

Nombre de las Asignaturas		Cuatrimestre de Dictado	Tipo	Calificación	Fecha	Créditos	Acta / Resolución
			Obligatorias /Optativas /Electivas				
CBC24	INTRODUCCIÓN AL CONOCIMIENTO DE LA	-	Obligatoria	5 (Cinco)	30/11/2018	4.00	UBAXXI
CBC40	INTRODUCCIÓN AL PENSAMIENTO CIENTÍFICO	-	Obligatoria	7 (Siete)	30/11/2018	4.00	UBAXXI
CBC05	QUÍMICA	-	Obligatoria	7 (Siete)	30/11/2018	6.00	UBAXXI
CBC62	ALGEBRA A	-	Obligatoria	2 (Dos)	19/07/2019	9.00	CBC16203
CBC03	FÍSICA	-	Obligatoria	9 (Nueve)	19/07/2019	6.00	CBC40301
CBC66	ANÁLISIS MATEMÁTICO A	-	Obligatoria	6 (Seis)	19/07/2019	9.00	CBC46608
CBC62	ALGEBRA A	-	Obligatoria	8 (Ocho)	29/11/2019	9.00	CBC46205
7540	ALGORITMOS Y PROGRAMACION I	3º	Obligatoria	8 (Ocho)	25/09/2020	6.00	95-0007361
7541	ALGORITMOS Y PROGRAMACION II	4º	Obligatoria	8 (Ocho)	05/04/2021	6.00	95-0007882
6107	MATEMATICA DISCRETA	-	Electiva	2 (Dos)	23/08/2021	6.00	61-0003495

OBSERVACIONES:

Firmado digitalmente,



Planificaciones

7540 - Algoritmos y Programación I

Docente responsable: ESSAYA DIEGO NICOLAS

OBJETIVOS

- Conocer un lenguaje de programación imperativo.
- Aplicar las capacidades que ofrece el lenguaje de programación elegido en el curso, en particular, estructuras de control, estructuras sencillas de datos e implementación de algoritmos elementales.
- Lograr habilidades para desarrollar programas sencillos.
- Aprender un conjunto de algoritmos básicos. Conocer y comprender algoritmos fundamentales utilizados con frecuencia, su desarrollo, análisis de eficiencia y corrección.
- Poder escribir programas de computadora en un lenguaje de programación imperativo. Poder compilar problemas (sencillos) y soluciones.
- Iniciarse en el diseño e implementación de estructuras de datos fundamentales.
- Desarrollar capacidades cognitivas y para la construcción de modelos y abstracción,
- Desarrollar una manera científica de abordar problemas (procedimiento metódico e inclusión del estado del arte),
- Desarrollar capacidad para documentar soluciones y su evaluación crítica, así como una presentación convincente.

CONTENIDOS MÍNIMOS

-

PROGRAMA SINTÉTICO

Concepto de algoritmo.
Concepto de programa.
Lenguajes de programación.
Tipos de datos simples.
Estructuras de control.
Modularización.
Tipos de datos estructurados.
Algoritmos básicos de ordenamiento y búsqueda.
Introducción a la complejidad computacional.
Archivos.
Tipos de datos abstractos.
Recursividad.

PROGRAMA ANALÍTICO

UNIDAD 1

Concepto de algoritmo. Introducción al manejo de la abstracción: del problema al algoritmo. Concepto de computadora y nociones básicas de su organización interna. Concepto de programa. Concepto de lenguaje de programación. Nociones y diferencias básicas entre compiladores e intérpretes. Estado y computación. Definición y uso de los conceptos de: tipo de dato, expresión, declaración, definición, sentencia. Entrada/ salida.

UNIDAD 2

Tipos de datos simples: enteros, reales, caracteres, lógicos; primitivas de cada tipo. Concepto y uso de constantes y variables. Operadores aritméticos. Prioridad de operaciones. Tipo de dato puntero. Definición de tipos.

UNIDAD 3

Sentencias de asignación, de bifurcación, de elección múltiple, de control de ciclos. Diseño y verificación de algoritmos. Introducción a las prácticas de documentación, verificación y prueba de programas. Manejo de errores. Introducción a los archivos de texto como entrada / salida de programas sencillos.

