



UANL

UNIVERSIDAD AUTÓNOMA DE NUEVO LEÓN



FIME

FACULTAD DE INGENIERÍA MECÁNICA Y ELÉCTRICA

UNIVERSIDAD AUTÓNOMA DE NUEVO LEÓN

Facultad de Ingeniería Mecánica y Eléctrica

PE Doctorado en Ingeniería de Sistemas

PROGRAMA ANALÍTICO

I. Datos de Identificación de la Unidad de Aprendizaje:

1. **Clave y nombre de la Unidad de Aprendizaje:** PD109 Análisis y diseño de algoritmos

2. **Frecuencia semanal:** horas de trabajo presencial 4

3. **Horas de trabajo extra aula por semana:** 2

4. **Modalidad:** ☒ Escolarizada ☐ No escolarizada ☐ Mixto

5. **Período académico:** ☒ Semestral ☐ Tetramestral ☐ Modular

6. **LGAC:** Sistemas estocásticos y simulación

7. **Ubicación semestral:** 1-8

8. **Área curricular:** formación, libre elección

9. **Créditos:** 4

10. **Requisito:** Ninguno

11. **Fecha de elaboración:** 20/01/2010

12. **Fecha de la última actualización:** 10/06/2021

13. **Responsable(s) del diseño:**

095808 Dr. Fernando López Irarragorri

096633 Dra. Satu Elisa Schaeffer



II. Presentación:

Se analiza la complejidad de dos conceptos diferentes de la computación: *problemas* y *algoritmos*. Un problema es un conjunto (posiblemente infinita) de instancias junto con una pregunta sobre alguna propiedad de las instancias. Un algoritmo es un proceso formal para encontrar la respuesta correcta a la pregunta de un problema para una cierta instancia del problema.

III. Propósito(s):

Aplicar los métodos de **análisis y diseño de algoritmos** para mejorar la eficiencia en de sus implementaciones.

IV. Competencias del perfil de egreso:

14. Competencias del perfil de egreso

P1) Realizar investigación original y resolver problemas en el área de toma de decisiones en ambientes operativos que pueden ser dinámicos o inciertos para lograr una asignación más efectiva de recursos y decidir el curso de acción óptimo para lograr objetivos establecidos.

15. Competencias generales a que se vincula la Unidad de Aprendizaje:

<i>Declaración de la competencia general vinculada a la unidad de aprendizaje</i>	<i>Evidencia</i>
C2) Utiliza los lenguajes lógico, formal, matemático, icónico, verbal y no verbal de acuerdo a su etapa de vida en el área de las ciencias para comprender, interpretar y expresar ideas, sentimientos, teorías y corrientes de pensamiento con un enfoque ecuménico.	Tareas
C3) Maneja las tecnologías de la información de acuerdo a los usos del campo de las ciencias y la comunicación como herramientas para el acceso a la información y su transformación en conocimiento, así como para el aprendizaje y trabajo colaborativo con técnicas de vanguardia que le permitan su participación constructiva en la sociedad.	Tareas
C5) Emplea pensamiento lógico, crítico, creativo y propositivo, siguiendo los modelos de pensamiento científico para analizar fenómenos naturales y sociales que le permitan tomar decisiones pertinentes en su ámbito de influencia con responsabilidad social.	Tareas, proyecto
C10) Practica los valores promovidos por la UANL: verdad, equidad, honestidad, libertad, solidaridad, respeto a la vida y a los demás, respeto a la naturaleza, integridad, ética profesional, justicia y responsabilidad, en su ámbito personal y profesional para contribuir a construir una sociedad sostenible.	Tareas, proyecto
C15) Logra la adaptabilidad que requieren los ambientes sociales y profesionales de incertidumbre de nuestra época para crear mejores condiciones de vida utilizando todos los avances científicos a los cuales ha tenido acceso	Tareas, proyecto



