Actividades Académicas Básicas
(Obligatorias)

| | | |
|---|---|---|
| | **UNIVERSIDAD NACIONAL AUTÓNOMA DE MÉXICO**<br>**PROGRAMA DE POSGRADO**<br>**POSGRADO EN CIENCIA E INGENIERÍA DE LA COMPUTACIÓN**<br>**Programa de actividad académica** | |

| | | | | |
|---|---|---|---|---|
| **Denominación:** ARQUITECTURA DE COMPUTADORAS - | | | | |
| **Clave:** | **Semestre(s): 1** | **Campo de Conocimiento:**<br>Teoría de la Computación<br>Ingeniería de Software y Bases de Datos<br>Inteligencia Artificial<br>Computación Científica<br>Señales, Imágenes y Ambientes Virtuales<br>Redes y Seguridad en Cómputo | | **No. Créditos:** 6 |
| **Carácter:** Obligatoria | | **Horas** | **Horas por semana** | **Horas al Semestre** |
| **Tipo:** Teórica | | **Teoría:** 3 | **Práctica:** 0 | 3 | 48 |
| **Modalidad:** Curso | | **Duración del programa:** Semestral | | |

**Objetivo general:**
El alumno comprenderá la organización de las computadoras digitales. Se pretende diseñar desde el punto de vista lógico los diferentes componentes de una computadora, específicamente se diseñara una copia de un microprocesador comercial. El alumno comprenderá también la diferencia entre las computadoras tipo CISC y RISC.

**Objetivos específicos:**
Al final del curso el alumno será capaz de:

1. Describir los elementos constitutivos y fundamentales de una computadora, así como su funcionamiento lógico y físico.
2. Construir máquinas de estados usando memorias.
3. Construir máquinas de estados usando secuenciadores
4. Diseñar los componentes básicos de un microprocesador
5. Diseñar un microprocesador comercial tipo CISC, el 6811, usando los componentes diseñados en la unidad anterior
6. Aprender el concepto de procesamiento encauzado (PIPILINE) en las computadoras tipo RISC
7. Describir las computadoras paralelas y las supercomputadoras

**Índice Temático**

| Unidad | Tema | Horas | |
|---|---|---|---|
| | | Teóricas | Prácticas |
| 1 | Conceptos basicos | 7 | 0 |
| 2 | Construcción de máquinas de estados usando memorias | 7 | 0 |
| 3 | Construcción de máquinas de estados usando secuenciadores | 7 | 0 |
| 4 | Componentes básicos de un procesador. | 7 | 0 |
| 5 | Diseño de un procesador cisc comercial de 8 bits. | 7 | 0 |
| 6 | Computadoras tipo risc. | 7 | 0 |
| 7 | Computadoras paralelas | 6 | 0 |
| | Total de horas: | 48 | 0 |
| | Suma total de horas: | 48 | |

**Contenido Temático**

| Unidad | Tema y Subtemas |
|---|---|
| **1** | CONCEPTOS BASICOS<br><br>1.1 ESTRUCTURA BÁSICA DE UNA COMPUTADORA<br>1.2 TIPOS DE COMPUTADORAS<br>1.2.1 Computadoras secuenciales (CISC)<br>1.2.2 Computadoras paralelas<br>1.2.3 ProcesadoresEncauzados(RISC)<br>1.3 EL CONTROLADOR DE LA COMPUTADORA |

