Nombre Corto	Análisis de algoritmos
Nombre Largo	Análisis de algoritmos
Descripción	La asignatura se centra en el análisis formal de problemas que son solubles con algoritmos, para estudiantes con experiencia en programación. Se desarrollan las habilidades para analizar problemas, describir la complejidad de un algoritmo y aplicar métodos avanzados para resolver problemas de optimización y decisión. El curso utiliza clases magistrales interactivas, talleres guiados y trabajo en equipo para desarrollar un proyecto.
Créditos	2
Condiciones	• Estructuras de datos. • Teoría de la computación.

Objetivos de formación

1. Presentar el lenguaje formal de diseño de problemas algorítmicos.
2. Mostrar los diferentes tipos y clases de problemas algorítmicos.
3. Exponer a los estudiantes a situaciones de formalización de problemas mal condicionados.
4. Presentar las principales estrategias de solución de problemas polinomiales (P).
5. Solucionar algunos problemas no-determinísticos polinomiales (NP-completos) con algoritmos de aproximación bien conocidos en la literatura.
6. Presentar estrategias para verificar y probar implementaciones.

Contenidos temáticos

1. Análisis asintótico.
2. Estrategias básicas de resolución de problemas.
3. Estrategias avanzadas de resolución de problemas.
4. Tractabilidad.

Competencias transversales

2.1 RAZONAMIENTO ANÁLITICO Y SOLUCIÓN DE PROBLEMAS
Escoger la alternativa más adecuada para modelar el problema. (2)
Proponer una solución al problema planteado (3)

2.2 EXPERIMENTACIÓN, INVESTIGACION Y DESCUBRIMIENTO DEL CONOCIMIENTO
Conocer conceptos relacionados al proceso investigativo: métodos, paradigmas, estrategias, hipótesis, experimentación, roles, consideraciones éticas (1)

4.1 CONTEXTO SOCIAL, AMBIENTAL Y EXTERNO

Entender las formas en las que las tecnologías de información afectan elementos del contexto social y ambiental (2)

4.3 CONCEPCIÓN, APLICACIÓN DE LA INGENIERÍA Y GESTIÓN DE SISTEMAS

4.4 DISEÑO
Entender los procedimientos necesarios para evaluar la calidad de un diseño disciplinar y multidisciplinar (2)

Resultados de aprendizaje esperados (RAE)

Al finalizar el curso el estudiante estará en capacidad de:

1. Analizar diferentes alternativas de solución para problemas algorítmicos (disciplinar 1-4, CDIO 2.1).
2. Proponer soluciones computacionalmente adecuadas para problemas algorítmicos (Disciplinar 1-4, CDIO 2.1).
3. Describir cuándo un problema es soluble exacta o aproximadamente por un computador (Disciplinar 1-4, CDIO 2.1).
4. Relacionar el análisis formal de algoritmos con ambientes formales de ingeniería de software (Disciplinar 1-4, CDIO 4.3).
5. Explicar el impacto del desarrollo de algoritmos en el uso de recursos (tiempo y almacenamiento) (Disciplinar 1-4) (CDIO 2.2 y 4.1).
6. Entender los procedimientos necesarios para evaluar la calidad de un diseño algorítmico aplicado a un problema real (Disciplinar 1-4) (CDIO 4.4)

Rúbricas

RAE 1: Analizar diferentes alternativas de solución para problemas algorítmicos						
Indicador de desempeño	*Insatisfactorio [0-2)*	*Bajo el estándar [2-3)*	*Competente [3-3.75)*	*Ejemplar [3.75-4.25)*	*Ejemplar de alto nivel [4.25-5]*	*Herramienta y valor porcentual dentro de ella*
Conoce los pasos para el análisis de complejidad por inspección	No conoce los pasos necesarios para realizar un análisis de complejidad por inspección	Conoce unos pocos pasos para realizar un análisis de complejidad por inspección	Conoce la mayoría de los pasos necesarios para realizar un análisis de complejidad por inspección, omitiendo pocos pasos y algunos detalles	Conoce en general los pasos necesarios para realizar un análisis de complejidad por inspección, omitiendo sólo algunos detalles	Conoce todos los pasos necesarios para realizar un análisis de complejidad por inspección	Parcial 1 (20%)
Usa el teorema maestro para analizar la complejidad de	No conoce el teorema maestro	Identifica las restricciones básicas del teorema maestro, pero	Conoce las restricciones básicas del teorema maestro, lo	Usa el teorema maestro, lo relaciona con un algoritmo dado, lo usa	Usa todos los pasos para analizar un algoritmo a la	Parcial 1 (20%)

