2 Entrada y salida de datos.

6.3 Interacción de la aplicación y la clase.

6.4 Estructuras selectivas.

6.4.1 Selectiva simple (si).

6.4.2 Selectiva doble (si / de otro modo).

6.4.3 Selectiva anidada.

6.4.4 Selectiva múltiple.

6.4.5 Selectiva Intenta (try/catch).

7. Estructuras de repetición.



7.1 Repetir mientras Selectiva simple (si).

7.2 Repetir hasta.

7.3 Repetir desde.

VII. Acervo bibliográfico