UNIDAD 4

Introducción a la modularización de programas mediante funciones. Conceptos de cohesión y acoplamiento de subprogramas. Ámbito de las variables y análisis de las prácticas correctas en los pasajes de las mismas a los módulos de un programa. Parámetros formales y reales. Pre y postcondiciones. Introducción a los métodos de división y conquista.

UNIDAD 5

Tipos de datos estructurados provistos por el lenguaje: arreglos unidimensionales y multidimensionales, cadenas, registros.

UNIDAD 6

Algoritmos básicos de ordenamiento y búsqueda en arreglos: selección, inserción, burbujeo, búsqueda secuencial, búsqueda binaria. Introducción a la complejidad computacional. Nociones de tiempo y espacio consumido. Comparación de algoritmos en cuanto a la complejidad computacional.

UNIDAD 7

Archivos: conceptos básicos, operaciones con archivos de texto y binarios. Procesamiento de archivos secuenciales y de acceso directo. Corte de control. Apareo de archivos. Manejo de índices.

UNIDAD 8

Iniciación al concepto de tipo de dato abstracto. Tipos de datos abstractos lista, pila, cola. Usos. Nociones básicas de recursividad.

BIBLIOGRAFÍA

Cátedra de Algoritmos y Programación I. Apuntes de cátedra.

John Zelle: \Python Programming: An Introduction to Computer Science, Franklin, Beedle & Associates, 2003.

Allen B. Downey: \Think Python: An Introduction to Software Design", 2008.
<http://www.greenteapress.com/thinkpython/>

Mark Lutz: \Learning Python (3rd Edition), OReilly Media Inc. 2007.
An Introduction to Python Paperback – March 1, 2011
by Guido Van Rossum , Fred L Drake Jr

N. Wirth: Algorithms + Data Structures = Programs. Editorial Perentice Hall.

N. Wirth: Introducción a la Programación Sistemática. Editorial El Ateneo.

B. Kernighan, D. Ritchie: El lenguaje de programación C. Editorial Pearson Educación

Apunte del curso: <https://algoritmos1rw.ddns.net/material>

RÉGIMEN DE CURSADA**Metodología de enseñanza****Clases teórico-prácticas**

Exposición teórica de conceptos fundamentales, con estudio de casos y resolución metódica de problemas tipo con ensayos sobre objetivos.

Clases prácticas

Resolución por parte de los alumnos -y orientada por los docentes- de problemas correspondientes a las unidades temáticas del programa, ya sea por escrito o por computadora. En general se tratará de problemas abiertos, que generen dudas y motiven la consulta a los docentes y la profundización del conocimiento a través de la bibliografía. Durante el curso se plantearán trabajos prácticos con problemas a resolver por programación, que los alumnos deberán desarrollar en grupo.

Modalidad de Evaluación Parcial

Desarrollo de trabajos prácticos por computadora.

El examen parcial es de tipo teórico-práctico e incluye la resolución de ejercicios, con dos oportunidades de recuperación.

La evaluación integradora escrita, de carácter teórico-práctico, con dos oportunidades de recuperación.

CALENDARIO DE CLASES

Semana	Temas de teoría	Resolución de problemas	Laboratorio	Otro tipo	Fecha entrega Informe TP	Bibliografía básica
<1> 09/03 al 14/03	Introducción a la materia. Concepto de algoritmo. Primeros pasos en Python. Sintaxis y gramática de Python. Variables. Funciones.					
<2> 16/03 al 21/03	Construcción de programas: análisis, especificación y diseño. Calidad de software. Expresiones booleanas. Operadores lógicos. Estructuras condicionales.					
<3> 23/03 al 28/03	Ciclos. Ciclos indefinidos. Uso de centinelas. Secuencias: Cadenas, tuplas y listas.					
<4> 30/03 al 04/04	Secuencias. Listas y mutabilidad. Buscar en listas, algoritmos de búsqueda (lineal, binaria). Pasaje de funciones como parametro.					
<5> 06/04 al 11/04	Map y filter. Listas por Comprension. Contratos: pre y post condiciones. Invariantes.					
<6> 13/04 al 18/04	Diccionarios. Parcial.				X	
<7> 20/04 al 25/04	Manejo de archivos. Excepciones. Manejo de errores.					
<8> 27/04 al 02/05	Recorridos de archivos: apareo y corte de control. Objetos. Propiedades. Creación de objetos. Abstracción.					
<9> 04/05 al 09/05	Listas enlazadas.					
<10> 11/05 al 16/05	Pilas y colas. Recursión.					
<11> 18/05 al 23/05	Métodos de ordenamiento sencillos.					
<12> 25/05 al 30/05	División y conquista. Ordenamientos recursivos.					
<13> 01/06 al 06/06	Lenguaje de programación C.					
<14> 08/06 al 13/06	Lenguaje de programación C.					
<15> 15/06 al 20/06	Recuperatorios.					
<16> 22/06 al 27/06	Recuperatorios.					