algoritmos recurrentes		no es capaz de relacionarlo con un algoritmo	relaciona con un algoritmo dado, pero no es capaz de usarlo para calcular la complejidad de un algoritmo.	para calcular la complejidad de un algoritmo omitiendo algunos detalles	luz del teorema maestro.	
Identifica los diferentes tipos de problemas computacionales	No identifica los tipos de problemas.	Identifica muy superficialmente los tipos de problema o los confunde con las clases de problemas	Identifica los tipos de problemas, los relaciona con las clases de problemas, pero no los relaciona con las estrategias de resolución de problemas.	Identifica los tipos de problemas, los relaciona con las clases de problemas y con las estrategias de resolución de problemas omitiendo algunos detalles.	Identifica los diferentes tipos de problemas y su relación con las clases de problemas.	Parcial 3 (50%)

RAE 2: Proponer soluciones computacionalmente adecuadas para problemas algorítmicos						
Indicador de desempeño	*Insatisfactorio [0-2)*	*Bajo el estándar [2-3)*	*Competente [3-3.75)*	*Ejemplar [3.75-4.25)*	*Ejemplar de alto nivel [4.25-5]*	*Herramienta y valor porcentual dentro de ella*
Escribe algoritmos que usan la estrategia de dividir-y-vencer	No identifica problemas solucionables con algoritmos dividir-y-vencer	Identifica problemas solucionables por dividir-y-vencer, pero no es capaz de modelar los pasos del algoritmo	Identifica problemas solucionables por dividir-y-vencer, es capaz de plasmar los pasos del algoritmo con algunos errores de implementación	Identifica problemas solucionables por dividir-y-vencer, es capaz de plasmar los pasos del algoritmo omitiendo algunos detalles	Identifica problemas solucionables por dividir-y-vencer, es capaz de plasmar los pasos del algoritmo y los implementa correctamente en un lenguaje de programación	Parcial 1 (100%)
Escribe algoritmos que usan la estrategia de programación dinámica	No identifica problemas solucionables con algoritmos de programación dinámica y/o voraces	Identifica problemas solucionables por programación dinámica, pero no expresa el diseño recurrente del algoritmo	Identifica problemas solucionables por programación dinámica, es capaz de expresar el diseño recurrente del algoritmo, pero no escribe el algoritmo iterativo “abajo-arriba”	Identifica problemas solucionables por programación dinámica, es capaz de expresar el diseño recurrente del algoritmo, escribe el algoritmo iterativo “abajo-arriba”, pero no escribe el backtracking para calcular soluciones finales.	Identifica problemas solucionables por programación dinámica, es capaz de expresar el diseño recurrente del algoritmo, de escribir el algoritmo iterativo “abajo-arriba” y es capaz de escribir el backtracking para calcular soluciones finales.	Parcial 2 (100%)

Formula aproximaciones a la solución de problemas con heurísticas	No identifica problemas super-polinomiales (pertenecientes a la clase NP)	Identifica problemas superpolinomiales, pero no formula un algoritmo de fuerza bruta para solucionarlo	Identifica problemas superpolinomiales, formula un algoritmo de fuerza bruta para solucionarlo, pero no formula una heurística de solución	Identifica problemas superpolinomiales, formula un algoritmo de fuerza bruta para solucionarlo, y formula una heurística de solución, pero no discute las ventajas y desventajas de dicho algoritmo	Identifica problemas superpolinomiales, formula un algoritmo de fuerza bruta para solucionarlo, y formula una heurística de solución.	Proyecto final (100%)

RAE 3: Describir cuándo un problema es soluble exacta o aproximadamente por un computador						
Indicador de desempeño	*Insatisfactorio [0-2)*	*Bajo el estándar [2-3)*	*Competente [3-3.75)*	*Ejemplar [3.75-4.25)*	*Ejemplar de alto nivel [4.25-5]*	*Herramienta y valor porcentual dentro de ella*
Identifica las clases de problemas	No identifica las clases de problemas.	Identifica problemas polinomiales y superpolinomiales, pero no identifica la estrategia de fuerza bruta del problema	Identifica problemas polinomiales y superpolinomiales, identifica la estrategia de fuerza bruta pero no identifica el algoritmo que verifica que una solución es correcta.	Identifica problemas polinomiales y superpolinomiales, identifica la estrategia de fuerza bruta e identifica el algoritmo que verifica que una solución es correcta con algunas omisiones.	Identifica problemas polinomiales y superpolinomiales, identifica la estrategia de fuerza bruta e identifica el algoritmo que verifica que una solución es correcta.	Parcial 3 (50%)
Escribe formalmente algoritmos de fuerza bruta	No identifica estrategias de fuerza bruta.	Identifica estrategias de fuerza bruta, pero no escribe el algoritmo que verifica la correctitud de una solución.	Identifica estrategias de fuerza bruta, escribe el algoritmo que verifica la correctitud de una solución, pero no identifica oportunidades de mejora del algoritmo.	Identifica estrategias de fuerza bruta, escribe el algoritmo que verifica la correctitud de una solución, identifica oportunidades de mejora del algoritmo, pero no da argumentos convincentes de esas mejoras.	Identifica estrategias de fuerza bruta, escribe el algoritmo que verifica la correctitud de una solución,identifica oportunidades de mejora del algoritmo y da argumentos convincentes de esas mejoras.	Proyecto final (100%)
Describe el grado de aproximación de una heurística	No entiende el uso de una heurística.	Entiende la utilidad de una heurística, pero no la implementa	Entiende el grado de aproximación de una heurística, pero no dar argumentos a favor y en contra de ese grado.	Entiende el grado de aproximación de una heurística y da argumentos a favor y en contra de ese grado, aunque con omisiones y	Entiende el grado de aproximación de una heurística y es capaz de dar argumentos a favor y en contra de ese grado.	Proyecto final (100%)