| | |
|---|---|
| **2** | CONSTRUCCIÓN DE MÁQUINAS DE ESTADOS USANDO MEMORIAS<br><br>2.1. DIRECCIONAMIENTO POR TRAYECTORIA<br>2.2. DIRECCIONAMIENTO ENTRADA-ESTADO<br>2.3. DIRECCIONAMIENTO IMPLÍCITO |
| **3** | CONSTRUCCIÓN DE MÁQUINAS DE ESTADOS USANDO SECUENCIADORES<br><br>3.1 EL SECUENCIADOR BÁSICO<br>3.2 SECUENCIADORES Y MEMORIAS<br>3.3 IMPLANTACIÓN DE CARTAS ASM USANDO SECUENCIADORES |
| **4** | COMPONENTES BÁSICOS DE UN PROCESADOR.<br><br>4.1 UNIDAD DE CONTROL DE LA COMPUTADORA<br>4.2 UNIDAD DE PROCESOS ARITMÉTICOS<br>4.3 REGISTROS INTERNOS<br>4.4 UNIDAD DE CONTROL DE PROGRAMA<br>4.5 REGISTRO DE ESTADOS O BANDERAS<br>4.6 UNIDAD DE CONTROL DE INTERRUPCIONES. |
| **5** | DISEÑO DE UN PROCESADOR CISC COMERCIAL DE 8 BITS.<br><br>5.1 ARQUITECTURA DEL MICROPROCESADOR 6811<br>5.2 TIPOS DE INSTRUCCIONES<br>5.3 MICROPROGRAMACIÓN |
| **6** | COMPUTADORAS TIPO RISC.<br><br>6.1 Definición de procesamiento encauzado (PIPELINE).<br>6.2 Procesamiento encauzado (PIPELINE) en el diseño de ALUs<br>6.3 Procesamiento encauzado (PIPELINE) en los procesadores<br>6.3.1 Microprocesador MIPS<br>6.3.2 Microprocesador Power PC<br>6.3.3 Microprocesador PIC<br>6.4 Restricciones de diseño.<br>6.4.1 Dependencias entre los registros<br>6.4.2 Saltos condicionales |
| **7** | COMPUTADORAS PARALELAS<br><br>7.1 Características generales.<br>7.2 Arquitecturas SIMD (Matriciales)<br>7.3 Arquitecturas MISD (Superpipeline)<br>7.4 Arquitecturas MIMD (Interconectadas) |

**Bibliografía Básica:**


- Savage, Jesús y Vazquez, Gabriel , *Diseño de Microprocesadores*, Facultad de Ingeniería, UNAM, México, 2004.
- Hennessy, John L. and Patterson David A., *Computer organization & design: The hardware-software interface*, Kaufmann Publishers, Inc., San Francisco, 1994.
- Lynch, Michel A., *Microprogrammed state machine design*, CRC Press, Boca Raton, 1993.
- Mano, Morris M., *Computer engineering hardware design*, Prentice-Hall, Englewood Cliffs, 1988.
- Mano, Morris M., *Computer system architecture*, Prentice-Hall, Englewood Cliffs, 1999.
- Mick, J. and Brick, *Bit-slice microprocessor design*, McGraw-Hill, New York, 1980.


**Bibliografía Complementaría:**


- Coelho, David R., *The VHDL handbook,* Kluwer Academic Publishers, Boston, 1989.
- Coelho, David R., *The VHDL handbook*, Kluwer Academic Publishers, Boston, 1989.
-Altera Corporation, *User configurable logic data book*, Santa Clara,1988
-Altera Corporation, *MAX7000 programmable logic device family data sheet*, ,
-Xilinx Corporation, *User configurable logic data book*, ,

| **Sugerencias didácticas:** | | **Mecanismos de evaluación de aprendizaje de los alumnos:** | |
|---|---|---|---|
| Exposición oral | (X) | Exámenes Parciales | (X) |
| Exposición audiovisual | (X) | Examen final escrito | (X) |
| Ejercicios dentro de clase | (X) | Trabajos y tareas fuera del aula | (X) |
| Ejercicios fuera del aula | (X) | Exposición de seminarios por los alumnos | (X) |
| Seminarios | (X) | Participación en clase | (X) |
| Lecturas obligatorias | (X) | Asistencia | (X) |
| Trabajo de Investigación | (X) | Seminario | ( ) |
| Prácticas de taller o laboratorio | (X) | Otras: | |
| Prácticas de campo | ( ) | | |
| Otros: | | | |

**Línea de investigación:**
Computación Científica
Ingeniería de Software y Bases de Datos
Inteligencia Artificial
Redes y Seguridad en Cómputo
Señales, Imágenes y Ambientes Virtuales
Teoría de la Computación

**Perfil profesiográfico:**
Académico con nivel mínimo de maestría, deseablemente doctorado, en un área afín a la ciencia e ingeniería de la computación. Con experiencia docente comprobable.

| | | |
|---|---|---|
| | **UNIVERSIDAD NACIONAL AUTÓNOMA DE MÉXICO**<br>**PROGRAMA DE POSGRADO**<br>**POSGRADO EN CIENCIA E INGENIERÍA DE LA COMPUTACIÓN**<br>**Programa de actividad académica** | |

| | | | | |
|---|---|---|---|---|
| **Denominación:** AUTOMATAS Y LENGUAJES FORMALES | | | | |
| **Clave:** | **Semestre(s): 1** | **Campos de Conocimiento:**<br>Teoría de la Computación<br>Ingeniería de Software y Bases de Datos<br>Inteligencia Artificial<br>Computación Científica<br>Señales, Imágenes y Ambientes Virtuales<br>Redes y Seguridad en Cómputo | | **No. Créditos:** 6 |
| **Carácter:** Obligatoria | | **Horas** | **Horas por semana** | **Horas al Semestre** |
| **Tipo:** Teórica | | **Teoría:** 3 | **Práctica:** 0 | 3 | 48 |
| **Modalidad:** Curso | | **Duración del programa:** Semestral | | |