UANL

UNIVERSIDAD AUTÓNOMA DE NUEVO LEÓN



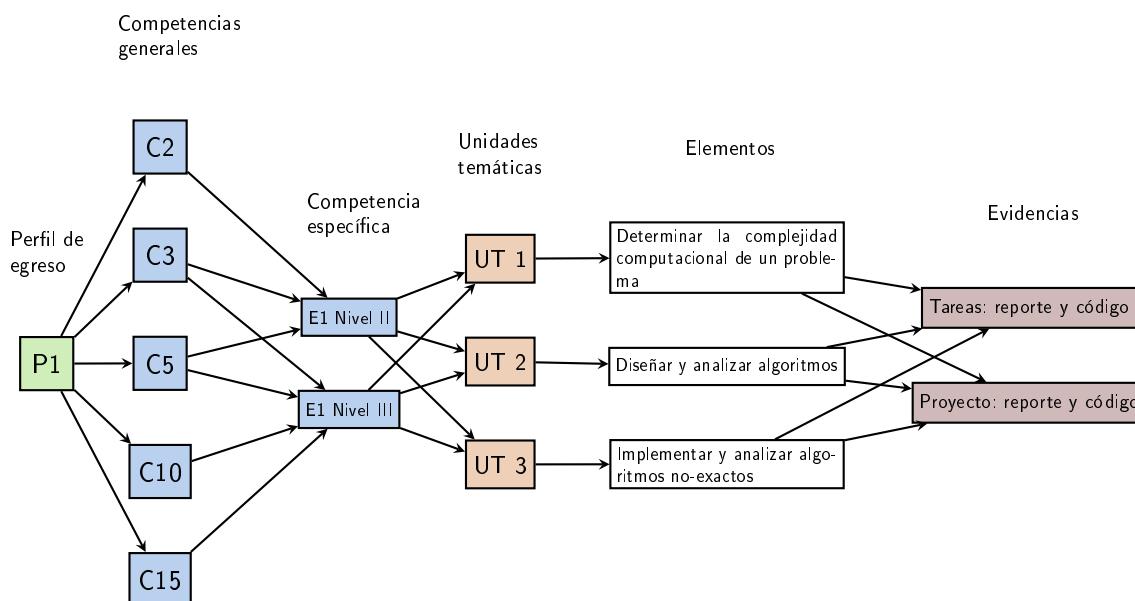
FIME

FACULTAD DE INGENIERÍA MECÁNICA Y ELÉCTRICA

16. Competencias específicas y nivel de dominio a que se vincula la unidad de aprendizaje:

Competencia Específica	Nivel I Inicial	Evidencia	Nivel II Básico	Evidencia	Nivel III Autónomo	Evidencia	Nivel IV Estratégico	Evidencia
E1) Realizar investigación original y resolver problemas en el área de toma de decisiones en ambientes operativos que pueden ser dinámicos o inciertos para lograr una asignación más efectiva de recursos y decidir el curso de acción óptimo para lograr objetivos establecidos.			Resuelve problemas de libro de texto en el área de toma de decisiones con bases científicas.	Tareas.	Encuentra soluciones para la consecución de objetivos establecidos para un problema dado, revisando literatura científica de frontera.	Tareas.		

V. Representación gráfica:



VI. Estructuración en capítulos, etapas o fases de la unidad de aprendizaje:

17. Desarrollo de las fases de la Unidad de Aprendizaje:

Se cubren los principios teóricos del análisis y el diseño de algoritmos computacionales. Desarrollar habilidades en el diseño como en el análisis en casos prácticos concretos basados en algoritmos clásicos. Se necesita contar con un buen entendimiento de varios de los conceptos matemáticos, especialmente de matemáticas discretas y probabilidad, o en el caso contrario, estar preparado a estudiarlos según necesidad. También se necesita conocimiento de programación.

Unidades temáticas

- U1** Fundamentos de la complejidad computacional (8 semanas)
- U2** Elementos básicos de algoritmos (7 semanas)
- U3** Algoritmos no exactos (2 semanas)

Las sesiones son de cuatro horas cada una y son veinte semanas en total. Las últimas tres semanas son para el desarrollo del proyecto integrador que combina elementos de las tres unidades de aprendizaje.