				huecos de razonamiento.		

RAE 4: Relacionar el análisis formal de algoritmos con ambientes formales de ingeniería de software						
Indicador de desempeño	*Insatisfactorio [0-2)*	*Bajo el estándar [2-3)*	*Competente [3-3.75)*	*Ejemplar [3.75-4.25)*	*Ejemplar de alto nivel [4.25-5]*	*Herramienta y valor porcentual dentro de ella*
Describe las clases y el tipo de un problema.	No identifica las clases de problemas ni su relación con los tipos.	Identifica problemas polinomiales y superpolinomiales, pero no identifica la estrategia de fuerza bruta del problema. Además, no aprovecha el tipo de problema para identificar posibles reducciones.	Identifica problemas polinomiales y superpolinomiales, es capaz de identificar la estrategia de fuerza bruta pero no identifica el algoritmo que verifica que una solución es correcta. Aprovecha el tipo de problema para identificar posibles soluciones.	Identifica problemas polinomiales y superpolinomiales, es capaz de identificar la estrategia de fuerza bruta e identifica el algoritmo que verifica que una solución es correcta con algunas omisiones. Aprovecha el tipo de problema para identificar posibles soluciones.	Identifica problemas polinomiales y superpolinomiales, es capaz de identificar la estrategia de fuerza bruta e identifica el algoritmo que verifica que una solución es correcta. Aprovecha el tipo de problema para identificar posibles soluciones.	Parcial 3 (50%)
Usa las herramientas de análisis asintótico en conjunto con el análisis de contexto del problema para escoger la mejor solución algorítmica.	No conoce los pasos necesarios para realizar un análisis de complejidad por inspección. No conoce el teorema maestro	Conoce unos pocos pasos para realizar un análisis de complejidad por inspección. Identifica las restricciones básicas del teorema maestro, pero no es capaz de relacionarlo con un algoritmo	Conoce la mayoría de los pasos necesarios para realizar un análisis de complejidad por inspección, omitiendo pocos pasos y algunos detalles. Conoce las restricciones básicas del teorema maestro, lo relaciona con un algoritmo dado, pero no es capaz de usarlo para calcular la complejidad de un algoritmo	Conoce en general los pasos necesarios para realizar un análisis de complejidad por inspección, omitiendo sólo algunos detalles. Conoce las restricciones básicas del teorema maestro, lo relaciona con un algoritmo dado, lo usa para calcular la complejidad de un algoritmo omitiendo algunos detalles	Conoce todos los pasos necesarios para realizar un análisis de complejidad por inspección. Conoce todos los pasos para analizar un algoritmo a la luz del teorema maestro.	Proyecto final (100%)
Explica la utilidad de los algoritmos de fuerza bruta	No identifica estrategias de fuerza bruta.	Explica las estrategias de fuerza bruta, pero no explica	Explica las estrategias de fuerza bruta, escribe el	Explica las estrategias de fuerza bruta, escribe el	Explica las estrategias de fuerza bruta, escribe el	Parcial 3 (100%)

		el algoritmo que verifica la correctitud de una solución.	algoritmo que verifica la correctitud de una solución, pero no identifica oportunidades de mejora del algoritmo.	algoritmo que verifica la correctitud de una solución, no identifica oportunidades de mejora del algoritmo, pero no da argumentos convincentes de esas mejoras.	algoritmo que verifica la correctitud de una solución, no identifica oportunidades de mejora del algoritmo y da argumentos convincentes de esas mejoras.	

