



Sede

Andina

Escuela de Docencia

Escuela de Producción, Tecnología y Medio Ambiente

Carrera

Ingeniería Electrónica – ingeniería en Telecomunicaciones

PROGRAM A ANALÍTICO DE	Programación y Algoritmos	Código SIU-Guaraní
		B5618

Correlativas según plan de estudios	Para Cursar		Para Aprobar
	Cursada Aprobada	Materia Aprobada	Materia Aprobada
	Informática		Informática
Ciclo Lectivo		Régimen de cursada	
		Semestral	
Carga horaria Semanal	4	Carga horaria total	64
Horas Teóricas Totales		Horas Prácticas Totales	
Horas de estudio extra clase recomendada			
Día/s y horario/s de cursado	Comisión 1: Martes y jueves de 14 a 16hs Laboratorio 2 de Informática Comisión 2: Martes y jueves de 8:30 a 10:30hs Laboratorio 2 de Informática		
Día/s y horario/s de Tutorías/Consultas	A definir con los docentes		

Profesor/a a cargo	Mónica Denham
Equipo de docencia	Mónica Denham, Gustavo Ortiz

Fundamentación

La asignatura tiene como propósito brindar a los alumnos herramientas para la creación de soluciones algorítmicas correctas para distintos problemas. Esto se basa en aprender las estructuras de datos básicas y su utilización en problemas típicos. Además se plantean los fundamentos para realizar un correcto análisis de las soluciones implementadas. Estas herramientas en conjunto proveen al alumno las bases para generar soluciones correctas y eficientes. Además se brindan los conocimientos básicos para la programación de dichas estructuras en el paradigma de objetos.

Propósitos de la asignatura

Presentar al alumno conceptos avanzados de la programación secuencial: análisis de eficiencia, complejidad de los algoritmos. Presentar de forma teórica el concepto de TAD y TADs más importantes. Implementación de dichos TADs. Relacionar estos conceptos con la programación orientada a objetos. Presentar herramientas básicas para el diseño y programación orientada a objetos.

Contenidos Mínimos según plan de estudios

Análisis de algoritmos. Estructuras de datos. Tipos de datos, abstracciones. Colas, listas, hash, árboles, colas priorizadas y conjuntos. Algoritmos de orden. Implementación usando objetos, variables, punteros y referencias. Objetos y clases. Herencia y Polimorfismo. Templates.

Propuesta Metodológica

La asignatura consistirá de clases teóricas y clases prácticas. En las clases teóricas se desarrollarán los temas del programa de la asignatura, incluyendo múltiples ejemplos que faciliten la asimilación de los contenidos conceptuales. Mediante ejemplos prácticos se relacionarán cada uno de los temas vistos. Se fomentará la interacción del alumno con el objetivo de que logre una actitud activa para, a través de su propio razonamiento, logre crear soluciones creativas a problemas planteados durante las clases.

Clases prácticas: durante el curso se propondrán distintas prácticas para favorecer la asimilación de los conceptos vistos en la materia. Al inicio de cada práctica el profesor hará una breve explicación de práctica donde se propondrán ejemplos similares a los ejercicios propuestos en la práctica. Se utilizará el blog de la materia Programación y Algoritmos (<https://sites.google.com/site/programacionunrn/>), para publicar novedades, trabajos prácticos, apuntes y todo el material necesario para el correcto desarrollo de la materia.

Asistencia: para alcanzar la condición de regularidad es necesario que el alumno tenga una asistencia del 75% de las clases (teóricas y prácticas). El alumno puede registrar un porcentaje inferior de presencias en clases, si las justifica debidamente ante los docentes de la asignatura y el coordinador de la carrera.

Regularización: el alumno será regularizado en la materia si alcanza el requisito de asistencia y aprueba los exámenes parciales y/o trabajos evaluatorios.

Cronograma de Actividades Teóricas, Prácticas, Salidas de Campo, etc.

Al ser una materia práctica, las clases son teórico-prácticas.

No hay salidas de campo.

¿Requiere extensión áulica? - modalidad virtual- no

Ajustes para estudiantes con discapacidad

Unidad 1 Introducción	Fecha Probable de Inicio y Finalización Desde el al
<p>Contenidos Datos predefinidos: caracteres, enteros (long, double), flotantes, strings, registros. punteros y referencias. datos definidos por el usuario: vectores, matrices, arreglos n-dimensionales. Algoritmos de orden iterativos: burbuja, inserción, selección, shellsort. Algoritmo de orden recursivo: Quick sort.</p> <p>Actividades Prácticas de la Unidad Trabajo práctico 0: Repaso estructuras de control, iteración, etc. Trabajo Práctico 1: Tipos de datos predefinidos, tipos de datos definidos por el usuario, memoria estática, algoritmos de ordenación, algoritmos de búsqueda.</p> <p>Bibliografía “Introducción a la programación en C”, Marco A. Peña Basurto, José M. Cela Espín. Ediciones UPS 2000. “Programación en C, C++, Java y UML”, Luis Joyanes Aguilar, Ignacio Zahonero Martínez. Mc Graw Hill.</p> <p>Bibliografía Complementaria “Introduction to algorithms”, Cormen. Leiserson. MIT Press 2001.</p>	

Unidad 2 Memoria dinámica	Fecha Probable de Inicio y Finalización Desde el al
<p>Contenidos Memoria dinámica. Gestión de memoria dinámica. Stack y Heap. Funciones malloc(), calloc(), realloc(), free(). Asignación dinámica de arreglos. Asignación dinámica de estructuras.</p>	

Asignación de memoria para arreglos, asignación de memoria para un arreglo de estructuras, arreglos dinámicos. Reglas de funcionamiento de la asignación de memoria. Punteros a funciones.

