



Información general

Asignatura	ALGORITMOS Y ESTRUCTURAS DE DATOS				
Código	11310006				
Tipo de asignatura	Obligatoria	X	Electiva		
Tipo de saber	Obligatoria básica o de fundamentación	X	Obligatoria profesional	Obligatoria complementaria	
Número de créditos	3 (tres)				
Tipo de crédito	2A + 1C				
Horas de trabajo con acompañamiento directo del profesor		Horas de trabajo independiente del estudiante		Total de horas	
Prerrequisitos	Programación de computadores (11310002)				
Correquisitos	Ninguno				

Horario		Martes: 7-10am. Jueves: 7-9am
Salón		Sede Claustro Sala Lovelace (5to piso – Torre 2)
Profesor	Nombre	Julián J. Rincón
	Correo electrónico	josej.jimenez@urosario.edu.co
	Lugar y horario de atención	Edificio Cabal. Oficina 408 Jueves: 9-11am
	Página web	
Profesor auxiliar o monitor	Nombre	Edgar Santiago Jaimes Oscar Velasco Chiquillo
	Correo electrónico	edgar.jaimes@urosario.edu.co oscar.velasco@urosario.edu.co
	Lugar y horario de atención	
	Página web	



UNIVERSIDAD DEL ROSARIO

Resumen y propósitos de formación del curso

Este curso se dedica a formar en el alumno las habilidades apropiadas para diseñar sus propios algoritmos y con ellos sus propias estructuras de datos apropiadas para resolver problemas. Se consideran estructuras generales como árboles, listas enlazadas, arreglos, pilas, colas, entre otros, así como algoritmos de recursión, ordenamiento, selección y búsqueda. No es un curso de programación, es un curso de análisis y diseño de algoritmos con propósitos específicos, que se implementan sobre una plataforma (compilador/interprete). Hace parte muy importante del curso la implementación, documentación, sustentación, análisis lógico metódico y demostraciones (pruebas) sobre casos de uso y teoremas. Se introducen diversos tipos de algoritmos y estructuras de datos para dar herramientas al alumno. Se fomenta el trabajo tanto independiente como en grupo para implementar soluciones algorítmicas a problemas prácticos.

Temas

1. Estructuras lineales: vectores, listas, pilas, colas.
2. Estructuras asociativas: conjuntos y mapas.
3. Algoritmos de búsqueda, ordenamiento y selección.
4. Análisis de Algoritmos. Complejidad computacional.
5. Recursión.
6. Punteros y manejo dinámico de memoria en C++.
7. Arreglos. Listas enlazadas.
8. Árboles de búsqueda binaria.
9. Tablas hash.
10. Montículos binarios (*binary heaps*).
11. Merge sort, heapsort y quicksort.

Resultados de aprendizaje esperados (RAE)

1. Identificar los patrones y partes de un problema.
2. Implementar algoritmos de conteo, búsqueda, selección y ordenamiento.
3. Utilizar estructuras de datos básicas y avanzadas.



UNIVERSIDAD DEL ROSARIO

4. Identificar los elementos lógicos y procedimentales que requiere un algoritmo con una función y propósitos específicos dentro de un sistema de compilación apropiado.
5. Explicar los conceptos y técnicas aprendidos en el curso.
6. Identificar las características clave que afecta la eficiencia de un algoritmo
7. Concebir nuevos algoritmos a partir de las herramientas aprendidas en clase.

Actividades de aprendizaje

1. Clases magistrales donde se ilustrarán los conceptos de programación y algorítmica.
2. Talleres donde implementarán los conceptos en programas de computación.
3. Quices y tareas donde trabajarán independientemente en los contenidos vistos y su implementación.
4. Proyecto final donde deberán aplicar las herramientas aprendidas durante el semestre, en conexión con otras clases del programa de primer y segundo semestre.

