

## 1. Datos Generales de la asignatura

<b>Nombre de la asignatura:</b>	<b>Análisis y Diseño de Algoritmos.</b>
<b>Especialidad:</b>	<b>Big Data Clave: ISIE-BIG-2019-02</b>
<b>Clave de la asignatura:</b>	<b>BIF-1901</b>
<b>SATCA<sup>1</sup>:</b>	<b>3-2-5</b>
<b>Carrera:</b>	<b>Ingeniería en Sistemas Computacionales</b>

## 2. Presentación

### Caracterización de la asignatura

Esta asignatura aporta al perfil del Ingeniero en Sistemas Computacionales las habilidades para diseñar, desarrollar y aplicar modelos computacionales para solucionar problemas, mediante la selección y uso de herramientas matemáticas.

Para integrar la materia se ha hecho un análisis del campo de los fundamentos del análisis y diseño de algoritmos. Estos temas son la base del cálculo del tiempo de ejecución de los algoritmos computacionales, que tienen una importante aplicación en el ámbito profesional de este ingeniero.

Esta asignatura forma parte de los módulos de especialidad de: Programación para dispositivos móviles y Big Data. Utiliza conocimientos de asignaturas anteriores del área de las ciencias básicas, de las materias de programación y estructura de datos, y al mismo tiempo da soporte, a través de estrategias formales de demostración, a otras asignaturas de la especialidad. Directamente se vincula con el desempeño profesional del egresado en el campo de las ciencias computacionales.

Se ubica dentro del plan de estudios de la carrera en el sexto semestre. De manera particular los conceptos revisados en esta asignatura se aplican en el estudio de las materias de las especialidades.

---

<sup>1</sup> Sistema de Asignación y Transferencia de Créditos Académicos

### **Intención didáctica**

Se organiza el temario en cinco unidades.

En la primera unidad se desarrolla el fundamento teórico del análisis de algoritmos iterativos y recursivos.

El resto de las unidades se estudian los patrones de diseño de algoritmos siguientes: algoritmos voraces, divide y vencerás, programación dinámica, retropropagación y poda. El profesor explicará las características de cada una de las estrategias y sus ventajas y desventajas en función de la complejidad de los algoritmos que producen.

La asignatura aporta al perfil del egresado habilidades matemáticas y metodológicas para el análisis y diseño de algoritmos aplicados a la solución de problemas en el ámbito del diseño y desarrollo de tecnología computacional. Adicionalmente cumple con los requerimientos de las asignaturas posteriores del plan de estudios, proporcionando las bases teóricas y metodológicas para las siguientes materias del módulo de especialidad.

El enfoque sugerido para la asignatura incorpora actividades prácticas que serán guiadas por el profesor para desarrollar habilidades para el análisis, diseño y la programación a través de la solución computacional de problemas sobre los conceptos teóricos revisados.

Mediante estas actividades prácticas se desarrollan habilidades para el trabajo en equipo; asimismo, se propician procesos intelectuales como inducción-deducción y análisis-síntesis con la intención de generar una actividad intelectual compleja. Por esta razón varias de las actividades prácticas se han descrito como actividades previas al tratamiento teórico de los temas, de manera que no sean una mera corroboración de lo visto previamente en clase, sino una oportunidad para conceptualizar a partir de lo observado. En las actividades prácticas sugeridas, es conveniente que el profesor guíe mediante preguntas a los alumnos para que ellos hagan la elección de las variables a controlar y registrar.

La lista de actividades de aprendizaje sugeridas no es exhaustiva y puede ser actualizada y complementada en función de experiencias particulares y de acuerdo a la evolución del grupo. Mientras que la implementación de las actividades sugeridas puede desarrollarse como actividad extra clase, comenzando el tratamiento en clase a partir de la discusión de los resultados de las observaciones preliminares. Es importante ofrecer escenarios distintos, ya sean contruados, artificiales, virtuales o naturales.