Actividades Prácticas de la Unidad

Trabajo práctico 2: Punteros. Memoria dinámica. Puntero a función.

Bibliografía

“Introducción a la programación en C”, Marco A. Peña Basurto, José M. Cela Espín. Ediciones UPS 2000.

“Programación en C, C++, Java y UML”, Luis Joyanes Aguilar, Ignacio Zahonero Martínez. Mc Graw Hill.

Bibliografía Complementaria

Haga clic aquí para escribir texto.

Unidad 3 Tipos abstractos de datos	Fecha Probable de Inicio y Finalización Desde el al
<p>Contenidos Tipos abstractos de datos: abstracción, encapsulamiento. TADs: listas, listas doblemente enlazadas, listas circulares, pilas, colas, colas priorizadas, conjuntos, árboles generales, árboles binarios, árboles binarios de búsqueda.</p> <p>Actividades Prácticas de la Unidad</p> <p>Trabajo práctico 3: listas (listas simples, listas doblemente enlazadas, listas circulares). Trabajo práctico 4: pilas, colas y conjuntos. Trabajo práctico 5: árboles genéricos, árboles binarios de búsqueda.</p> <p>Bibliografía “Estructuras de Datos y Algoritmos”, Aho Alfred, Hopcroft John y Ullman Jeffrey. Addison Wesley Publising Company. 1998. “Estructuras de Datos y Algoritmos”, Hernandez R. Dormido R. Lazaro J, Ros S. Pearson Education. 2000.</p> <p>Bibliografía Complementaria “TADs, Estructuras de datos y resolución de problemas con C++”. Larry R. Nyhoff. Prentice Hall. 2da edición. “Programación en C, C++, Java y UML”, Luis Joyanes Aguilar, Ignacio Zahonero Martínez. Mc Graw Hill.</p>	

Unidad 4 Análisis de algoritmos	Fecha Probable de Inicio y Finalización Desde el al
<p>Contenidos Complejidad de algoritmos, eficiencia de algoritmos. Análisis: tiempo de ejecución y utilización de memoria. Comportamiento en el mejor caso, caso promedio y peor caso. Análisis de eficiencia de distintos algoritmos (búsqueda, ordenación, recursivos, etc). Notación O(). Tipos de errores.</p> <p>Actividades Prácticas de la Unidad</p> <p>Trabajo práctico 6: análisis de complejidad de algoritmos iterativos, recursivos.</p> <p>Bibliografía “Estructuras de Datos y Algoritmos”, Aho Alfred, Hopcroft John y Ullman Jeffrey. Addison Wesley Publising Company. 1998. “Estructuras de Datos y Algoritmos”, Hernandez R. Dormido R. Lazaro J, Ros S. Pearson Education. 2000.</p> <p>Bibliografía Complementaria “TADs, Estructuras de datos y resolución de problemas con C++”. Larry R. Nyhoff. Prentice Hall. 2da edición.</p>	

Unidad 5 Programación Orientada a objetos	Fecha Probable de Inicio y Finalización Desde el al
<p>Contenidos Programación orientada a objetos. Concepto de abstracción, encapsulación y herencia. Objetos, mensajes, clases, instancias, métodos. Polimorfismo, sobrecarga de funciones. Templates. Diagramas de clase (UML). Implementación en C++: creación de clases, datos miembro, ocultación de datos (miembros public, private, protected), entrada, salida, constructores y destructores, herencia.</p> <p>Actividades Prácticas de la Unidad</p> <p>Trabajo práctico 7: programación orientada a objetos.</p> <p>Bibliografía “Programación en C, C++, Java y UML”, Luis Joyanes Aguilar, Ignacio Zahonero Martinez. Mc Graw Hill.</p>	

Bibliografía Complementaria

C++ Primer (5ta. Edición). Stanley B. Lippman, Josée Lajole, Barbara Moo. 2012.

C++ Primer Plus (6ta. Edición). Stephen Prata.

Propuesta de evaluación

Evaluación: la evaluación del curso puede consistir en la entrega de trabajos evaluatorios, en la aprobación de exámenes parciales o en ambos. La forma de entrega de los trabajos será presentando un informe del trabajo realizado, y de ser aplicable, el código fuente desarrollado (el profesor evaluará el correcto funcionamiento de dicho código).

Asignatura posible de ser promocionada sin examen final

No

Características del Sistema de Promoción

Requisitos de acreditación

Es requisito de acreditación que el alumno alcance una asistencia del 75% de las clases (teóricas y prácticas). El alumno puede registrar un porcentaje inferior de presencias en clases, si las justifica debidamente ante los docentes de la asignatura y el coordinador de la carrera.

Fechas tentativas de evaluaciones previstas

El primer parcial o trabajo evaluatorio se prevee en el transcurso de la Unidad 3. El segundo parcial o trabajo evaluatorio se prevee en el transcurso de las últimas 2 semanas de cursada, siendo la última semana de examen recuperatorio.

Vigencia del Programa		
2019	2020	2021
Firma y Aclaración Docente	Firma y Aclaración Docente	Firma y Aclaración Docente
Firma y Aclaración Director	Firma y Aclaración Director	Firma y Aclaración Director