**Seriación:** Sin Seriación ( X ) Obligatoria ( ) Indicativa ( )

**Actividad académica antecedente:**
**Actividad académica subsecuente:**

**Objetivo general:**
El alumno manejará conocimientos tanto de principios básicos como avanzados de autómatas y lenguajes formales

**Objetivos específicos:**
Al término del curso el alumno será capaz de:

1. Analizar los lenguajes regulares y autómatas finitos
2. Analizar los lenguajes libres de contexto y autómatas de pila
3. Analizar los lenguajes recursivamente enumerables y máquinas de Turing
4. Aplicar los conceptos de decidibilidad
5. Desarrollar aplicaciones de lenguajes formales

**Índice Temático**

| Unidad | Tema | Horas | |
|---|---|---|---|
| | | Teóricas | Prácticas |
| 1 | Lenguajes regulares y autómatas finitos | 10 | 0 |
| 2 | Lenguajes libres de contexto y autómatas de pila | 10 | 0 |
| 3 | Lenguajes recursivamente enumerables y máquinas de Turing | 10 | 0 |
| 4 | Decidibilidad | 9 | 0 |
| 5 | Aplicaciones de lenguajes formales | 9 | 0 |
| | Total de horas: | 48 | 0 |
| | Suma total de horas: | 48 | |

**Contenido Temático**

| Unidad | Tema y Subtemas |
|---|---|
| **1** | Lenguajes regulares y autómatas finitos<br><br>1.1 Autómatas finitos deterministas, no deterministas y otras variantes<br>1.2 Expresiones regulares y su equivalencia con autómatas finitos |
| **2** | Lenguajes libres de contexto y autómatas de pila<br><br>2.1 Gramáticas libres de contexto<br>2.2 Autómatas de pila |
| **3** | Lenguajes recursivamente enumerables y máquinas de Turing |

| | |
|---|---|
| | 3.1 Máquinas de Turing<br>3.2 Lenguajes recursivos y recursivamente enumerables<br>3.3 Equivalencia de las máquinas de Turing y las gramáticas sin restricciones |
| **4** | Decidibilidad<br><br>4.1 Problemas indecidibles<br>4.2 Teorema de Rice<br>4.3 Clasificación de problemas indecidibles |
| **5** | Aplicaciones de lenguajes formales<br><br>Contenido:<br>5.1 Aplicaciones en compiladores<br>5.2 Aplicaciones en procesamiento de lenguajes naturales<br>5.3 Aplicaciones en teoría de la concurrencia<br>5.4 Otras aplicaciones |

**Bibliografía Básica:**


- Dexter C. Kozen , *Automata and Computability. Undergraduate Texts in Computer Science*, Springer,1997.
- Michael Sipser, *Introduction to the Theory of Computation*, PWS Publishing Company, 1997.
- Thomas A. Sudkamp y , et_al., *Languages and Machines*, Addison Wesley, , 1997.
- Song Y Yan y , et_al., *Formal Languages and Machine Computation*, World Scientific, 1, 1998.


**Bibliografía Complementaría:**


- Martin Davis, *Computability, Complexity and Languages: Fundamentals of Theoretical Computer Science*, Academic Press, 1983.
- J.E. Hopcroft and J. Ullman, *Introduction to Automata Theory, Languages, and Computation*, Addison Wesley,1979.
- G. Rozenberg and A. Salomaa, editores., *Handbook of Formal Languages, volumen 1*, Springer, 1997.
- G. Rozenberg and A. Salomaa, editores, *Handbook of Formal Languages, volumen 2*, Springer, 1997.


| **Sugerencias didácticas:** | | **Mecanismos de evaluación de aprendizaje de los alumnos:** | |
|---|---|---|---|
| Exposición oral | (X) | Exámenes Parciales | (X) |
| Exposición audiovisual | ( ) | Examen final escrito | (X) |
| Ejercicios dentro de clase | (X) | Trabajos y tareas fuera del aula | (X) |
| Ejercicios fuera del aula | (X) | Exposición de seminarios por los alumnos | ( ) |
| Seminarios | (X) | Participación en clase | ( ) |
| Lecturas obligatorias | ( ) | Asistencia | ( ) |
| Trabajo de Investigación | (X) | Seminario | ( ) |
| Prácticas de taller o laboratorio | ( ) | Otras: | |
| Prácticas de campo | ( ) | | |
| Otros: | | | |

**Línea de investigación:**
Teoría de la Computación

**Perfil profesiográfico:**
Académico con nivel mínimo de maestría, deseablemente doctorado, en un área afín a la ciencia e ingeniería de la computación. Con experiencia docente comprobable.

