



**Universidad Autónoma de Nayarit**  
Área Económico-Administrativa  
Licenciatura en Informática

**1. DATOS DE IDENTIFICACIÓN**

**NOMBRE Y CLAVE DE LA UNIDAD DE APRENDIZAJE**

|                                 |  |
|---------------------------------|--|
| Análisis y Diseño de Algoritmos |  |
|---------------------------------|--|

**DOCENTE(S) RESPONSABLE(S)**

|  |
|--|
|  |
|--|

| TIPO DE UNIDAD DE APRENDIZAJE | ACADEMIA     |
|-------------------------------|--------------|
| Obligatoria                   | Programación |

| ÁREA DE FORMACIÓN | LÍNEA DE FORMACIÓN                    | T.U.D.C.       |
|-------------------|---------------------------------------|----------------|
| Disciplinar       | Programación e Ingeniería de Software | Curso - Taller |

| Horas teoría | Horas práctica | Horas de estudio independiente | Total de horas | Valor en créditos |
|--------------|----------------|--------------------------------|----------------|-------------------|
| 40           | 39             | 17                             | 96             | 6                 |

| FECHA DE ELABORACIÓN | FECHA DE ACTUALIZACIÓN |
|----------------------|------------------------|
| Junio 2012           |                        |

| ELABORADO POR:   | ACTUALIZADO POR: |
|--|------------------|
| Mtro. Rubén Paul Benítez Cortes<br>Mtro. Marco Antonio Fernández Zepeda<br><br>Revisado por:<br>Comité Curricular de Informática |                  |

## 2. PRESENTACIÓN

Cuando usamos una computadora, estamos intentando resolver un problema. Puede ser un problema de negocios que involucra ganancia y pérdida; un problema científico que emplea modelos de comportamiento físico; una investigación estadística que evalúa la posibilidad de que ocurra algún evento; un ejercicio de lingüística en la interpretación del lenguaje natural, o sólo simple procesamiento de texto. La gente resolvía problemas mucho antes de que las computadoras llegaran a ser algo común, lo cual dio como resultado un tesoro de experiencia para beneficiar hoy.

Los programas de computadora se utilizan para resolver problemas, y ha habido miles de años de trabajo en matemáticas para este fin. Los lenguajes de programación están especificados por reglas para formar instrucciones correctas, organizándolas en módulos, someterlas hacia un compilador, el cual traduce el código en un lenguaje comprensible para una computadora en particular, y finalmente ejecutar el programa; es decir, someter la entrada hacia la computadora, la cual la transforma en una salida de acuerdo con las instrucciones en el programa.

De ahí que el Comité Curricular del Programa Académico de Informática plantee la necesidad de implementar una unidad de aprendizaje que desarrolle estas competencias; denominando esta unidad de aprendizaje como Análisis y Diseño de Algoritmos, para impartirse en el área de formación disciplinar del plan de estudios.

Este curso – taller se enfoca en la solución de problemas, los cuales deben ser resueltos a partir del análisis y diseño de algoritmos computacionales. Este curso de programación promueve el pensamiento algorítmico de los estudiantes, para analizar y diseñar soluciones a problemas computacionales.

Esta unidad de aprendizaje corresponde al área disciplinar y cuenta con un valor curricular de 6 créditos con un total de 79 horas presenciales las que se dividirán en sesiones de 5 horas a la semana.

## 3. UNIDAD DE COMPETENCIA

**Objeto de estudio:** Análisis y Diseño de Algoritmos como un elemento clave en programación.

**Competencia:** Capacidad analítica, organizada y de toma de decisiones para desarrollar software para las organizaciones públicas, sociales y privadas.

## 4. SABERES

|                  |   |
|------------------|---|
| Saberes Teóricos | Conocer el proceso para la solución de problemas.<br>Conocer la importancia de los algoritmos para la solución de problemas.<br>Conocer las principales técnicas de diseño de algoritmos computacionales. |
|------------------|---|

|                       |   |
|-----------------------|---|
| Saberes Prácticos     | Aplicar estrategias para la solución de problemas.<br>Diseñar y analizar algoritmos empleando distintas técnicas.<br>Distinguir y utilizar, con base al problema, la técnica de diseño más apropiada que debe emplearse en la solución.<br>Implementar algoritmos en un lenguaje de programación de computadoras. |
| Saberes Metodológicos | Relacionar el proceso de análisis y diseño de algoritmos computacionales con el desarrollo de programas a través de un lenguaje de programación de computadoras. Analizar, diseñar y desarrollar algoritmos y programas con las técnicas y conceptos de un paradigma de programación.                             |
| Saberes Formativos    | Análisis y síntesis, pensamiento abstracto, resolución de problemas, planificación y organización, pensamiento crítico, preparación y presentación de documentación técnica, expresión oral y escrita.  |

