

**INSTITUTO/S:** Tecnología e Ingeniería

**CARRERA/S:** Licenciatura en Informática

**MATERIA:** Algoritmos

**NOMBRE DEL RESPONSABLE DE LA ASIGNATURA:** Cristian Ciarallo

**EQUIPO DOCENTE:**-----

**CUATRIMESTRE:** 2<sup>do</sup>

**AÑO:** 2<sup>do</sup>

**PROGRAMA N°:** 15

**(Aprob. Por Cons.Directivo 04/06/2022)**

**Instituto/s:** Tecnología e Ingeniería

**Carrera/s:** Licenciatura en Informática

**Nombre de la materia:** Algoritmos

**Responsable de la asignatura y equipo docente:** Cristian Ciarallo

**Cuatrimestre y año:** 2<sup>do</sup> del 2<sup>do</sup> año

**Carga horaria semanal:** 6 hs

**Programa N°:** 15

**Código de la materia en SIU:** 768

## Algoritmos

### 1. Fundamentación

El objeto de estudio de la materia es el análisis de algoritmos complejos, mediante la medición de su complejidad computacional, para que el/la alumno/a pueda entender y evaluar entre distintas soluciones de problemas específicos, cuál es la mejor. Esto implica un aporte fundamental al área de acción principal del egresado que es la problemática de la construcción de software, que se corresponde con las tareas de análisis, diseño y programación.

### 2. Propósitos y/u objetivos

#### Objetivos

Que los/as alumnos/as

- incorporen la noción formal de orden de complejidad temporal y espacial de un algoritmo para el peor caso y el caso promedio.
- elijan entre dos algoritmos en base a su propio cálculo de complejidad temporal y espacial en peor caso para los mismos.
- pueden desarrollar algoritmos más eficientes que los triviales para distintos problemas, mediante el uso de las técnicas algorítmicas de Divide y Vencerás y Programación Dinámica.
- encuentren un algoritmo a problemas de decisión, enumeración u optimización de cierta complejidad mediante el uso de Backtracking.
- conozcan los algoritmos de Precondicionamiento Transformación de Dominio y los problemas de propagación de errores en los algoritmos numéricos.
- conozcan los problemas clásicos de grafos, sus soluciones algorítmicas más importantes y sus aplicaciones más relevantes.
- conozcan los problemas clásicos sobre cadenas, sus soluciones algorítmicas más importantes y sus aplicaciones más relevantes, en particular aquellas de la bioinformática.

### 3. Programa sintético:

Noción de algoritmo, ejemplos de algoritmos (criba de Eratostenes, mcd, etc). Criterios de selección de un algoritmo. Notación O y W. Análisis teórico del tiempo de ejecución de un

algoritmo Análisis práctico del tiempo de ejecución de un algoritmo. Algoritmos Divide y Vencerás. Recursividad. Análisis de procedimientos recursivos sobre estructuras complejas. Algoritmos Basados en Programación Dinámica. Estrategias de diseño de algoritmos. Algoritmos Greedy. Algoritmos de programación matemática. Uso de Heurísticas en Algoritmos. Algoritmos numéricos y propagación del error. Casos: algoritmo de Huffman, encriptación, compresión, búsqueda, actualización, ordenamiento, estructuras de datos y algoritmos, árboles estrella, matrices. Algoritmos sobre grafos (DFS, BFD, Prim, Kruskal, Dijkstra, Floyd, sort topológico, etc). Algoritmos básicos sobre cadenas: matching, alineamiento, sufijos. Hashing. TDA Conjunto y Diccionario. Análisis de complejidad de métodos de ordenamiento: quick - merge - radix - tree - heap y shell sort.

#### **4. Programa analítico**

##### **4.1 Organización del contenido:**

UNIDAD 1: Estructura de la memoria - Punteros a datos simples - Pedido de memoria y liberación

Punteros a estructuras complejas – Doble. Análisis de algoritmos: Análisis asintótico, comportamiento en el mejor caso, caso promedio y peor caso. Modelo computacional. Concepto de tiempo de ejecución. Notación  $O()$ ,  $\Omega$ ,  $\Theta$ . Reglas generales para el cálculo del tiempo de ejecución. Cálculo de tiempo y orden de ejecución en algoritmos iterativos y recursivos. Análisis de los métodos de ordenamiento: quick - merge - radix - tree - heap y shell sort Comparación de distintas estrategias de diseño de algoritmos.