**UNIVERSIDAD NACIONAL AUTÓNOMA DE MÉXICO**
**PROGRAMA DE POSGRADO**
**POSGRADO EN CIENCIA E INGENIERÍA DE LA COMPUTACIÓN**
**Programa de actividad académica**

| | | | | |
|---|---|---|---|---|
| **Denominación:** DISEÑO Y ANALISIS DE ALGORITMOS | | | | |
| **Clave:** | **Semestre(s): 1** | **Campos de Conocimiento:**<br>Teoría de la Computación<br>Inteligencia Artificial<br>Computación Científica<br>Señales, Imágenes y Ambientes Virtuales<br>Ingeniería de Software y Bases de Datos<br>Redes y Seguridad en Cómputo | | **No. Créditos:** 6 |
| **Carácter:** Obligatoria | | **Horas** | **Horas por semana** | **Horas al Semestre** |
| **Tipo:** Teórica | | **Teoría:** 3 | **Práctica:** 0 | 3 | 48 |
| **Modalidad:** Curso | | **Duración del programa:** Semestral | | |

**Seriación:** Sin Seriación ( X ) Obligatoria ( ) Indicativa ( )

**Actividad académica antecedente:**
**Actividad académica subsecuente:**

**Objetivo general:**
El alumno manejará con efectividad conocimientos y habilidades técnicas avanzadas en diseño y análisis de algoritmos.

**Objetivos específicos:**
**Al final del curso el alumno será capaz de :**

1. Presentar los conceptos fundamentales del diseño y análisis de algoritmos.
2. Implementar algoritmos de ordenación y búsquedas.
3. Aplicar la programación dinámica.
4. Aplicar los algoritmos greedy.
5. Analizar los algoritmos en gráficas.
6. Definir y analizar los problemas NP.

**Índice Temático**

| Unidad | Tema | Horas | |
|---|---|---|---|
| | | Teóricas | Prácticas |
| 1 | Introducción | 8 | 0 |
| 2 | Ordenación y Búsquedas | 8 | 0 |
| 3 | Programación Dinámica | 8 | 0 |
| 4 | Algoritmos Greedy | 8 | 0 |
| 5 | Algoritmos en Gráficas | 8 | 0 |
| 6 | Problemas NP | 8 | 0 |
| | Total de horas: | 48 | 0 |
| | Suma total de horas: | 48 | |

**Contenido Temático**

| Unidad | Tema y Subtemas |
|---|---|
| **1** | Introducción<br>1.1 Algoritmos<br>1.2 Complejidad<br>1.3 Algunos ejemplos |
| **2** | Ordenación y Búsquedas<br>2.1 Cota inferior<br>2.2 MergeSort<br>2.3 HeapSort<br>2.4 QuickSort<br>2.5 Ordenación en tiempo lineal<br>2.6 Búsquedas binarias<br>2.7 Árboles binarios<br>2.8 Mediana y k-selección |

| | |
|---|---|
| **3** | Programación Dinámica<br><br>3.1 Elementos de la programación dinámica<br>3.2 Multiplicación de matrices<br>3.3 Subsucesión común mas larga |
| **4** | Algoritmos Greedy<br>4.1 La estrategia greedy<br>4.2 Códigos de Huffman |
| **5** | Algoritmos en Gráficas<br><br>5.1 Representación de Gráficas<br>5.2 Búsquedas en profundidad y amplitud<br>5.3 Árbol generador de peso mínimo<br>5.4 La ruta más corta<br>5.5 Flujos en redes |
| **6** | Problemas NP<br><br>6.1 Tiempo polinomial<br>6.2 NP-Completez<br>6.3 Reducción de problemas<br>6.4 Pruebas de NP-Completez<br>6.5 Lista de problemas NP-completos<br><br>Se cubrirán opcionalmente algunos temas extras de relevancia sobre el Diseño y Análisis de Algoritmos como:<br><br>Geometría Computacional<br>Algoritmos Distribuidos<br>Algoritmos Paralelos<br>Estructuras de Datos Dinámicas |

**Bibliografía Básica:**

T.H. Cormen, C.E. Leiserson, R. L. Riverst, C. Stein. Introduction to Algorithms, 2nd Ed. The MIT Press, 2003.
J. Kleinberg, E. Tardos. Algorithm Design. Addison-Wesley. 2006.
A.V. Aho, J.E. Hopcroft, J. D. Ullman. Data Strcutures and Algorithms. Addison-Wesley, 1983.