## 5. DESGLOSE DE CONTENIDO (temática)

### **Unidad 1 Resolución de problemas con computadora**

- 1.1 La programación y sus paradigmas
- 1.2 La importancia de la programación
- 1.3 Evolución histórica de los lenguajes de programación
- 1.4 Los paradigmas de la programación
- 1.5 Abstracción y solución de problemas
- 1.6 El proceso de resolución de un problema
- 1.7 Análisis
- 1.8 Diseño
- 1.9 Construcción

### **Unidad 2 Fundamentos de programación**

- 2.1 Conceptos básicos de programación
- 2.2 Diagramas de Flujo y/o pseudocódigo
- 2.3 Tipos de datos primitivos
- 2.4 Identificadores y palabras reservadas
- 2.5 Variables, literales y constantes
- 2.6 Operadores y operandos
- 2.7 Expresiones y evaluación de operaciones
- 2.8 Estructura de un programa
- 2.9 Proceso de creación y ejecución de un programa

### **Unidad 3 Control de flujo de programas: Estructuras repetitivas**

- 3.1 Concepto de repetición de instrucciones
- 3.2 Repetición con condición inicial
- 3.3 Repetición con condición final
- 3.4 Repetición fija

### **Unidad 4 Control de flujo de programas: Estructuras selectivas**

- 4.1 Selección simple
- 4.2 Selección alternativa
- 4.3 Selección en cascada

- 4.4 Selección anidada
- 4.5 Selección múltiple
- 4.6 Combinación de estructuras

#### **Unidad 5 Tipos de datos compuestos**

- 5.1 Arreglos
- 5.2 Registros

#### **Unidad 6 Implementación de algoritmos**

- 6.1 Uso de Java para codificar los algoritmos del curso

## **6. ACCIONES**

### **Unidad I Resolución de problemas con computadora**

- Investigar conceptos
- Realizar investigación y presentar los resultados para la discusión grupal.
- Resolver casos de estudios
- Lecturas obligatorias

### **Unidad II Fundamentos de programación**

- Reporte de Lecturas
- Ejercicios prácticos
- Resolver casos de estudios
- Dinámicas grupales

### **Unidad III Control de flujo de programas: Estructuras repetitivas**

- Revisión de vídeos
- Resolver casos de estudios
- Realizar mesas redondas para discutir conceptos que se manejan en el tema.
- Elaborar mapas conceptuales
- Elaborar mapas mentales
- Dinámicas grupales

### **Unidad IV Control de flujo de programas: Estructuras selectivas**

- Exposiciones orales individuales y grupales
- Revisión de apoyos audiovisuales
- Dinámicas grupales
- Lecturas obligatorias
- Resolver casos de estudios
- Análisis de casos de estudio
- Trabajos de investigación

### **Unidad IV Control de flujo de programas: Estructuras selectivas**

- Exposiciones orales individuales y grupales
- Revisión de apoyos audiovisuales
- Dinámicas grupales

- Lecturas obligatorias
- Resolver casos de estudios
- Análisis de casos de estudio
- Trabajos de investigación

#### **Unidad V Tipos de datos compuestos**

- Exposiciones orales individuales y grupales
- Revisión de apoyos audiovisuales
- Dinámicas grupales
- Lecturas obligatorias
- Resolver casos de estudios
- Análisis de casos de estudio
- Trabajos de investigación

#### **Unidad VI Implementación de algoritmos**

- Exposiciones orales individuales y grupales
- Revisión de apoyos audiovisuales
- Dinámicas grupales
- Lecturas obligatorias
- Resolver casos de estudios
- Análisis de casos de estudio
- Trabajos de investigación

### **7. CAMPO DE APLICACIÓN**

Áreas de desarrollo de software en los departamentos de Informática en las empresas del sector privado y organismos.