RAE 5: Explicar el impacto del desarrollo de algoritmos en el uso de recursos (tiempo y almacenamiento)						
Indicador de desempeño	*Insatisfactorio [0-2)*	*Bajo el estándar [2-3)*	*Competente [3-3.75)*	*Ejemplar [3.75-4.25)*	*Ejemplar de alto nivel [4.25-5]*	*Herramienta y valor porcentual dentro de ella*
Identifica los diferentes tipos de problemas computacionales	No identifica los tipos de problemas.	Identifica superficialmente los tipos de problema o los confunde con las clases de problemas	Identifica los tipos de problemas, los relaciona con las clases de problemas, pero no los relaciona con las estrategias de resolución de problemas.	Identifica los tipos de problemas, los relaciona con las clases de problemas y con las estrategias de resolución de problemas omitiendo algunos detalles.	Identifica los diferentes tipos de problemas y su relación con las clases de problemas. Además, identifica las estrategias de resolución de problemas	Parcial 3 (100%)

RAE 6: Entender los procedimientos necesarios para evaluar la calidad de un diseño algorítmico aplicado a un problema real						
Indicador de desempeño	*Insatisfactorio [0-2)*	*Bajo el estándar [2-3)*	*Competente [3-3.75)*	*Ejemplar [3.75-4.25)*	*Ejemplar de alto nivel [4.25-5]*	*Herramienta y valor porcentual dentro de ella*
Identifica el proceso formal de escritura de algoritmos	No identifica el proceso formal de escritura.	Identifica únicamente el contrato del algoritmo (entradas y salidas).	Identifica el contrato del algoritmo (entradas y salidas) y el seudo-algoritmo.	Identifica el diseño preliminar del problema, el contrato del algoritmo (entradas y salidas) y el seudo-algoritmo.	Identifica el diseño preliminar del problema, el contrato del algoritmo (entradas y salidas) y el seudo-algoritmo. Además, identifica la invariante del algoritmo.	Proyecto final (100%)
Usa el proceso de pruebas unitarias	No prueba si los resultados obtenidos por el algoritmo son los esperados	Prueba sólo algunos pocos resultados obtenidos con el algoritmo sin determinar si	Revisa un conjunto mínimo de resultados obtenidos con el algoritmo,	Comprueba la mayoría de los resultados obtenidos con el algoritmo para verificar	Comprueba si los resultados obtenidos por el algoritmo corresponden con lo esperado	Proyecto final (100%)

		son los esperados	pero no determina si son los esperados.	que son los esperados		

Estrategias Pedagógicas

Durante el curso se utilizarán 3 estrategias. La primera de ellas será el aprendizaje directivo mediado mediante clases magistrales. La segunda el aprendizaje basado en problemas mediante talleres guiados en clase, donde se expone a los estudiantes a problemas reales solubles con algoritmos. La tercera será el aprendizaje por proyectos, donde se propone la elaboración de una solución que responda a un problema algorítmico real.

Evaluación

Las estrategias de evaluación están centradas en la valoración de los resultados de aprendizaje esperado de la asignatura; las cuales pueden ser formativas que suscitan la comprensión y construcción de conocimiento, y sumativas que incluyen porcentajes de evaluación con el fin de corroborar el logro de los aprendizajes y el desarrollo de las competencias en los estudiantes. Las estrategias de evaluación de la asignatura son:

- Primer parcial 20%
- Segundo parcial 20%
- Tercer parcial 20%
- Proyecto 20%
- Quices y talleres guiados … 20%

Recursos Bibliográficos

7. Cormen, T. H.; Leiserson, C. E.; Rivest, R. L. & Stein, C. (2009), *Introduction to Algorithms*, Third Edition, The MIT Press.
8. Bohorquez, J.; Cardoso R. (1992), *Análisis de algoritmos*, Universidad de los Andes.
9. Aho, A. V.; Hopcroft, J. E.; Ullman, J. D. (1973), *The design and analysis of computer algorithms*, Addison Wesley.
10. Baase, S. (1978), *Computer algorithms: Introduction to design and analysis*, Addison Wesley.
11. Brassard, G.; Bratley, P. (1996), *Fundamentos de Algoritmia*, Prentice Hall.
12. Horowitz, E.; Sahni, S. (1978), *Fundamentals of Computer Algorithms*, Computer Science Press.
13. Kaldewaij, A. (1990), *Programming: the derivation of algorithms*, Prentice-Hall.
14. Manber, U. (1989), Introduction to Algorithms: A Creative Approach, Addison-Wesley.
15. McConnell, J. J. (2001), Analysis of algorithms: an active learning approach. Second Edition, Sudbury, Massachusetts: Jones and Bartlett Publishers.
16. Parberry, I. (1995), Problems on algorithms, Prentice-Hall.

Tabla de Contenidos

Comentarios

Instrumentos de evaluación: los tres parciales y la entrega final del proyecto.