

## **ALGORITMOS I**

**Profesor: Fabiana Daian**

**Día y Horario: Miércoles de 18:45 a 22:45 modalidad online sincrónica**

### **OBJETIVOS**

- Adquirir conocimientos de la teoría de algoritmos y en particular en aquellos temas que le sirvan para el desarrollo de aplicaciones prácticas
- Familiarizarse con la problemática de eficiencia en tiempo versus eficiencia en espacio (Time Complexity vs Space complexity)
- Adquirir conocimientos específicos de algoritmos no numéricos (recorrido de grafos, algoritmos basados en autómatas finitos, algoritmos de parsing)
- Introducir el concepto de algoritmos heurísticos para la solución de problemas complejos (genéticos, gradientes)
- Contribuir al desarrollo de las siguientes competencias:

Competencia	Relación con la asignatura
Identificación, formulación y resolución de problemas de informática	En esta asignatura el estudiante afronta sus primeros problemas de análisis de algoritmos en informática. A través de los requerimientos que le genera la concepción, diagramación y programación de algoritmos, inicia su desarrollo de capacidades de identificación, formulación y resolución de problemas, poniendo énfasis en la eficiencia, balance entre tiempo de proceso y espacio en memoria y complejidad del algoritmo.
Utilización de técnicas y herramientas de aplicación en informática	El estudiante adquiere conocimientos de las herramientas utilizadas en el mundo corporativo que le permiten analizar y desarrollar mejores soluciones algorítmicas. Además aplica distintas estrategias de diseño de algoritmos.
Generación de desarrollos tecnológicos y/o innovaciones tecnológicas	El estudiante adquiere conocimientos de las herramientas utilizadas en el mundo corporativo que le permiten analizar y diseñar mejores soluciones alineadas a las necesidades del cliente. El estudiante genera propuestas innovadoras cuando se lo requieren los problemas planteados.

## **CONTENIDOS MÍNIMOS**

El problema de eficiencia de los algoritmos. C. Tiempo de ejecución. Notación O y W. Balance entre tiempo y espacio en los algoritmos. Análisis práctico del tiempo de ejecución de un algoritmo. Conceptos básicos de la teoría de computabilidad y complejidad. Problemas computables y no computables. Notación asintótica. Complejidad de algoritmos. Algoritmos generalizados. Análisis de algoritmos. Análisis de la complejidad de algoritmos: Análisis asintótico. Análisis numérico. Solución aproximada de sistemas de ecuaciones lineales. Comportamiento en el mejor caso, caso promedio y peor caso. Determinar la complejidad asintótica experimentalmente. Algoritmos recursivos. Algoritmos de ordenamiento. Distintos medios de ordenamiento. Ordenamiento óptimo. Estrategias de diseño de algoritmos. Búsquedas de sub-cadenas.

## **PROGRAMA ANALÍTICO**

### **Unidad N°1: EL PROBLEMA DE EFICIENCIA DE LOS ALGORITMOS**

#### **Módulo 1: Eficiencia de los algoritmos**

La eficiencia de los algoritmos. Métodos de evaluación. La notación asintótica de Brassard y Bratley. Análisis teórico de ejecución de un algoritmo.

Resolución de problemas comparando tiempos de ejecución de los mismos.

### **Unidad N° 2: TIEMPO DE EJECUCIÓN**

#### **Módulo 2: Análisis del tiempo de ejecución**

Análisis teórico del tiempo de ejecución de un algoritmo. Análisis práctico del tiempo de ejecución de un algoritmo. Análisis de programas con llamadas a procedimientos. Análisis de procedimientos recursivos. Algunos ejemplos prácticos.

Ejercicios comparando tiempos de ejecución y optimización.

### **Unidad N°3: INTRODUCCIÓN A LA COMPLEJIDAD**

#### **Módulo 3: Notación Asintótica**

## **Facultad de Ingeniería**

Notación asintótica. Problemas P y NP. Problemas computables y no computables. Ejemplos. Notación asintótica condicional. Operaciones con notación asintóticas.

### **Unidad N°4: COMPLEJIDAD DE ALGORITMOS**

#### **Módulo 4: Complejidad de los Algoritmos**

Cálculo de métricas. Algoritmos generalizados. Análisis de los algoritmos. Bajo promedio. Peor caso. Análisis con O definición formal de la notación asintótica. Determinar la complejidad asintótica experimentalmente.