CALENDARIO DE EVALUACIONES

Evaluación Parcial

Oportunidad	Semana	Fecha	Hora	Aula
1º	6	13/04	17:30	
2º	16	22/06	19:00	
3º		04/12	19:00	
4º				



Planificaciones

7541 - Algoritmos y Programación II

Docente responsable: CALVO PATRICIA MABEL

OBJETIVOS

Que el estudiante, ante un problema, pueda indicar cuáles son los los Tipos de Datos Abstractos involucrados.

Que especifique correctamente las operaciones propias de cada TDA.

Que manipule adecuadamente distintas estructuras de datos.

Que aplique adecuadamente estrategias de resolución de problemas.

Que, dadas las especificaciones de un TDA, pueda plantear diversas implementaciones.

Que utilice elementos del análisis de algoritmos para comparar los algoritmos correspondientes a las distintas implementaciones.

Que pueda elegir la implementación más eficiente para un TDA.

Que justifique rigurosamente la elección de una implementación para un TDA.

Que elabore productos de software adecuadamente documentados.

Que trabaje en equipo de manera proactiva y eficiente.

CONTENIDOS MÍNIMOS

-

PROGRAMA SINTÉTICO

Unidad 1: TDA: especificación e implementación. Módulos y clases.

Unidad 2: Recursividad, tipos. Recursividad de cola. Estrategia “Divide y Vencerás”.

Unidad 3: TDA Vector, TDA Pila, TDA Cola. Implementaciones en arrays y en estructuras de listas ligadas. Pilas y recursividad.

Unidad 4: La complejidad de algoritmos. Eficiencia temporal y espacial. Concepto de Big Omicron, Big Omega; y Big Theta. Cálculo de complejidad de algoritmos iterativos y recursivos. Recurrencias básicas. Análisis de ordenamientos internos lentos y rápidos.

Unidad 5: TDA Conjunto. TDA Diccionario. Implementaciones con estructura de Árbol Binario de Búsqueda, y con árboles AVL. Estrategia “Vuelta atrás”. Implementación del TDA Conjunto con mapa de bits. Concepto de árbol B y uso en implementaciones del TDA Conjunto. Concepto de Trie y aplicaciones. Concepto de tablas de dispersión (hashing) y uso en implementaciones del TDA Conjunto.

Unidad 6: TDA Cola con Prioridad. La estructura montículo (heap) y sus aplicaciones.

Unidad 7: TDA Grafo. Recorridos básicos, prueba de aciclicidad, recorridos topológicos. Análisis de la estrategia “Voraz” (“Greedy”). Algoritmo de caminos mínimos (Dijkstra), caminos mínimos entre todos los pares de vértices (Floyd), Cerradura transitiva (Warshall). Estrategia de “Programación Dinámica”. Obtención del árbol de expansión de coste mínimo (Prim, Kruskal).

PROGRAMA ANALÍTICO

Unidad 1: Concepto de TDA: especificación e implementación de un TDA. Primitivas, pre y post condiciones. Concepto de Módulo y de Clase: principios de diseño. Los métodos formales de verificación. Manejo de la memoria dinámica a través de punteros y referencias.

Unidad 2: Recursividad: tipos, la recursividad ‘de cola’. Pasaje de algoritmos recursivos a iterativos y viceversa. Estrategia “Divide y Venceras” (“Divide and Conquer”).

Unidad3: El TDA Vector: especificación e implementaciones diversas. El TDA Lista: especificación e implementaciones en arrays y con estructura de lista ligada simple, doble, circular, doble circular, (iterativas y recursivas). El TDA Pila y el TDA Cola: concepto e implementaciones diversas, con arrays y con estructura de lista. El TDA Pila y la recursividad.