Las actividades de aprendizaje sugeridas incluyen la formalización de los conceptos. Para este fin, se plantea en primer término la solución de problemas seleccionados ya sea mediante herramientas de software para el cálculo matemático o mediante programas desarrollados en algún lenguaje de programación. Se busca que el alumno tenga el primer contacto con el concepto en forma concreta y sea a través de la observación, la reflexión y la discusión que se dé la formalización. La resolución de problemas de aplicación más generales se hará después de este proceso. Esta resolución de problemas no se especifica en la descripción de actividades, por ser más familiar en el desarrollo de cualquier curso.

### 3. Participantes en el diseño y seguimiento curricular del programa

Lugar y fecha de elaboración o revisión	Participantes	Observaciones
Tecnológico Nacional de México - Instituto Tecnológico de Ciudad Madero. Unidad Académica de Sistemas y Computación del 15 al 19, del 25 al 30 de junio y el 1 de julio del 2015.	Dra. Guadalupe Castilla Valdez Dra. Laura Cruz Reyes Dra. Claudia G. Gómez Santillán Dr. Rodolfo Pazos Rangel Dr. Héctor Joaquín Fraire Huacuja. Dr. Nelson Rangel Valdez Dr. Juan Frausto Solís.	Elaborado durante el Curso y el Taller para el desarrollo de los Módulos de Especialidad para la carrera de Ingeniería en Sistemas Computacionales.
Tecnológico Nacional de México - Instituto Tecnológico de Ciudad Madero. Unidad Académica de Sistemas del 20 al 24 de Agosto del 2018.	Dra. Guadalupe Castilla Valdez Dra. Laura Cruz Reyes Dr. Rodolfo Pazos Rangel Dr. Héctor Joaquín Fraire Huacuja. Dr. Nelson Rangel Valdez Dr. Juan Frausto Solís. Dra. Claudia Guadalupe Gómez Santillán	Elaborado en el periodo intersemestral de Agosto 2018.

### 4. Competencia(s) a desarrollar

Competencia(s) específica(s) de la asignatura
Conoce y aplica los siguientes patrones de diseño de algoritmos: algoritmos voraces, divide y vencerás, programación dinámica, retropropagación y ramificación y acotamiento. Analizar formalmente la complejidad temporal de algoritmos iterativos y recursivos.

## 5. Competencias previas

- Analiza, diseña e implementa algoritmos computacionales para la solución de problemas del contexto en un lenguaje de programación de alto nivel.
- Conoce, aplica e implementa las estructuras de datos en la solución de problemas del contexto.

## 6. Temario

No.	Temas	Subtemas
1	Análisis de la complejidad algorítmica	1.1 Funciones asintóticas: definición (O-grande, omega y theta) y propiedades. 1.2 Complejidad algorítmica de algoritmos no recursivos. 1.3 Solución de relaciones de recurrencia (método de sustitución, árboles de recursión y Teorema Maestro). 1.4 Complejidad algorítmica de algoritmos irectivos 1.5 Análisis de la complejidad del algoritmo de burbuja 1.6 Análisis de la complejidad del algoritmo de inserción
2.	Análisis y diseño de algoritmos de divide y vencerás	2.1 Estrategia de divide y vencerás 2.2 Análisis de la complejidad del algoritmo de ordenamiento por mezcla 2.3 Análisis de la complejidad del algoritmo de ordenamiento rápido 2.4 Análisis de la complejidad del algoritmo de búsqueda binaria
3.	Análisis y diseño de algoritmos de búsqueda en árboles.	3.1 Estrategia de backtracking 3.2 Análisis de la complejidad del algoritmo de backtracking para el problema de la mochila. 3.3 Estrategia de ramificación y poda 3.4 Análisis de la complejidad del algoritmo de ramificación y poda para el problema de planificación de tareas
4.	Análisis y diseño de algoritmos de programación dinámica	4.1 Estrategia de programación dinámica 4.2 Análisis de la complejidad del algoritmo para el problema de determinar las rutas más cortas en grafos (Dijkstra) 4.3 Análisis de la complejidad del algoritmo para el problema del cambio 4.4 Análisis de la complejidad del algoritmo para el problema de la mochila
5.	Análisis y diseño de algoritmos voraces	5.1 Estrategia de los algoritmos voraces 5.2 Análisis de la complejidad del algoritmo voráz para el problema del cambio 5.3 Análisis de la complejidad del algoritmo voráz para el problema de la mochila