### **Unidad N°5: ANÁLISIS DE ALGORITMOS**

#### **Módulo 5: Recurrencias homogéneas**

Resolución de recurrencias. Recurrencias homogéneas. Método iterativo de Jacoby. Método iterativo de Gauss-Siede: Polinomio interpolador Polinomio interpolador de Lagrange. Acotación del error. Polinomio interpolador en la forma de Newton.

#### **Módulo 6: Recurrencias no homogéneas**

Recurrencias in homogéneas, cambio de variables, formaciones del rango, recurrencia asintótica.. Separación de raíces reales. Teorema de Bolzano. Método de aproximación de Newton-Raphson. Método del punto fijo. Condiciones de convergencia. Acotación del error.

### **Unidad N°6: ORDENAMIENTO**

#### **Módulo 7: Ordenamiento iterativo**

Distintos medios de ordenamiento: Inserción, intercambio, selección, fusión, distribución. Ordenamientos óptimos.

#### **Módulo 8: Ordenamiento recursivo**

Recursividad. Tipos. Algoritmos recursivos. Vectores. Distintos medios de ordenamiento: Merge Sort, Quick sort.

### **Unidad N°7: ESTRATEGIAS DE DISEÑO DE ALGORITMOS**

## Facultad de Ingeniería

### Módulo 9: Programación dinámica y Dividir y conquistar

Diseño por fuerza bruta. Diseño por inducción, división y programación dinámica. Programación dinámica, Dividir y conquistar.

### Módulo 10: Algoritmos voraces y de vuelta atrás

Branco and bound. Algoritmo de Greedy. Los problemas de asignación. Algoritmos genéticos. La programación genética. El problema del viajante.

## Unidad N°8: BÚSQUEDAS DE SUB-CADENAS

### Módulo 11: Búsqueda aproximada

Definición del problema. Aplicaciones para Búsqueda bruta. Búsqueda aproximada. Sub-cadenas repetidas.

### Módulo 12: Algoritmo Boyer-Moore

El algoritmo de Boyer-Moore. Resolución de casos de situaciones reales.

## BIBLIOGRAFÍA OBLIGATORIA

- Brassard, G. y Bratley, P. (2008). *Fundamentos de Algoritmia*. Madrid: Prentice Hall.
- Cormen, T., Leiserson, C., Rivest, R. y Stein C. (2009). *Introduction to Algorithms* (3<sup>a</sup> ed.). Massachusetts: MIT Press.
- Guerequeta, R. y Vallecillo. A. (2000). *Técnica de diseño de algoritmos*. Servicio de publicaciones de la Universidad de Málaga. Recuperado de <http://www.lcc.uma.es/~av/Libro/>
- Joyanes Aguilar, L. y Zahonero Martínez, I. (2001). *Programación en C: metodología, algoritmos y estructura de datos*. Madrid: McGraw-Hill.

## **BIBLIOGRAFÍA COMPLEMENTARIA**

- Kernighan, B. W. y Ritchie, D. M. (1992). *El lenguaje de programación C* (2ª ed.). México: Prentice Hall Hispanoamericana.
- Knuth, D. E. (2002). *El arte de programar ordenadores*. Barcelona: Reverté.

## **ACTIVIDADES DE FORMACIÓN PRÁCTICA**

Las actividades prácticas planteadas en la asignatura permitirán el desarrollo de habilidades y competencias. Para ello:

- Se le solicitará al estudiante una activa participación, ya que deberá realizar actividades de manera individual o grupal.
- La formación práctica comienza desde el diseño de algoritmos que puedan representar una situación real, y luego codificar estos algoritmos en el lenguaje de programación C, Java o Python. Estas actividades serán propuestas en todas las unidades que lo ameriten.
- La resolución de problemas requerirá tanto el desarrollo del pensamiento lógico de los estudiantes como la capacidad de elaborar y testear un programa en máquina que lo resuelva.

## **Desarrollo de las competencias**

Las competencias se desarrollan a través de las actividades prácticas, en forma acumulativa junto con las otras asignaturas de la carrera. A continuación, se incluye una tabla con las actividades que aportan al desarrollo de cada competencia relacionada con esta asignatura.

Competencia	Actividades
Identificación, formulación y resolución de problemas de informática.	Act.: Resolver ejercicios de ordenamiento de arreglos recursivo Act.: Resolver Ejercicios aplicando las técnicas de diseño de algoritmos.
Utilización de técnicas y herramientas de aplicación en informática.	Act: Aplicar algoritmos de búsqueda en cadena de textos Act. Aplicar técnicas de diseño de algoritmos Act: Diseñar algoritmos con una determinada complejidad
Generación de desarrollos tecnológicos y/o innovaciones tecnológicas	Act: Diseñar algoritmos con una complejidad determinada. Act: Utilizar modelos de algoritmos según su técnica de diseño.