Unidad 4: Conceptos de complejidad algorítmica. La eficiencia temporal y espacial. Medidas asintóticas y notación Big Omicron, Big Omega y Big Theta. Aplicación en algoritmos iterativos y recursivos. Análisis de métodos de ordenamiento interno (lentos y rápidos - “Mergesort”, “Quicksort”. “Ordenamiento por Urnas”). Análisis de la eficiencia temporal y espacial de algoritmos diversos. Análisis de recurrencias básicas. Teorema Maestro de reducción por división.

Unidad 5: El TDA Conjunto, el TDA Diccionario: especificación e implementaciones diversas. Estructura de Árbol Binario de Búsqueda, concepto de balanceo, árboles AVL. Implementaciones con estructuras dinámicas utilizando punteros. Implementación con mapa de bits. La estrategia "Vuelta atrás" ("Backtracking"); aplicaciones a la resolución de problemas. Árboles multivías. Árboles B. Definición, Representación y Operaciones. Aplicaciones en Índices. Árbol B+ y B*. Concepto de árbol digital (Trie) e implementaciones. Las tablas de dispersión, el hashing, modos de resolución de colisiones. El Radix Sort.

Unidad 6: El TDA cola con prioridades; implementación con montículos (árboles Heap). El ordenamiento por montículo (Heapsort).

Unidad 7: El TDA Grafo: especificación e implementaciones. Recorridos en profundidad y anchura, prueba de aciclicidad, recorridos topológicos. Problemas de caminos: existencia de camino entre dos vértices, caminos mínimos con un mismo origen (Dijkstra), caminos mínimos entre todos los pares de vértices (Floyd), cerradura transitiva (Warshall), obtención del árbol de expansión de coste mínimo (Prim, Kruskal). La estrategia "Voraz" ("Greedy"). La estrategia de "Programación Dinámica".

BIBLIOGRAFÍA

"Estructuras de Datos y Algoritmos" Drozdek, A., Ed. Brooks & Cole, 2001

"Introduction to Algorithms", Cormen, Leiserson, Ed. McGraw-Hill, 1990

"Data Structures and Program Design in C++" Kruse, Ryba, Ed. Prentice Hall, 2005

"Algoritmos y Estructuras de datos" Weiss, Ed. Addison Wesley, 1995

"Estructuras de Datos y Algoritmos", Aho, Hopcroft, Ullman, Ed. Alhambra Mexicana, 1998

"Estructuras de Datos, especificación, diseño e implementación" Franch Gutierrez, Ed. Alfaomega, 2002

"El lenguaje de programación C++", Stroustrup, Ed. Addison Wesley, 2005

"Programming - Principles and Practice Using C++", Stroustrup, Ed. Addison Wesley, 2014

"C++, cómo programar", Deitel y Deitel, Ed. Pearsen, 2008

"Thinking in C++", Eckel, Ed. Pearson, 2004

"Técnicas de Diseño de Algoritmos" Guerequeta, Vallecillo, Servicio de Publicaciones de la Universidad de Málaga. 1998

RÉGIMEN DE CURSADA

Metodología de enseñanza

Clases teórico-prácticas

Exposición teórica de conceptos fundamentales, con resolución metódica de problemas tipo y ensayos sobre objetivos.

Clases prácticas

Resolución por parte de los alumnos y controlada por los docentes auxiliares de problemas correspondientes a las unidades temáticas del programa, ya sea por escrito o por máquina (programas). En general se tratará de problemas abiertos, que generen dudas y motiven la consulta a los docentes y la profundización del conocimiento a través de la bibliografía. Durante el curso se plantearán trabajos prácticos con problemas complejos a resolver por programación, que los alumnos deberán desarrollar en grupo

-Clases de consulta

Modalidad de Evaluación Parcial

De manejo de conceptos, aplicación de conocimientos y dominio de técnicas, mediante preguntas y resolución de problemas por escrito.

Las evaluaciones parciales e integradoras son por unidades o subunidades temáticas.