### **8. SISTEMA DE EVALUACIÓN**

| <b>Evidencias de aprendizaje</b>        | <b>Criterios de desempeño</b>   |
|---|---|
| — Mapa conceptual y resumen por unidad. | — Elabora un mapa conceptual por cada unidad que incluya los conceptos claves de la referencia bibliográfica. |
| — Cuestionario por unidad               | — Realiza un resumen correspondiente a la unidad, con la amplitud indicada por el docente.                    |
| — Participación en clase.               | — Contesta el cuestionario correspondiente de cada unidad.  |
| — Examen departamental                  | — Participación en clase con aportaciones sustantivas relacionadas con el tema.                               |
| — Exposición por parte de los alumnos   | — La exposición de clase se realiza utilizando los medios adecuados para                                      |
| — Lecturas                              |   |
| — Mapas mentales                        |   |
| — Análisis de videos tutoriales         |   |
| — Análisis de los casos de estudio      |   |

|  |  |
|--|--|
|  | presentarla en forma visual ya sea por medio de acetatos o cañón. (No se permite leer el contenido). |
|--|--|

## 9. CRITERIOS DE CALIFICACIÓN

| Evidencias de aprendizaje           | Porcentaje |
|-------------------------------------|------------|
| Ejercicios                          | 20%        |
| Exámenes departamentales            | 40%        |
| Portafolio de evidencias            | 5%         |
| Resolución de casos de estudio      | 25%        |
| Participación en plenarios grupales | 10%        |

## 10. CRITERIOS DE ACREDITACIÓN

- Cumplir con el 80% de asistencias
- Calificación mínima para acreditar la unidad de aprendizaje será un acumulado de 60
- Los criterios escritos de los Trabajos de Investigación y la Resolución de casos de estudio son obligatorios a acreditarse con una calificación mínima de 60 c/u.

## 11. ACERVOS DE CONSULTA

### ACERVOS BÁSICOS

CEBALLOS, FRANCISCO JAVIER. **JAVA CURSO DE PROGRAMACIÓN**, 3ª ED., RAMA, ESPAÑA, 2007.

GARCÍA FÉLIX, CARRETERO JESÚS, FERNÁNDEZ JAVIER, CALDERÓN ALEJANDRO. **EL LENGUAJE DE PROGRAMACIÓN JAVA**, EDIT. PEARSON EDUCACIÓN, ESPAÑA, 2002.

ZHANG, TONY. **APRENDIENDO JAVA EN 24 HORAS**, EDIT. PEARSON EDUCACIÓN, ESPAÑA, 2001.

### ACERVOS COMPLEMENTARIOS

JOHN S. MURPHY. **FUNDAMENTOS DE PROGRAMACIÓN DE COMPUTADORAS DIGITALES**, EDITORIAL LIMUSA, 1984.

ROBERT L, KRUSE. **ESTRUCTURAS DE DATOS Y DISEÑO DE PROGRAMAS**, MÉXICO, PRENTICE HALL, 1998.

SHNERDER, G. MICHAEL. **INTRODUCCIÓN A LA PROGRAMACIÓN Y LA SOLUCIÓN DE PROBLEMAS CON PASCAL**, MÉXICO, LIMUSA, 1986.

TENENBAUM AARON M. **ESTRUCTURAS DE DATOS EN C**, PRENTICE HALL, 1993.

WIRTH, NIKLAUS. **ALGORITMOS Y ESTRUCTURAS DE DATOS**, MÉXICO, PRENTICE HALL, 1987.

## **12. PERFIL DE LOS DOCENTES PARTICIPANTES EN LA UNIDAD DE APRENDIZAJE:**

Docentes con Licenciatura o grado de Maestría que dominen el conocimiento de las áreas computacionales y dominio de programación estructurada y orientada a objetos.

Docentes con la experiencia mínima de 3 años que dominen los aspectos mencionados.

Constancia de cursos didácticos y disciplinados. Manejo de técnicas didácticas.



**Universidad Autónoma de Nayarit**

Área Económico-Administrativa

Licenciatura en Economía

### **1. DATOS DE IDENTIFICACIÓN**

**NOMBRE Y CLAVE DE LA UNIDAD DE APRENDIZAJE**

|  |  |
|--|--|
| Análisis y diseño de Sistemas de Información |  |
|--|--|

**DOCENTE(S) RESPONSABLE(S)**