### DISTRIBUCIÓN DE LAS ACTIVIDADES FORMATIVAS

Se prevé la siguiente distribución de actividades y cargas de estudio:

- La carga horaria de esta asignatura es de 68 horas distribuidas en las semanas que comprende el semestre académico.
- En el Calendario correspondiente a cada ciclo lectivo, las fechas de su inicio y final (incluyendo su examen final) podrán consultarse en la web de la Facultad de Ingeniería.

### METODOLOGÍA DE EVALUACIÓN Y EVIDENCIAS DEL APRENDIZAJE

Las habilidades y competencias conforman parte esencial de la formación del profesional en Informática. Estas se demuestran a partir de las actividades, por lo tanto el acento de las evaluaciones estará sobre las actividades más que en la expresión de conocimientos.

Las actividades que involucran: creación, desarrollo, análisis, y resolución, son las que más evidencian el estado del proceso de aprendizaje del estudiante.

Las evaluaciones se materializan a través de:

- exámenes escritos
- exámenes orales
- actividades de resolución de problemas
- actividades de desarrollo de sistemas

En el caso de las competencias para las cuales esta asignatura contribuye a desarrollar se detallan a continuación las evidencias particulares para cada una:

Competencia	Evidencias
Identificación, formulación y resolución de problemas de informática.	El estudiante identifica y delimita el problema a resolver. Analiza correctamente el requerimiento a desarrollar. Propone alternativas viables funcional y computacionalmente para alcanzar el algoritmo óptimo.
Utilización de técnicas y herramientas de aplicación en informática.	El estudiante utiliza correctamente el entorno de desarrollo para programación, compilación y debugging durante su trabajo en los requerimientos de desarrollo en las guías.
Generación de desarrollos tecnológicos y/o innovaciones tecnológicas	El estudiante construye diferentes algoritmos iterativos y recursivos para llegar al más adecuado de acuerdo a su complejidad.

Estas mismas competencias completan su desarrollo a lo largo de toda la carrera a través de aquellas asignaturas con las que se relacionan específicamente según se indica en sus syllabus respectivos.

### REQUISITOS DE APROBACIÓN

La aprobación de la asignatura requerirá aprobar su Cursado y su Examen Final, en los que deberá demostrar conocimiento de los contenidos y el desarrollo de las competencias

## Facultad de Ingeniería

asociadas a esta materia. El grado esperado de desarrollo de las competencias estará acorde a la posición relativa de esta asignatura en el plan de estudios.

Es obligatorio tener una asistencia al 75% de las clases.

### Requisitos para la aprobación del cursado

La evaluación de cursado por parte del profesor instructor se realiza mediante una rúbrica con los siguientes coeficientes de ponderación:

Actividad	Ponderación en %
Completar la totalidad de las actividades prácticas de cada módulo individuales y/o grupales, realizando las diferentes tareas que se proponen en cada uno de los mismos	30
Concepto del profesor referente a la participación del estudiante en el curso y al avance en el proceso de adquisición de las habilidades y competencias vinculadas a la asignatura	15
Aprobar dos exámenes parciales	55
Total	100%

- Los exámenes parciales se aprueban con una nota igual o superior a 4 (cuatro). Un porcentaje menor de respuestas correctas implica la reprobación del examen.
- Las evaluaciones durante el cursado serán de carácter formativo e incluirán una retroalimentación del profesor.
- Cualquier obligación académica no cumplida o reprobada podrá ser recuperada. Los exámenes de recuperación de parciales serán tomados dentro del período de cursada o dentro del período que el calendario académico indique para la recuperación.

La aprobación del cursado habilitará al estudiante a rendir el examen final.

## **Facultad de Ingeniería**

### **Requisitos para la aprobación del examen final**

La evaluación de la asignatura se completa mediante la evaluación de un Examen Final.

El profesor formulará sus exámenes de forma tal que evalúen que el estudiante haya alcanzado los objetivos de aprendizaje de la asignatura. Se incluirá en la evaluación los temas indicados en el Syllabus vigente de la asignatura, tanto en su faz teórica como en su faz práctica.

Las evaluaciones comprenderán preferentemente diversas modalidades y tipologías de preguntas y actividades que permitan evaluar los conocimientos, habilidades y competencias del estudiante.

Las evaluaciones y la calificación son siempre individuales.

Los exámenes se aprueban con una nota mínima de 4 (cuatro) puntos que implica mayor porcentaje de respuestas correctas..