		5.4 Análisis de la complejidad del algoritmo voráz para el problema de planificación de tareas 5.5 Análisis de la complejidad del algoritmo voráz para el problema del agente viajero
--	--	--

## 7. Actividades de aprendizaje de los temas

Nombre de tema 1	
Análisis de la complejidad algorítmica	
Competencias	Actividades de aprendizaje
<p><b>Específica(s):</b></p> <p>Conoce e identifica herramientas matemáticas para determinar la eficiencia de algoritmos.</p> <p>Revisa y aplica herramientas matemáticas para determinar la eficiencia de diferentes algoritmos con respecto al tiempo de ejecución.</p> <p><b>Genéricas:</b></p> <p><u>Instrumentales:</u></p> <ul style="list-style-type: none"> <li>• Capacidad de análisis y síntesis.</li> <li>• Conocimientos generales básicos.</li> <li>• Comunicación oral y escrita.</li> <li>• Solución de problemas.</li> <li>• Conocimientos básicos de la carrera.</li> </ul> <p><u>Interpersonales:</u></p> <ul style="list-style-type: none"> <li>• Capacidad crítica y autocrítica.</li> <li>• Capacidad de trabajar en equipo.</li> </ul> <p><u>Sistemáticas:</u></p> <ul style="list-style-type: none"> <li>• Capacidad de aplicar los conocimientos en la práctica.</li> <li>• Habilidades de investigación.</li> <li>• Capacidad de aprender.</li> <li>• Habilidad para trabajar en forma autónoma.</li> </ul>	<ul style="list-style-type: none"> <li>• Revisar la definición de las funciones asintóticas (O- grande, omega y theta) y sus propiedades.</li> <li>• Revisar como se determina la complejidad de una secuencia de instrucciones, condicionales, ciclos y llamadas a funciones.</li> <li>• Resolver problemas para determinar la complejidad temporal de algoritmos no recursivos.</li> <li>• Revisar los métodos de solución de recurrencias: árboles de recursión, iterativo o sustitución y maestro.</li> <li>• Resolver problemas de solución de recurrencias con los métodos de árboles de recursión, iterativo o sustitución y maestro.</li> <li>• Resolver problemas para determinar la complejidad temporal de algoritmos recursivos.</li> <li>• Resolver problemas cuya solución requiere aplicar estrategias similares a las de los métodos de ordenamiento de burbuja o inserción.</li> </ul>

Nombre de tema 2	
Análisis y diseño de algoritmos de divide y vencerás	
Competencias	Actividades de aprendizaje
<p><b>Específica(s):</b></p> <p>Aplica técnicas de diseño de algoritmos para obtener soluciones eficientes.</p> <p><b>Genéricas:</b></p> <p><u>Instrumentales:</u></p> <ul style="list-style-type: none"> <li>• Capacidad de análisis y síntesis.</li> <li>• Conocimientos generales básicos.</li> <li>• Comunicación oral y escrita.</li> <li>• Solución de problemas.</li> <li>• Conocimientos básicos de la carrera.</li> </ul> <p><u>Interpersonales:</u></p> <ul style="list-style-type: none"> <li>• Capacidad crítica y autocrítica.</li> <li>• Capacidad de trabajar en equipo.</li> </ul> <p><u>Sistémicas:</u></p> <ul style="list-style-type: none"> <li>• Capacidad de aplicar los conocimientos en la práctica.</li> <li>• Habilidades de investigación.</li> <li>• Capacidad de aprender.</li> <li>• Habilidad para trabajar en forma autónoma.</li> </ul>	<ul style="list-style-type: none"> <li>• Revisar el enfoque de diseño de divide y vencerás.</li> <li>• Revisar la definición del problema de ordenamiento.</li> <li>• Revisar el algoritmo de ordenamiento por mezcla enfatizando la estrategia de diseño.</li> <li>• Realizar el análisis de la complejidad del algoritmo de ordenamiento por mezcla.</li> <li>• Revisar el algoritmo de ordenamiento rápido enfatizando la estrategia de diseño.</li> <li>• Realizar el análisis de la complejidad del algoritmo de ordenamiento rápido.</li> <li>• Revisar el algoritmo de búsqueda binaria enfatizando la estrategia de diseño.</li> <li>• Realizar el análisis de la complejidad del algoritmo de búsqueda binaria.</li> <li>• Resolver problemas cuya solución requiere aplicar la estrategia de divide y vencerás.</li> </ul>
Nombre de tema 3	
Análisis y diseño de algoritmos de búsqueda en árboles	
Competencias	Actividades de aprendizaje
<p><b>Específica(s):</b></p> <p>Aplica técnicas de diseño de algoritmos</p>	<ul style="list-style-type: none"> <li>• Revisar el enfoque de diseño de backtracking.</li> </ul>