Temario semanal

- Introducción; selección de temas de proyecto
- U1: Problemas y algoritmos (2 semanas)
- U1: Modelos de computación (2 semanas)
- U1: Complejidad computacional de problemas (2 semanas)
- U1: Clases de complejidad (2 semanas)



UANL

UNIVERSIDAD AUTÓNOMA DE NUEVO LEÓN



FIME

FACULTAD DE INGENIERÍA MECÁNICA Y ELÉCTRICA

- U2: Estructuras de datos (2 semanas)
- U2: Análisis de algoritmos (2 semanas)
- U2: Técnicas de diseño de algoritmos (2 semanas)
- U2: Optimización combinatoria (1 semana)
- U3: Algoritmos de aproximación (1 semana).
- U3: Algoritmos aleatorizados (1 semana)
- Presentaciones de proyectos
- Revisión de portafolios de evidencia

Unidad temática 1: Fundamentos de la complejidad computacional

Periodo: 8 semanas

Elementos de competencia:

Evidencias de aprendizaje	Criterios de desempeño	Actividades de aprendizaje	Contenidos	Recursos
Ocho (8) tareas semanales que consisten cada una en un reporte escrito de un aspecto básico de la teoría de la computación.	Calidad de la redacción científica del reporte.	Experimentación con ejemplos; lectura de material de apoyo; modificación de ejemplos; diseño y ejecución de experimentos; análisis y reportaje de resultados obtenidos.	Conceptos básicos diversos de diseño y análisis de algoritmos.	Material en la página web de la unidad y la literatura citada; lenguaje Python o similar; paquete \LaTeX para redacción científica; repositorios de públicos de código fuente.

Unidad temática 2: Elementos básicos de algoritmos

Periodo: 7 semanas

Elementos de competencia:

Evidencias de aprendizaje	Criterios de desempeño	Actividades de aprendizaje	Contenidos	Recursos
Siete (7) tareas semanales consistiendo en un reporte escrito y código de la implementación de un algoritmo y su análisis.	Calidad de la redacción científica del reporte; precisión del algoritmo propuesto; eficiencia de la implementación del algoritmo; cobertura de la experimentación.	Experimentación con ejemplos; lectura de material de apoyo; modificación de ejemplos; diseño y ejecución de experimentos; análisis y reportaje de resultados obtenidos.	Diversos algoritmos fundamentales.	Material en la página web de la unidad y la literatura citada; lenguaje Python o similar; paquete \LaTeX para redacción científica; repositorios de públicos de código fuente.

Unidad temática 3: Algoritmos no exactos

Periodo: 2 semanas

Elementos de competencia:

Evidencias de aprendizaje	Criterios de desempeño	Actividades de aprendizaje	Contenidos	Recursos
Dos (2) tareas consistiendo cada una en un reporte escrito y código de la implementación un algoritmo no exacto y su análisis.	Calidad de la redacción científica del reporte; precisión del algoritmo propuesto; eficiencia de la implementación del algoritmo; cobertura de la experimentación.	Experimentación con ejemplos; lectura de material de apoyo; modificación de ejemplos; diseño y ejecución de experimentos; análisis y reporte de resultados obtenidos.	Métodos diversos de algoritmos no exactos.	Material en la página web de la unidad y la literatura citada; lenguaje Python o similar; paquete \LaTeX para redacción científica; repositorios de públicos de código fuente.



UANL

UNIVERSIDAD AUTÓNOMA DE NUEVO LEÓN



FIME

FACULTAD DE INGENIERÍA MECÁNICA Y ELÉCTRICA

VII. Evaluación integral de procesos y productos:

Las tareas son individuales; se recomienda estudiar juntos y discutir las soluciones, pero no se tolera ningún tipo de plagio en absoluto, ni de otros estudiantes ni de la red ni de libros — toda referencia bibliográfica tiene que ser apropiadamente citada. La entrega se realiza por un repositorio público que debe reflejar todas las fases del trabajo.