Actividades de evaluación

Tema	Actividad de evaluación	Porcentaje
	Evaluación escrita individual – 1er parcial	20
	Evaluación escrita individual – 2do parcial	20
	Evaluación escrita individual – 3er parcial	20
Proyecto Final	Proyecto Final	20
Trabajo individual y en clase	Tareas y Quices	20

Programación de actividades por sesión

Sesión Fecha	Tema	Evaluación	Recursos
1	Tipos de datos abstractos Estructuras de datos		[2, sec. 3.1] [3, par. 3 intro.]
2	Estructuras lineales y adaptadores: vectores y listas	Quiz 1	[1, cap. 5] [2, cap. 3] [3, cap. 10]



UNIVERSIDAD DEL ROSARIO

3	Estructuras lineales y adaptadores: pilas y colas		
4	Estructuras asociativas e iteradores: conjuntos y mapas		[1, cap. 5] [3, cap. 10]
Feb. 1	Monitoría GNU/Linux		
5	Algoritmos de búsqueda básicos	Quiz 2	[2, sec. 2.4.4] [1, sec. 7.5]
6	Algoritmos de ordenamiento y selección básicos		[1, secs. 10.1, 10.2] [2, sec. 7.2] [2, sec. 10.2.3]
7			
8	Análisis de Algoritmos: complejidad Peor, mejor caso y caso promedio	Quiz 3	[1, cap. 10] [2, cap. 2] [3, caps. 2, 3]
9			
Feb. 15	Monitoría Parcial 1		
10 Feb. 20	PRIMER PARCIAL		
11	Recursión	Quiz 4	[1, caps. 7, 8] [3, cap. 4]
12			
Feb. 29	Monitoría Análisis de algoritmos		
13	Punteros y arreglos (repaso) Manejo dinámico de la memoria en C++		[1, cap. 11] [2, cap. 1] [1, secs. 12.1, 12.3, 12.5]
14	Clases en C++: C++14		[1, caps. 2, 6] [2, sec. 1.4] cplusplus.com cppreference.com
15	Implementación de pila: Clase CharStack	Tarea 1	[1, cap. 12] [2, cap. 3]
16	Listas enlazadas: Clase Cell		[1, sec. 13.5] [2, cap. 3]
17	Templates Pila con listas		[2, caps. 1, 3] [1, cap. 14]
18	Implementación de cola: listas y arreglos dinámicos	Tarea 2	[1, cap. 14] [2, cap. 3]
19	Árboles binarios de búsqueda		[1, cap. 16] [2, cap. 4] [3, cap. 12]
Mar. 21	Monitoría Parcial 2		



UNIVERSIDAD DEL ROSARIO

20 Mar. 26	SEGUNDO PARCIAL		
21	Árboles binarios de búsqueda	Tarea 2	[1, cap. 16]
22			[2, cap. 4] [3, cap. 12]
23	Tablas hash Implementación de HashMap	Tarea 3	[1, cap. 15]
24			[2, cap. 5]
25			[3, cap. 11]
26	Binary heaps Priority queues		[2, secs. 6, 7.5, 7.7]
27			[3, caps. 6, 7]
28	Algoritmos de ordenamiento avanzados: heapsort, merge sort y quicksort	Tarea 4	[2, secs. 6, 7.5, 7.7]
29			[3, caps. 6, 7]
May. 2	Monitoría Parcial 3		
30 May. 7	TERCER PARCIAL		
31	Monitoría Proyecto Final		
32	Monitoría Proyecto Final		

Monitorías (obligatorias) programadas

Sesión	Tema	Fecha	Recursos
1	GNU/Linux: Línea de comandos	Feb. 1	Notas de clase
2	Análisis de algoritmos	Feb. 15	
3	Referente al Parcial 1	Feb. 29	
4	Referente al Parcial 2	Mar. 21	
5	Referente al Parcial 3	May. 2	

Bibliografía

- [1] Roberts, Eric. (2013). *Programming Abstractions in C++*. Pearson.
- [2] Weiss, Mark A. (2013). *Data Structures and Algorithm Analysis in C++*. Prentice Hall.
- [3] Cormen, Thomas H., Leiserson, Charles E., Rivest, Ronald L. & Stein, Clifford. (2009). *Introduction to Algorithms*. Tercera edición. MIT Press.



UNIVERSIDAD DEL ROSARIO

Bibliografía complementaria

- [4] Dasgupta, S., Papadimitriou, C. H. & Vazirani, U. V. (2006). *Algorithms*. McGraw-Hill.
[5] Kleinberg, Jon & Tardos, Éva. (2005). *Algorithm Design*. Pearson.
[6] Deitel, Paul & Deitel, Harvey. (2016). *C++ How to Program*. Décima edición. Pearson.

Acuerdos de funcionamiento (Reglas de juego)

No está permitido comer o usar dispositivos móviles dentro de clase.

Todas las sesiones y actividades del curso son de carácter teórico-práctico; es decir, incluyen clases magistrales, discusiones, ejercicios o talleres.