<p>para obtener soluciones eficientes.</p> <p><b>Genéricas:</b></p> <p><u>Instrumentales:</u></p> <ul style="list-style-type: none"> <li>• Capacidad de análisis y síntesis.</li> <li>• Conocimientos generales básicos.</li> <li>• Comunicación oral y escrita.</li> <li>• Solución de problemas.</li> <li>• Conocimientos básicos de la carrera.</li> </ul> <p><u>Interpersonales:</u></p> <ul style="list-style-type: none"> <li>• Capacidad crítica y autocrítica.</li> <li>• Capacidad de trabajar en equipo.</li> </ul> <p><u>Sistémicas:</u></p> <ul style="list-style-type: none"> <li>• Capacidad de aplicar los conocimientos en la práctica.</li> <li>• Habilidades de investigación.</li> <li>• Capacidad de aprender.</li> <li>• Habilidad para trabajar en forma autónoma.</li> </ul>	<ul style="list-style-type: none"> <li>• Revisar la definición del problema de la mochila.</li> <li>• Resolver el problema de la mochila utilizando backtracking.</li> <li>• Realizar el análisis de la complejidad del algoritmo de solución dl problema de la mochila utilizando backtracking.</li> <li>• Resolver problemas cuya solución requiere aplicar la estrategia de backtracking.</li> <li>• Revisar la definición del problema de planificación de tareas.</li> <li>• Resolver el problema de planificación de tareas utilizando ramificación y acotamiento.</li> <li>• Realizar el análisis de la complejidad del algoritmo de solución dl problema de planificación de tareas utilizando ramificación y acotamiento.</li> <li>• Resolver problemas cuya solución requiere aplicar la estrategia de ramificación y acotamiento.</li> </ul>
<p>Nombre de tema 4</p> <p>Análisis y diseño de algoritmos de programación dinámica</p>	
<p>Competencias</p>	<p>Actividades de aprendizaje</p>
<p><b>Específica(s):</b></p> <p>Aplica técnicas de diseño de algoritmos para obtener soluciones eficientes.</p> <p><b>Genéricas:</b></p> <p><u>Instrumentales:</u></p> <ul style="list-style-type: none"> <li>• Capacidad de análisis y síntesis.</li> <li>• Conocimientos generales básicos.</li> </ul>	<ul style="list-style-type: none"> <li>• Revisar el enfoque de diseño de programación dinámica.</li> <li>• Revisar la definición del problema de determinar la ruta más corta en grafos (Dijkstra).</li> <li>• Resolver el problema de Dijkstra utilizando programación dinámica.</li> </ul>