CALENDARIO DE CLASES

Semana	Temas de teoría	Resolución de problemas	Laboratorio	Otro tipo	Fecha entrega Informe TP	Bibliografía básica
<1> 09/03 al 14/03	Introducción a la materia	Revisión de conceptos básicos.				La indicada en sección "Bibliografía"
<2> 16/03 al 21/03	Introducción Punteros-Referencias	Ejercitación				
<3> 23/03 al 28/03	Concepto de TDA. La Abstracción.	Ejercitación.				
<4> 30/03 al 04/04	Recursividad. Tipos. Recursividad de cola. Eliminación de la recursividad. Concepto de estrategia 'Divide y Venceras'. Aplicaciones	Ejercicios de aplicación				
<5> 06/04 al 11/04	El TDA Vector. El TDA Lista	Implementaciones diversas				
<6> 13/04 al 18/04	TDA Lista	Ejercicios diversos. Implementaciones en estructuras de listas con punteros				
<7> 20/04 al 25/04	TDA Pila, TDA Cola.	Implementaciones diversas.				
<8> 27/04 al 02/05	Conceptos de complejidad algorítmica y aplicación en algoritmos iterativos y recursivos. Notaciones asintóticas. Propiedades.	Ejercicios de cálculo de complejidad de algoritmos				
<9> 04/05 al 09/05	Conceptos de complejidad algorítmica y aplicaciones. Recurrencias básicas. Teorema Maestro.	Análisis del coste temporal y espacial de diversos métodos de ordenamiento y búsqueda en arrays, operaciones básicas en estructuras estudiadas. Análisis de complejidad de algoritmos diversos.				
<10> 11/05 al 16/05	TDA Conjunto. TDA Diccionario. Mapas de bits. Estructura de	Resolución de problemas.				

Semana	Temas de teoría	Resolución de problemas	Laboratorio	Otro tipo	Fecha entrega Informe TP	Bibliografía básica
	Arbol Binario de Búsqueda Estructuras de arbol balanceadas.					
<11> 18/05 al 23/05	Arboles AVL. Conceptos básicos de Backtracking.	Ejercicios de aplicación				
<12> 25/05 al 30/05	Arboles multivías Arboles multivías balanceados (B) Tries y aplicaciones.	Resolución de problemas				
<13> 01/06 al 06/06	TDA cola con prioridades. Arboles heap y aplicaciones. Heapsort	Ejercitación				
<14> 08/06 al 13/06	El hashing. Utilización en la implementación del TDA Conjunto y TDA Diccionario.	Ejercitación.				
<15> 15/06 al 20/06	TDA Grafo. Implementaciones , recorridos, pruebas de aciclicidad, recorridos topológicos. Problemas de caminos: existencia de camino entre dos vértices, problema de caminos mínimos con un mismo origen en grafo orientado ponderado (Dijkstra) , La estrategia "Voraz".	Ejercitación.				
<16> 22/06 al 27/06	Problemas de caminos en grafos: Caminos entre todos los pares de vértices en grafo ponderado (Floyd). Algoritmo de Warshall, Concepto de programación dinámica. Problemas de Arboles de expansión de coste mínimo (algoritmo de Prim, Kruskal).	Ejercitación				

Semana	Temas de teoría	Resolución de problemas	Laboratorio	Otro tipo	Fecha entrega Informe TP	Bibliografía básica

CALENDARIO DE EVALUACIONES

Evaluación Parcial

Oportunidad	Semana	Fecha	Hora	Aula
1º	10	22/10	18:00	
2º	14	20/11	18:00	
3º	16	10/12	18:00	
4º				



Dirección de
Alumnos

Por la presente se certifica que corresponde al Plan de Estudios **1.986**, de la carrera de **INGENIERÍA EN INFORMÁTICA** , con que el señor **BEVILACQUA BRUNO**, DNI **42.146.206** , Legajo N° **105.534**, figura inscripto en esta Facultad. Se deja constancia que las asignaturas del primer y segundo cuatrimestre, corresponden al Ciclo Básico Común y que los créditos que figuran a continuación de cada materia, corresponden a la cantidad de horas de clase teórico- prácticas semanales que tiene asignada la misma. A su pedido y al solo efecto de ser presentado antes las autoridades que correspondan, se extiende el presente en la Ciudad Autónoma de Buenos Aires, República Argentina a los 22 días del mes de Noviembre de 2.021.



Certificado

Hoja Adicional de Firmas

1821 Universidad de Buenos Aires

Número:

Referencia: BEVILACQUA BRUNO

El documento fue importado por el sistema GEDO con un total de 15 pagina/s.