

Teoría de la Computación

El siguiente texto describe las reglas, metodología y temario del curso de "*Teoría de la Computación*" a dictarse en el segundo semestre de 2023 por Guillermo Apollonia (primera parte) y Mauricio Velasco (segunda parte). La cátedra se reserva el derecho de corregir, modificar y ampliar el siguiente documento según crean conveniente.

Objetivos del curso

La Teoría de la Computación es el área de las ciencias de la computación y las matemáticas que trata con modelos de cómputo y algoritmos. Comenzando con los trabajos de Alan Turing y Alonzo Church (*et alia*), se intenta determinar cuáles son las capacidades y limitaciones de las computadoras y de los algoritmos que pueden ejecutarse en éstas. Al final del curso el estudiante debería ser capaz de:

- Estimar la viabilidad teórica de resolver un problema de cómputo con una computadora.
- Plantear algoritmos de cálculo simples en términos de Máquinas de Turing.
- Demostrar si un algoritmo es o no correcto rigurosamente.
- Analizar algoritmos (iterativos o recursivos) y estructuras de datos para estimar su eficiencia así como familiarizarse con grupos distinguidos de algoritmos (de programación dinámica, greedy, randomizado, etc.)

Estructura general del curso

El contenido del curso se divide en siete temas principales, marcándose para cada uno de ellos lecturas previas obligatorios (ver temas y lecturas asignadas abajo).

Calificaciones y aprobación del curso

Para la calificación del curso se tendrán en cuenta los siguientes componentes:

Actividad	Parte del curso	Porcentaje	Modalidad
3 Prácticos	Parte 1	25%	Grupal
1 Parcial	Parte 1	25%	Individual
3 Prácticos	Parte 2	25%	Grupal
1 Parcial	Parte 2	25%	Individual

A esto se suma un **bono de 5% adicional** para aquellos estudiantes que hagan al menos el 50% de las entregas en LaTeX. La nota numérica final N se calculará redondeando el promedio ponderado de las notas obtenidas arriba. La nota en letras final se decidirá a partir de la asistencia y la nota numérica de acuerdo a las siguientes condiciones:

- Para ser **elegible para aprobar el curso** el estudiante debe asistir a por lo menos el 60% de las clases (es decir, el estudiante reprueba independientemente de la nota que tenga si su asistencia es 59% o menos).
- Para **ser elegible para exonerar la materia** el estudiante debe asistir a por lo menos el 75% de las clases y tener una nota numérica N de por lo menos el 83% (es decir, debe dar examen si tiene).

Adicional al requisito de asistencia del párrafo anterior la nota numérica N determina la nota final así:

- D: 0 - 30
- R: 31 – 60
- B: 61 – 74
- BMB: 75 – 86
- MB: 87 – 93
- S: 94 – 100

Requisitos para cursar

El estudiante debe contar con acceso a la Web Asignatura del curso, y utilizarlo. Será responsabilidad del alumno asegurar su correcto acceso a ésta, recurrir inmediatamente al departamento correspondiente para solucionar rápidamente el inconveniente y notificar a los docentes de la situación.

Pruebas individuales y en equipo

La no asistencia a una prueba implicará una calificación de 0% en la prueba correspondiente. De la misma forma, la no participación en el desarrollo de un trabajo de aplicación (por no presencia física) puede implicar una calificación de 0% en ese trabajo.

Las evaluaciones individuales o en equipos de cada unidad podrán ser tomados utilizando cuestionarios de la webasignatura.

En cada oportunidad la cátedra comunicará (mediante la Webasignatura) la política en vigencia sobre disponibilidad de materiales durante las diferentes pruebas, así como también disposiciones con respecto al uso de dispositivos electrónicos (computadoras, teléfonos, tabletas, etc.).

Webasignatura

La Webasignatura de la materia y los links en ella referidos contienen todos los materiales puestos a disposición de los alumnos por los docentes. El **foro general del sitio** es el medio de comunicación básico de los docentes.

Es responsabilidad de cada alumno verificar, una vez cerrado el plazo de inscripciones al curso, su estado de registro en la webasignatura, y la consulta habitual a la misma para tomar conocimiento de todas las comunicaciones y avisos efectuados por los docentes, referentes a cualquier aspecto relacionado con el desarrollo de la asignatura.

La no lectura de los avisos publicados en la misma o en los links por esta referidos no constituirá en ningún caso justificativo para el no cumplimiento de tareas, asistencias o cualquier otra actividad indicada por los docentes. En caso de malfuncionamiento o avería de la web asignatura, los docentes publicarán los avisos en las carteleras de la facultad, siendo responsabilidad de cada alumno el tomar conocimiento de los mismos.