UNIDAD 2: Grafos orientados y no orientados. Grafos pesados. Distintas representaciones: Listas de Adyacencia y Matriz de Adyacencia. Definiciones básicas y conceptos fundamentales. Grafos acíclicos. Grafos conexos y dígrafos fuertemente conexos. Árboles balanceados – AVL; árboles multivías - árboles B - B+ y B\*

UNIDAD 3: Algoritmos de recorrido: DFS y BFS. Árbol generador DFS: en grafos dirigidos y no dirigidos. Determinación de componentes conexas y fuertemente conexas. Análisis del tiempo de ejecución de los algoritmos mencionados.

UNIDAD 4: Ordenamiento topológico. Ejemplos de aplicación. Distintas implementaciones. Análisis de la eficiencia de cada uno. Problema del camino mínimo: estudio de distintos casos. Su desarrollo para grafos pesados y no pesados; y grafos dirigidos y acíclicos. Algoritmos de Dijkstra y Floyd. Árbol generador mínimo. Algoritmos de Prim y Kruskal. Análisis del tiempo de ejecución de los algoritmos vistos,

UNIDAD 5: Algoritmos básicos sobre cadenas: matching, alineamiento, sufijos. Algoritmos concurrentes, distribuidos y paralelos. Estrategias de implementación. Casos de uso. Funciones de hashing: Tipos. Colisiones: Hash cerrado y abierto. Funciones de hashing perfectas.

##### **4.2 Bibliografía y recursos obligatorios:**

**Cormen, T.H.; Stein, C.; Rivest, R. L.; Leiserson, Ch. E.** (2009). *Introduction to Algorithms (3rd ed.)*. McGraw-Hill Higher Education.

**Wolsey, L. A.** (1998). *Integer Programming*. Wiley-Interscience

**Weiss, M.A.** (2012). *Data Structures and Algorithm Analysis in Java, 3<sup>rd</sup> Edition*. Addison-Wesley.

#### **4.3 Bibliografía optativa:**

**Sedgewick, R.; Wayne, K.** (2011). *Algorithms*. (4th Edition). Addison-Wesley Professional

**Talbi, El-Ghazali.** (2009). *Metaheuristics: From Design to Implementation*. Wiley.

### **5. Metodologías de enseñanza:**

Las clases se conformarán por una serie de actividades teórico-prácticas y prácticas con ejercitación en salas de informática principalmente orientadas a la resolución de problemas.

#### **Plan de trabajo en el campus:**

El Campus Virtual es un espacio fundamental para el desarrollo de la asignatura. En el aula virtual se propondrá material educativo, apuntes de clase, bibliografía, así como también el programa y cronograma de la asignatura y las guías de Trabajos Prácticos y ejercicios. Además, habrá un foro permanente para las consultas y/o dudas de los/as alumnos/as.

### **6. Actividades de investigación y extensión (si hubiera)**

No aplica.

### **7. Evaluación y régimen de aprobación**

Consistirá en dos exámenes parciales con recuperatorios, según el cronograma previsto, de la totalidad de la materia descripta en el programa. Los mismos se realizarán en las fechas que se establezcan en el cronograma correspondiente.

#### **7.1 Aprobación de la cursada**

Para aprobar la cursada y obtener la condición de regular, el régimen académico establece que debe obtenerse una nota no inferior a cuatro (4) puntos. Todas las instancias evaluativas deberán tener una instancia de recuperatorio. Podrán acceder a la administración de esta modalidad solo aquellos y aquellas estudiantes que hayan obtenido una nota inferior o igual a 6 (seis) puntos en el examen parcial.

Siempre que se realice una evaluación de carácter recuperatorio, la calificación que los/as estudiantes obtengan reemplazará la calificación obtenida en el examen que se ha recuperado y será la considerada definitiva a los efectos de la aprobación.

El/La alumno/a deberá poseer una asistencia no inferior al 75% en las clases presenciales.

En cuanto a la cursada de manera virtual se requerirá que el/la estudiante ingrese al aula virtual como mínimo una vez por semana.

## 7.2 Aprobación de la materia

La materia puede aprobarse por promoción, evaluación integradora, examen final o libre.