Para todas las sesiones se espera que el estudiante realice un trabajo independiente previo que permite un avance continuo en los temas y facilite el entendimiento de los mismos.

No se realizará aproximación de notas al final del semestre. Las notas solo serán cambiadas con base en reclamos OPORTUNOS dentro de los límites de tiempo determinados por el Reglamento Académico. Si por motivos de fuerza mayor el estudiante falta a algún parcial o quiz, deberá seguir el procedimiento regular determinado por el Reglamento Académico para presentar supletorios. No habrá acuerdos informales al respecto. No se eximirá a ningún estudiante de ningún examen.

ASISTENCIA AL CURSO

Con el propósito de afianzar el modelo pedagógico contemplado en el Proyecto Educativo Institucional y promover un rendimiento académico óptimo, es necesario asegurar un espacio de interacción entre estudiantes y profesores que facilite la reflexión y el debate académico en torno al conocimiento. En este sentido, se valora la participación en las actividades académicas y esta se considera como un deber y un derecho del estudiante. (Artículo 48 Reglamento Académico). **De no asistir a más del 80% de las clases el 15% se pierde con 0.0.**

Si el estudiante se presenta 20 minutos luego de dar inicio a alguna evaluación parcial o final, no podrá presentarla y deberá solicitar supletorio siguiendo la reglamentación institucional.

PROCESOS DISCIPLINARIOS-FRAUDE EN EVALUACIONES



UNIVERSIDAD DEL ROSARIO

Teniendo en cuenta el reglamento formativo-preventivo y disciplinario de la Universidad del Rosario, y la certeza de que las acciones fraudulentas van en contra de los procesos de enseñanza y aprendizaje, cualquier acto corrupto vinculado a esta asignatura será notificado a la secretaría académica correspondiente de manera que se inicie el debido proceso disciplinario. Se recomienda a los estudiantes leer dicho reglamento para conocer las razones, procedimientos y consecuencias que este tipo de acciones pueden ocasionar, así como sus derechos y deberes asociados a este tipo de procedimientos.

La asignatura no tiene ningún tipo de bono.

Protocolo para las actividades con evaluación

Los siguientes lineamientos serán seguidos de forma estricta y sin excepción en actividades como quices, tareas y evaluaciones parciales.

1. Solamente será posible tener acceso a e-aulas.urosario.edu.co y a los sitios web correspondientes a la documentación de C++ dispuestos por el profesor.
2. Maletas, morrales, bolsos, etc. deben estar ubicados al frente del salón.
3. Celulares y otros dispositivos electrónicos deben estar apagados y ser guardados dentro de las maletas antes de ser ubicadas en su respectiva posición.
4. El estudiante no debe intentar ocultar ningún código que no sea propio en la solución a la actividad.
5. El estudiante solo podrá disponer de hojas en blanco como borrador de apuntes (opcional).
6. El estudiante puede tener una hoja manuscrita de resumen (opcional). Esta hoja debe estar marcada con nombre completo.
7. Los grupos pueden consultar entre sí *las ideas básicas*; sin embargo, la solución y detalles del ejercicio debe realizarse *individualmente*. Cualquier tipo de fraude o plagio es causa de anulación directa de la evaluación y correspondiente proceso disciplinario.
8. El grupo de trabajo debe indicar en su entrega de la solución a la actividad cualquier asistencia que haya recibido.
9. El grupo no debe consultar ninguna solución de la solución a la actividad que no sea la suya.
10. El grupo no debe intentar ocultar ningún código que no sea propio en la solución a la actividad.
11. E-aulas se cerrará a la hora en punto acordada para el final de la evaluación. La solución de la actividad debe ser subida antes de esta hora. El material entregado a través de e-aulas será calificado tal como está. Si ningún tipo de material es entregado



UNIVERSIDAD DEL ROSARIO

por este medio, la nota de la evaluación será 0.0.

Se aconseja subir a e-aulas versiones parciales de la solución a la actividad.

12. Todas las evaluaciones serán realizadas en el sistema operativo GNU/Linux.
13. Todas las entregas están sujetas a herramientas automatizadas de detección de plagio en códigos.
14. La evaluación debe presentarse exclusivamente en uno de los computadores ubicados en el salón de clase y a la hora acordada. Presentar la evaluación desde otro dispositivo o en otro horario diferente al estipulado es causa de anulación.

No habrá excepciones a estas reglas.