Los foros de la Webasignatura son exclusivamente para discusión en los temas académicos de la asignatura y anuncios por parte de los docentes. Por tanto, no podrán ser utilizados para ningún otro fin. Las consultas por temas administrativos, de agenda, gestión o personales deberán ser tratadas personalmente con los docentes o en su defecto con las autoridades correspondientes de la Carrera. Los docentes eliminarán de los foros todo mensaje que no se ajuste a estas normas.

Entregables

Si el trabajo de aplicación requiere la realización de programas, en ningún caso los docentes ejecutarán el programa objeto, sino que compilarán todo el programa y luego lo correrán. Por ello, deben entregarse todos los archivos de código fuente que sean necesarios para realizar una compilación completa, así como también las instrucciones correspondientes (directorios, directivas de compilación utilizadas, etc.). La cátedra actuará como si el código no hubiese sido entregado, calificándolo con un 0%, si sucede cualquiera de las siguientes situaciones:

- No es posible compilar y ejecutar el programa.
- El código compila, pero no puede ejecutar o no ejecuta en un tiempo razonable.
- Los archivos que componen la entrega están corruptos o infectados con cualquier forma de malware.
- Similitud sospechosa (a criterio de los docentes) total o parcial con otras entregas de otros equipos, sin perjuicio de iniciar las acciones indicadas en el Reglamento general de estudiantes para los casos de fraude académico, en caso que los docentes lo consideren pertinente.

Los programas se deberán ajustar estrictamente a lo planteado, tanto en el análisis como en la solución del problema, no pudiendo realizarse ningún cambio a los formatos de los archivos de datos de entrada o salida, ni a la presentación del programa o su interfaz de usuario (en caso de existir alguna).

Fechas de parciales

- Parcial 1: 28 de Setiembre de 2023
- Parcial 2: 30 de Noviembre de 2023

Cronograma

En cada tema se destina el primer módulo para el aseguramiento de la preparación (*readiness assurance*). Las lecturas necesarias y complementarias se detallarán en la página del curso en Webasignatura, en la sección del cronograma correspondiente a la unidad temática.

Tema 0 Introducción al curso. [Primera clase]

El propósito general de esta clase es que los alumnos comprendan los objetivos principales de la asignatura, qué se espera que sean capaces de realizar al finalizar la misma, y que comiencen a conocer los aspectos fundamentales de la metodología pedagógica que ha de utilizarse a lo largo del curso. Al finalizar el alumno deberá ser capaz de:

- Describir los objetivos de aprendizaje del curso, su contenido y la metodología que se aplicará.
- Planificar su plan de estudios en base al cronograma del curso.

Tema 1 Computabilidad y Máquinas de Turing [3 a 4 semanas]

Comenzando por el principio, la primer unidad temática se enfoca en el trabajo fundacional de Alan Turing. Se trabajará con *máquinas de Turing* como modelo computacional para decisión y cálculo; sus posibles variantes y equivalencias, terminando con la *Máquina de Turing Universal*. Luego se definen lenguajes aceptables y decidibles, previo a cubrir el *Halting Problem*. Se termina con argumentos de no computabilidad por reducción.

Al final de esta unidad temática el estudiante debería ser capaz de:

- Construir una máquina de Turing para un problema de decisión o cálculo muy simple.
- Identificar el lenguaje de una máquina de Turing simple.
- Explicar el concepto de Máquina de Turing Universal.
- Argumentar que un problema no es decidible mediante reducción del Halting Problem al mismo.

La preparación necesaria para esta unidad temática es:

- Lectura esencial [HOPCROFT2001] Capítulo 8 "*Introduction to Turing Machines*", secciones 8.1, 8.2, 8.4 y 8.5.
- Lectura esencial [HOPCROFT2001] Capítulo 9 "*Undecidability*", secciones 9.1, 9.2 y 9.3.

Tema 2 Algoritmos y su análisis [2 a 3 semanas]

En ésta unidad se repasa el cálculo de órdenes de ejecución dado en las asignaturas de *Algoritmos y Estructuras de Datos* y expande agregando algoritmos recursivos y el *teorema maestro*. Se repasa recursividad.

Al final de la unidad temática el alumno deberá ser capaz de:

- Estimar el orden de ejecución de un algoritmo iterativo o recursivo.
- Verificar empíricamente el resultado teórico del orden de ejecución.