<ul style="list-style-type: none"> <li>• Comunicación oral y escrita.</li> <li>• Solución de problemas.</li> <li>• Conocimientos básicos de la carrera.</li> </ul> <p><u>Interpersonales:</u></p> <ul style="list-style-type: none"> <li>• Capacidad crítica y autocrítica.</li> <li>• Capacidad de trabajar en equipo.</li> </ul> <p><u>Sistémicas:</u></p> <ul style="list-style-type: none"> <li>• Capacidad de aplicar los conocimientos en la práctica.</li> <li>• Habilidades de investigación.</li> <li>• Capacidad de aprender.</li> <li>• Habilidad para trabajar en forma autónoma.</li> </ul>	<ul style="list-style-type: none"> <li>• Realizar el análisis de la complejidad del algoritmo de solución del problema de Dijkstra utilizando programación dinámica.</li> <li>• Revisar la definición del problema del cambio.</li> <li>• Resolver el problema del cambio utilizando programación dinámica.</li> <li>• Realizar el análisis de la complejidad del algoritmo de solución del problema del cambio utilizando programación dinámica.</li> <li>• Resolver el problema de la mochila utilizando programación dinámica.</li> <li>• Realizar el análisis de la complejidad del algoritmo de solución del problema de la mochila utilizando programación dinámica.</li> <li>• Resolver problemas cuya solución requiere aplicar la estrategia de programación dinámica.</li> </ul>
<p align="center">Nombre de tema 5</p> <p align="center">Análisis y diseño de algoritmos voraces</p>	
Competencias	Actividades de aprendizaje
<p><b>Específica(s):</b></p> <p>Aplica técnicas de diseño de algoritmos para obtener soluciones eficientes.</p> <p><b>Genéricas:</b></p> <p><u>Instrumentales:</u></p> <ul style="list-style-type: none"> <li>• Capacidad de análisis y síntesis.</li> <li>• Conocimientos generales básicos.</li> <li>• Comunicación oral y escrita.</li> <li>• Solución de problemas.</li> <li>• Conocimientos básicos de la carrera.</li> </ul>	<ul style="list-style-type: none"> <li>• Revisar el enfoque de diseño de algoritmos voraces.</li> <li>• Resolver el problema de la mochila utilizando un algoritmo voraz.</li> <li>• Realizar el análisis de la complejidad del algoritmo voraz de solución del problema de la mochila.</li> <li>• Resolver el problema del cambio utilizando un algoritmo voraz.</li> <li>• Realizar el análisis de la complejidad</li> </ul>



<p><u>Interpersonales:</u></p> <ul style="list-style-type: none"> <li>• Capacidad crítica y autocrítica.</li> <li>• Capacidad de trabajar en equipo.</li> </ul> <p><u>Sistémicas:</u></p> <ul style="list-style-type: none"> <li>• Capacidad de aplicar los conocimientos en la práctica.</li> <li>• Habilidades de investigación.</li> <li>• Capacidad de aprender.</li> <li>• Habilidad para trabajar en forma autónoma.</li> </ul>	<p>del algoritmo de solución voraz del problema del cambio.</p> <ul style="list-style-type: none"> <li>• Resolver el problema de planificación de tareas utilizando un algoritmo voraz.</li> <li>• Realizar el análisis de la complejidad del algoritmo voraz de solución del problema de planificación de tareas.</li> <li>• Revisar la definición del problema del agente viajero.</li> <li>• Resolver el problema del agente viajero utilizando un algoritmo voraz.</li> <li>• Realizar el análisis de la complejidad del algoritmo voraz de solución del problema del agente viajero.</li> <li>• Resolver problemas cuya solución requiere aplicar la estrategia de algoritmos voraces.</li> </ul>
---	--

## 8. Práctica(s)

<ol style="list-style-type: none"> <li>1. Solución de problemas que requieren utilizar algoritmos de ordenamiento.</li> <li>2. Solución de problemas que requieren utilizar algoritmos de búsqueda.</li> <li>3. Solución de problemas que requieren utilizar la estrategia divide y vencerás.</li> <li>4. Solución de problemas que requieren utilizar backtracking.</li> <li>5. Solución de problemas que requieren utilizar ramificación y acotamiento.</li> <li>6. Solución de problemas que requieren utilizar programación dinámica.</li> <li>7. Solución de problemas que requieren utilizar algoritmos voraces.</li> </ol> <p>Se sugiere utilizar problemas similares a los utilizados en los sitios de entrenamiento para concursos de programación como el de la Universidad de Valladolid (<a href="https://uva.onlinejudge.org/">https://uva.onlinejudge.org/</a>).</p>
--

## 9. Proyecto de asignatura

<p>El objetivo del proyecto que planteé el docente que imparta esta asignatura, es demostrar el desarrollo y alcance de la(s) competencia(s) de la asignatura, considerando las</p>
---

siguientes fases:

- **Fundamentación:** marco referencial (teórico, conceptual, contextual, legal) en el cual se fundamenta el proyecto de acuerdo con un diagnóstico realizado, mismo que permite a los estudiantes lograr la comprensión de la realidad o situación objeto de estudio para definir un proceso de intervención o hacer el diseño de un modelo.
- **Planeación:** con base en el diagnóstico en esta fase se realiza el diseño del proyecto por parte de los estudiantes con asesoría del docente; implica planificar un proceso: de intervención empresarial, social o comunitario, el diseño de un modelo, entre otros, según el tipo de proyecto, las actividades a realizar los recursos requeridos y el cronograma de trabajo.
- **Ejecución:** consiste en el desarrollo de la planeación del proyecto realizada por parte de los estudiantes con asesoría del docente, es decir en la intervención (social, empresarial), o construcción del modelo propuesto según el tipo de proyecto, es la fase de mayor duración que implica el desempeño de las competencias genéricas y específicas a desarrollar.
- **Evaluación:** es la fase final que aplica un juicio de valor en el contexto laboral-profesión, social e investigativo, ésta se debe realizar a través del reconocimiento de logros y aspectos a mejorar se estará promoviendo el concepto de “evaluación para la mejora continua”, la metacognición, el desarrollo del pensamiento crítico y reflexivo en los estudiantes.

## 10. Evaluación por competencias

Son las técnicas, instrumentos y herramientas sugeridas para constatar los desempeños académicos de las actividades de aprendizaje.

### Instrumentos de Evaluación:

- Resolución de problemas.
- Prácticas.
- Portafolio de evidencias.
- Exámenes orales y escritos.
- Proyecto.

Las evidencias de los aprendizajes que contribuyen al desarrollo de competencias son:  
**De comportamiento (actitud):** Dinámica de grupos, métodos de toma de decisiones, observación en participaciones individuales o grupales en clase, dialogo en forma de interrogatorio.

**De desempeño:** Reportes de investigación sean individuales o grupales, problemas

desarrollados en forma independiente,

**De producto:** AOP aprendizaje orientado a proyectos, ABP aprendizaje basado en problemas, Método de casos, Métodos de creatividad, Métodos de simulación, resolución de problemas, Interactividad con la computadora, Portafolio de evidencias, Rúbricas de evaluación.

**De conocimiento:** Pruebas objetivas de los temas vistos en clase, Método de casos, Análisis de situaciones, Experimentos, Rúbricas de evaluación.

## 11. Fuentes de información

1. Sara Basse y Allen Van Gelder (2002), *Algoritmos Computacionales: Introducción al Análisis y Diseño*. Addison Wesley, Tercera Edición.
2. Thomas H. Cormen, Charles E. Leiserson, Ronald R. Rivest y Clifford Stein (1990), *Introduction to Algorithms*, MIT Press, second edition.
3. Steven S. Skiena (2008), *The Algorithm Design Manual*, Springer Verlag.
4. Ahmed Shamsul Arefin (2006). Art of programming contest. Gyankosh Prokashoni 2006. (Descargado el 7 de octubre de 2018 de [https://www.comp.nus.edu.sg/~stevenha/database/Art\\_of\\_Programming\\_Contest\\_SE\\_for\\_uva.pdf](https://www.comp.nus.edu.sg/~stevenha/database/Art_of_Programming_Contest_SE_for_uva.pdf))
5. Elliot Koffman (2010), *Data Structures: Abstraction and Design Using Java*, Second Edition, Wiley.
6. M. T. Goodrich, R. Tomassia y M.H. Goldwasser (2014), *Data Structures and Algorithms in Java*, 6/e, Wiley.
7. Mark A. Weiss (2010), *Data Structures and Problem Solving Using Java*, fourth edition, E. Pearson.

Textos electrónicos, bases de datos y programas informáticos (fecha de consulta 18 de junio del 2015):

1. [http://www.revista.unam.mx/vol.7/num1/art03/ene\\_art03.pdf](http://www.revista.unam.mx/vol.7/num1/art03/ene_art03.pdf)
2. <http://personales.ya.com/casanchi/libros.htm>
3. Wolfram, [www.wolfram.com](http://www.wolfram.com)