**Bibliografía Complementaría:**

J.A. Bondy, U.S.R. Murty. Graph Theory with Applications. American Elseiver, 1976.
F.P. Preparata, M.I. Shamos. Computational Geometry: An Introduc¬tion. Springer-Verlag, 1985.
M. de Berg, M. van Kreveld, M. Overmars, 0. Schwarzkopf. Compu¬tational Geometry (3rd ed.), Springer Verlag, Berlin, 2008.

**Sugerencias didácticas:**

| | |
|---|---|
| Exposición oral | ( ) |
| Exposición audiovisual | ( ) |
| Ejercicios dentro de clase | ( ) |
| Ejercicios fuera del aula | ( ) |
| Seminarios | ( ) |
| Lecturas obligatorias | (X) |
| Trabajo de Investigación | ( ) |
| Prácticas de taller o laboratorio | ( ) |
| Prácticas de campo | ( ) |
| Otros: | |

**Mecanismos de evaluación de aprendizaje de los alumnos:**

| | |
|---|---|
| Exámenes Parciales | (X) |
| Examen final escrito | (X) |
| Trabajos y tareas fuera del aula | ( ) |
| Exposición de seminarios por los alumnos | ( ) |
| Participación en clase | ( ) |
| Asistencia | (X) |
| Seminario | ( ) |
| Otras: | |

**Línea de investigación:**
Teoría de la Computación

**Perfil profesiográfico:**
Académico con nivel mínimo de maestría, deseablemente doctorado, en un área afín a la ciencia e ingeniería de la computación. Con experiencia docente comprobable.

| | | |
|---|---|---|
| | **UNIVERSIDAD NACIONAL AUTÓNOMA DE MÉXICO**<br>**PROGRAMA DE POSGRADO**<br>**POSGRADO EN CIENCIA E INGENIERÍA DE LA COMPUTACIÓN**<br>**Programa de actividad académica** | |

| | | | | |
|---|---|---|---|---|
| **Denominación:** PROGRAMACION AVANZADA | | | | |
| **Clave:** | **Semestre(s): 1** | **Campo de Conocimiento:**<br>Teoría de la Computación<br>Ingeniería de Software y Bases de Datos<br>Inteligencia Artificial<br>Computación Científica<br>Señales, Imágenes y Ambientes Virtuales<br>Redes y Seguridad en Cómputo | | **No. Créditos:** 6 |
| **Carácter:** Obligatoria | | **Horas** | **Horas por semana** | **Horas al Semestre** |
| **Tipo:** Teórica | | **Teoría:** 3 | **Práctica:** 0 | 3 | 48 |
| **Modalidad:** Curso | | **Duración del programa:** Semestral | | |

**Seriación:** Sin Seriación ( X ) Obligatoria ( ) Indicativa ( )
**Actividad académica antecedente:**
**Actividad académica subsecuente:**

**Objetivo general:**
El alumno dominará los conceptos y técnicas avanzadas de programación, centrando la atención en el paradigma de programación orientada a objetos, concurrencia y el modelo cliente-servidor. Los conceptos se implementarán con el lenguaje de modelado unificado UML.

**Objetivos específicos:**
Al término del curso el alumno será capaz de aplicar:

1. Los distintos tipos de lenguajes que existen para programar.
2. Los conceptos básicos de programación orientada a objetos.
3. Los conceptos básicos de concurrencia y sincronización, así como los de distribución de objetos en red.
4. El diseño y programación de una aplicación en internet (WEB) utilizando la arquitectura Modelo-Vista-Controlador (MVC), el lenguaje de programación Java y acceso a base de datos.