La preparación necesaria para esta unidad temática es:

- Lectura esencial [CORMEN2002] Capítulo 3 "*Growth of Functions*", 4 "*Recurrences*", 34 "*NP-Completeness*".
- Lectura esencial [HOPCROFT2001] Capítulo 10 "*Intractable Problems*", secciones 10.1, 10.2 y 10.3.

Tema 3 Programación Dinámica [2 a 3 semanas]

En ésta unidad se definirá la programación dinámica y mediante resolución de problemas se compararán las soluciones con otros enfoques.

Al finalizar la unidad temática el alumno deberá ser capaz de:

- Identificar problemas que puedan ser resueltos con PD.
- Definir algoritmos con programación dinámica

La preparación necesaria para esta unidad temática es:

- Lectura esencial [CORMEN2002] Capítulo 4.15 "*Dynamic Programming*".

Tema 4 Programación codiciosa [1 a 2 semanas]

Presentación del tema *Greedy algorithms* y ejemplificación.

Al finalizar la unidad temática el alumno deberá ser capaz de:

- Definir algoritmos con PC en los que se escoge un óptimo local en cada instante
- Identificar problemas que puedan ser resueltos de manera óptima mediante PC.

La preparación necesaria para esta unidad temática es:

- Lectura esencial [ROUGHGARDEN_3, 2019] Capítulo 13 "*Introduction to Greedy Algorithms*".

Tema 5 Algoritmos probabilísticos [1 a 2 semanas]

En ésta unidad temática se estudian los algoritmos probabilísticos, enfocándose en los algoritmos de *Monte Carlo* y *Vegas*. Al final de la unidad temática el alumno deberá ser capaz de:

- Convertir un algoritmo determinístico en uno probabilístico.
- Estimar la probabilidad de la correctitud de un algoritmo de *Monte Carlo*.
- Verificar empíricamente el resultado teórico de la probabilidad de correctitud de un algoritmo *Monte Carlo*.

La preparación necesaria para esta unidad temática es:

- Lectura esencial [CORMEN2002] Capítulo 5 "*Probabilistic Analysis and Randomized Algorithms*".

Tema 6 Clases de complejidad [1 a 2 semanas]

En ésta unidad temática se definirán las clases de complejidad P, NP y PSPACE y se hablara de reducibilidad polinomial entre problemas. Se definirá también el concepto de problema completo para una clase de complejidad. Al final de la unidad temática el alumno deberá ser capaz de:

- Definir las clases de complejidad
- Demostrar algunas reducciones polinomiales entre distintos problemas
- Definir el concepto de problema completo y dar ejemplos de problemas completos para las clases P, NP y PSPACE.
- Entender el significado de la pregunta P vs NP.
- Lectura esencial [KLEINBERGandTARDOS2006] Capítulo 8.1 "*Polynomial time reductions*".

Tema 7 Grafos [1 a 2 semanas]

Se repasarán conceptos y algoritmos básicos de grafos, pasando por búsqueda en amplitud, búsqueda en profundidad, entre otros.

Al finalizar el tema el alumno deberá ser capaz de:

- Describir y resolver con rigor matemático, algoritmos que involucren grafos.

Bibliografía.

La bibliografía y otros recursos necesarios (lecturas, videos, ejemplos, ejercicios, recursos web, etc.) para la preparación de cada unidad temática será oportunamente especificada por la Cátedra en la sección correspondiente de cada tema en la Webasignatura.

En términos generales, se utilizarán los siguientes libros:

- [CORMEN2002] Thomas Cormen et al: ["Introduction to Algorithms"](#). MIT Press, 2nd Edition. ISBN 0-262-03293-7.
- [HOPCROFT2001] John Hopcroft, Rajeev Motwani, Jeffrey Ullman: ["Introduction to automata theory, languages, and computation"](#), 2nd edition, Addison-Wesley, 2001. ISBN 0-201-44124-1.
- [KLEINBERGandTARDOS2006] Kleinberg Jon and Tardos Eva: ["Algorithm Design"](#), 1st Edition, Addison-Wesley, 2001. ISBN 0-321-29535-8.
- [NIELSON1999] H.R. Nielson & F. Nielson: ["Semantic with Applications: A formal introduction"](#), John Wiley & Sons, revised edition ISBN 0-471-92980-8.
- [ROUGHGARDEN_3,2019] Roughgarden Tim: ["Algorithms Illuminated Part 3: Greedy algorithms and dynamic programming"](#), Soundlikeyourself Publishing, LLC 2019.
- [SLONNEGER1995] Kenneth Slonneger & Barry Kurtz: ["Formal syntax and semantics of programming languages: a laboratory based approach"](#), Addison-Wesley, 1995. ISBN 0-201-65697-3.

Fin