No habrá examen. Son 17 tareas (A1–A17) que reportan avances semanales de aplicación de la lectura de la semana para el proyecto del estudiante, otorgando por máximo 5 puntos por tarea:

NP = tarea omitida

5 = excede lo que se esperaba

4 = cumple con lo que se esperaba

3 = débil en alcance y/o calidad

2 = débil en ambos alcance y calidad

1 = sin contribuciones o méritos aunque fue entregada

0 = completamente inadecuado en alcance y calidad

El proyecto final (A18) otorga un máximo de 15 puntos, evaluados en los siguientes rubros

1. Variedad de técnicas de empleadas
2. Cobertura y validez de la experimentación
3. Claridad y relevancia de los resultados
4. Calidad de visualización científica
5. Calidad de redacción científica

con la escala:

3 = cumple con lo que se esperaba

2 = débil en alcance y/o calidad

1 = débil en ambos alcance y calidad

0 = inadecuado en alcance y calidad

Ponderación específica

Actividad																			Total
	A1	A2	A3	A4	A5	A6	A7	A8	A9	A10	A11	A12	A13	A14	A15	A16	A17	A18	
Ponderación	5 %	5 %	5 %	5 %	5 %	5 %	5 %	5 %	5 %	5 %	5 %	5 %	5 %	5 %	5 %	5 %	5 %	15 %	100 %



VIII. Producto integrador de aprendizaje de la unidad:

18. Producto integrador de Aprendizaje:

Portafolio en un repositorio digital público que contiene los reportes escritos y los códigos de la implementación de todas las tareas y el proyecto integrador.

IX. Fuentes de apoyo y consulta:

19. Fuentes de apoyo y consulta

Básicas

- R. SEDGEWICK & P. FLAJOLET: *An Introduction to the Analysis of Algorithms*. Addison Wesley, 512 páginas, 1995. ISBN-13 978-0201400090.
- C.H. PAPADIMITRIOU: *Computational Complexity*. Addison Wesley, 500 páginas, 1993. ISBN-13 978-0201530827.
- D.L. KREHER & Douglas R. STINTON: *Combinatorial Algorithms — Generation, Enumeration, and Search*. CRC Press, 344 páginas, 1998. ISBN-13 978-0849339882.
- M.R. GAREY & D.S. JOHNSON: *Computers and Intractability: A Guide to the Theory of NP-Completeness*. W. H. Freeman, 340 páginas, 1979. ISBN-13: 978-0716710455.

Complementarias

- T.H. CORMEN, C.E. LEISERSON, R.L. Rivest & C. STEIN: *Introduction to Algorithms*. MIT Press, 1184 páginas, segunda edición, 2001. ISBN-13 978-0262032933.
- R. DIESTEL: *Graph Theory*. Graduate Texts in Mathematics, Volume 173. Springer-Verlag, 431 páginas, 2005. ISBN 3-540-26183-4. Tercera edición.
- M. MITZENMACHER y Eli UPFAL: *Probability and Computing: Randomized Algorithms and Probabilistic Analysis*. Cambridge University Press, 368 páginas, 2005. ISBN-13 978-0521835404.
- D.E. KNUTH: *The Art of Computer Programming*. Volúmenes 1–3. Addison Wesley, 896 páginas, segunda edición, 1998. ISBN-13 978-0201485417. Volumen 4: *Generating All Trees—History of Combinatorial Generation*. Addison Wesley, 128 páginas, 2006. ISBN-13 978-0321335708.
- D. JUNGnickel: *Graphs, Networks and Algorithms*. Springer, 611 páginas, segunda edición, 2004. ISBN-13 978-3540219057.
- R.L. GRAHAM, D.E. KNUTH & O. PATASHNIK: *Concrete Mathematics: A Foundation for Computer Science*. Addison Wesley, 672 páginas, segunda edición, 1994. ISBN-13 978-0201558029.
- N. ALON & J.H. SPENCER: *The Probabilistic Method*. Wiley Interseience, 328 páginas, 2000. ISBN-13 978-0471370468.
- E. AARTS & J.K. LENSTRA: *Local Search in Combinatorial Optimization*. Princeton University Press, 536 páginas, 2003. ISBN-13 978-0691115221.
- A.V. AHO, et al: *Compilers — Principles, Techniques & Tools*. Addison Wesley, 1040 páginas, 2006. (Segunda edición.) ISBN-13 978-0321486813.

Artículos científicos especializados.