**Índice Temático**

| Unidad | Tema | Horas | |
|---|---|---|---|
| | | Teóricas | Prácticas |
| 1 | Introducción a tipos de lenguajes de programación. | 3 | 0 |
| 2 | Conceptos de programación orientada a objetos. | 12 | 0 |
| 3 | Concurrencia y distribución. | 15 | 0 |
| 4 | Programación de aplicaciones WEB. | 18 | 0 |
| | Total de horas: | 48 | 0 |
| | Suma total de horas: | 48 | |

**Contenido Temático**

| Unidad | Tema y Subtemas |
|---|---|
| **1** | Introducción a tipos de lenguajes de programación.<br><br>1.1 Lenguajes compilados.<br>1.2 Lenguajes interpretados<br>1.3 Lenguajes de ?script?<br>1.4 Lenguajes declarativos.<br>1.5 Lenguajes visuales |
| **2** | Conceptos de programación orientada a objetos. |

| | |
|---|---|
| | 1.1 Clases.<br>1.2 Objetos.<br>1.3 Herencia.<br>1.4 Relaciones de uso.<br>1.5 Generadores.<br>1.6 Iteradores.<br>1.7 Manejo de excepciones.<br>1.8 Paquetes (módulos).<br>1.9 Interfaces. |
| **3** | Concurrencia y distribución.<br><br>3.1 Hilos de control ( threads ).<br>3.2 Sincronización.<br>3.3 Tipos de datos para la concurrencia (colas concurrentes).<br>3.4 Conexión a servidores.<br>3.5 Invocación a métodos remotos.<br>3.6 Arquitectura cliente-servidor. |
| **4** | Programación de aplicaciones WEB.<br><br>4.1 La arquitectura MVC ( Model-View-Controller ).<br>4.2 Servlets y JSP´s.<br>4.3 Conexión a base de datos. |

**Bibliografía Básica:**


- Booch, G., Maksimchuk, R. A., Engel, M. W., Young, *Object-Oriented Analysis and Design with Applications*, Addison-Wesley, 3rd edition, 2007.
- Booch, G., Rumbaugh, J., and Jacobson, I., *Unified Modeling Language User Guide*, Addison-Wesley, 2005.
- Horstmann, C. S. and Cornell, G., *Core Java 2: Vol II Advanced Features.* Pearson Prentice Hall.,2005.
- John Allen Robinson, *Software design for engineers and scientists*, Newnes.2004.
- Smith, M., *Java: an Object-Oriented Language*, Mac Graw Hill, 2000.
- Brian Kernigan & P. J. Plauger, *The elements of programming style*, Mcgraw-Hill 2nd edition, 1978.


**Bibliografía Complementaría:**


- Aho, A. V., Lam, M. S., Sethi, R., and Ullman, JD *Compilers, principles, techniques and tools*, Addison Wesley, second edition, 2007.
- Herlihy, M. and Shavit, N., *The Art of Multiprocessor Programming*, Morgan Kaufmann,2008.
- Kernighan, B. W. and Pike, R., *The Practice of Programming. Addison-Wesley Professional Computing Series*, Addison-Wesley,1999.
- Martelli, A., *Python in a Nutshell*, O'Reilly Media, Inc., 2nd edition, 2006.
- Steve McConnell, *Code Complete*, Microsoft-Press 2nd edition,2004.
- M. A. Jackson, *Principles on Programming Design*, Academic Press.
- Bundi Kurnhiawn, *Java for de Web with Servlets, JSP and EJB*, Sams, 2004.
- Alan R. Williamson and Ceri L. Moran, *Java Database Programming: Servlet & JDBC*, Prentice Hall., 1998.

| **Sugerencias didácticas** | | **Mecanismos de evaluación de aprendizaje de los alumnos:** | |
|---|---|---|---|
| Exposición oral | (X) | Exámenes Parciales | (X) |
| Exposición audiovisual | (X) | Examen final escrito | (X) |
| Ejercicios dentro de clase | (X) | Trabajos y tareas fuera del aula | (X) |
| Ejercicios fuera del aula | (X) | Exposición de seminarios por los alumnos | ( ) |
| Seminarios | (X) | Participación en clase | (X) |
| Lecturas obligatorias | (X) | Asistencia | ( ) |
| Trabajo de Investigación | ( ) | Seminario | ( ) |
| Prácticas de taller o laboratorio | (X) | Otras: | |
| Prácticas de campo | ( ) | | |
| Otros | | | |

**Línea de investigación:**
Ingeniería de Software y Bases de Datos

**Perfil profesiográfico:**
Académico con nivel mínimo de maestría, deseablemente doctorado, en una area afín a la ciencia e ingeniería de la computación. Con experiencia docente comprobable.