**Promoción directa:** tal como lo establece el art°17 del Régimen Académico, para acceder a esta modalidad, el/la estudiante deberá aprobar la cursada de la materia con una nota no inferior a siete (7) puntos, no obteniendo en ninguna de las instancias de evaluación parcial menos de seis (6) puntos, sean evaluaciones parciales o recuperatorios. El promedio estricto resultante deberá ser una nota igual o superior a siete (7) sin mediar ningún redondeo.

**Evaluación integradora:** tal como lo establece el art°18 del Régimen Académico, podrán acceder a esta evaluación aquellos estudiantes que hayan aprobado la cursada con una nota de entre cuatro (4) y seis (6) puntos.

La evaluación integradora tendrá lugar por única vez en el primer llamado a exámenes finales posterior al término de la cursada. Deberá tener lugar en el mismo día y horario de la cursada y será administrado, preferentemente, por el/la docente a cargo de la comisión. Se aprobará tal instancia con una nota igual o superior a cuatro (4) puntos, significando la aprobación de la materia.

La nota obtenida se promediará con la nota de la cursada.

**Examen final:** Instancia destinada a quienes opten por no rendir la evaluación integradora o hayan regularizado la materia en cuatrimestres anteriores. Se evalúa la totalidad de los contenidos del programa de la materia y se aprueba con una calificación igual o superior a cuatro (4) puntos. Esta nota no se promedia con la cursada.

## 7.3 Criterios de calificación

La calificación de cada evaluación se determinará en la escala 0 a 10, con los siguientes valores: 0, 1, 2 y 3: insuficientes; 4 y 5 regular; 6 y 7 bueno; 8 y 9 distinguido; 10 sobresaliente.

## 8 Cronograma

| Clase | Materia                   | Temas   | Modalidad  |
|-------|---------------------------|---|------------|
| 1     | C++                       | Introducción. Proceso de compilación - IDEs - Nociones básicas del lenguaje   | Presencial |
| 2     |                           | Nociones básicas del lenguaje C++ - ámbitos variables - archivos - espacios de nombre   | Virtual    |
| 3     |                           | Estructura de la memoria - Punteros a datos simples - Pedido de memoria y liberación  | Virtual    |
| 4     |                           | Punteros a estructuras complejas - Doble punteros -   | Virtual    |
| 5     | Complejidad computacional | Análisis de algoritmos: Análisis asintótico, comportamiento en el mejor caso, caso promedio y peor caso. Modelo computacional. Concepto de tiempo de ejecución. Notación $O()$ , $\Omega$ , $\Theta$ . Reglas generales para el cálculo del tiempo de ejecución. Presentación de enunciado de TP. | Presencial |

|    |                           |   |            |
|----|---------------------------|---|------------|
| 6  | Complejidad computacional | Cálculo de tiempo y orden de ejecución en algoritmos iterativos y recursivos. Análisis de los métodos de ordenamiento: quick - merge - radix - tree - heap y shell sort Comparación de distintas estrategias de diseño de algoritmos. | Virtual    |
| 7  | Recursividad              | Principios de recursividad - Funcionamiento interno - Divide y vencerás   | Virtual    |
| 8  | Complejidad               | Complejidad algorítmica aplicada a algoritmos recursivos  | Virtual    |
| 9  | Árboles                   | Introducción a Árboles - Árboles binarios - ABB   | Virtual    |
| 10 | Grafos                    | Grafos: definición - implementaciones - Backtracking  | Presencial |
| 11 |                           | Programación dinámica - Floyd - Kruskal - Prim - Bellman Ford   | Virtual    |
| 12 |                           | Práctica de grafos. Análisis de tiempo de ejecución   | Virtual    |
| 13 | Árboles                   | árboles balanceados – AVL. árboles multivías - árboles B - B+ y B* Analisis de complejidad  | Virtual    |
| 14 | Dispersión                | Funciones de hashing: Tipos. Colisiones: Hash cerrado y abierto. Funciones de hashing perfectas.  | Presencial |
| 15 | Ordenamiento              | Métodos de ordenamiento: quick - merge - radix - tree - heap y shell sort. Analisis de complejidad  | Virtual    |
| 16 | Cierre                    | Defensa tp grupal   | Presencial |