

<b>Facultad de Ingeniería - Universidad Nacional de Cuyo</b>			
<b>P1- PROGRAMA DE ASIGNATURA</b>			
<b>Asignatura:</b>	<b>Algoritmos Y Estructuras de Datos I</b>		
<b>Profesor Titular:</b>	<b>Carlos Catania</b>		
<b>Carrera:</b>	<b>Licenciatura en Ciencias de la Computación.</b>		
<b>Año: 2018</b>	<b>Semestre: 2do</b>	<b>Horas Semestre: 96 hs</b>	<b>Horas Semana: 6hs</b>

### **OBJETIVOS**

Al finalizar el curso los alumnos conocerán:

1. Los conceptos necesarios para Identificación de problemas algorítmicos.
2. Los conceptos de estructuras de datos básicas
3. El procedimiento para la resolución problemas algorítmicos aplicando el pensamiento abstracto.
4. Las nociones de complejidad computacional para el diseño de algoritmos.

#### **Objetivos Específicos**

Se espera que al finalizar el curso el alumno sea capaz de:

1. Implementar algoritmos numéricos básicos
2. Implementar algoritmos de ordenamiento y búsquedas y poder explicar sus diferencias en su complejidad temporal.
3. Conocer otros factores además de la eficiencia computacional que influyen en la elección de un algoritmo. Como ser tiempo de desarrollo, mantenimiento, la inclusión de patrones específicos en los datos de entrada, etc.
4. Demostrar habilidad para evaluar algoritmos, seleccionar el adecuado en función de un rango posible de opciones, justificar esa selección e implementar el algoritmo en particular.

### **CONTENIDOS**

#### **UNIDAD 1 TIPOS DE DATOS ABSTRACTOS**

**1.A Abstracción:** Abstracción. Especificación e implementación. Tipos abstractos de datos. Sintaxis. Especificación de propiedades. Implementación como modelo. Ejemplos.

**1.B Modulación:** Modularización. Ventajas del ocultamiento de información.

#### **UNIDAD 2 ARREGLOS, ESTRUCTURAS y ARCHIVOS**

Arreglos. Usos más comunes. Estructuras básicas (records). Concepto abstracto y concretización. Archivos. Operaciones de inserción, búsqueda y borrado. Acceso secuencial y aleatorio.

#### **UNIDAD 3 COMPLEJIDAD ESPACIAL Y TEMPORAL**

Introducción al análisis de complejidad. Reglas para estimación de cantidad de operaciones. Aplicaciones.

#### **UNIDAD 4: LISTAS**

Punteros. Listas enlazadas. Lista simplemente y doblemente enlazada. Operaciones de inserción, búsqueda y borrado. Listas circulares. Análisis de complejidad. Aplicaciones. Pilas y Colas. Colas con prioridad.

#### **UNIDAD 5: ÁRBOLES**

Árboles. Operaciones de inserción, búsqueda y borrado. Recursión. Tipos de datos recursivos. Árboles binarios de búsqueda. Definición. Propiedades. Ventajas y limitaciones. Representación de un árbol mediante una lista: Heaps. Aplicaciones. Análisis de complejidad.

#### **UNIDAD 6: ALGORITMOS Y ORDENACIÓN**

##### **6.A Algoritmos básicos de ordenamiento**

Métodos básicos de ordenación. Algoritmo de la burbuja. Ordenación por inserción. Ordenación por selección. Comparación de los métodos básicos.

### 6.B Algoritmos avanzados

Técnica divide & conquer. Mergesort y Quicksort. Estructuras intermedias en la resolución de algoritmos. HeapSort. Tipos de algoritmos. Greedy y Backtracking

## UNIDAD 7: VERIFICACIÓN FORMAL DE ALGORITMOS

7.A Introducción. Conceptos preliminares. Concepto de corrección. Reglas generales sobre precondiciones y postcondiciones.

7.B. Invariantes. Definición. Preservación de invariantes en las operaciones.

Demostración de corrección de un algoritmo usando invariantes. Verificación de códigos sin bucles. Bucles. Terminación del programa.

## METODOLOGÍA DE ENSEÑANZA

Para el dictado de la materia se propone aplicar una metodología basada en el modelo pedagógico conocido como “**aula invertida**”. Se trata de un modelo que plantea la necesidad de transferir parte del proceso de enseñanza y aprendizaje fuera del aula con el fin de utilizar el tiempo de clase para el desarrollo de procesos cognitivos de mayor complejidad, que favorezcan el aprendizaje significativo.

La enseñanza de algoritmos y estructura de datos resulta adecuada para aplicar una metodología de aula invertida, ya que se trata de una materia con una fuerte componente práctica.

Se presenta a continuación el detalle de las actividades deberán realizar los alumnos durante el cursado de la misma.

### Actividades extra-aula

1. Previo a cada clase, el alumno deberá tener una primera aproximación a los conceptos a desarrollar en dicha clase mediante material fundamentalmente audiovisual, junto a otro material de consulta debidamente preparado por la cátedra.
2. Al finalizar de consumir el material, el alumno deberá responder una serie de **preguntas guías** destinadas a evaluar el grado de comprensión sobre cada tema tratado. Dichas preguntas guías apuntan no sólo a detectar problemas de comprensión por parte del alumno, sino que también serán indicadores de mejoras futuras en el material provisto al alumno.

### Actividades intra-aula

1. Entre el 25 y 30 por ciento del tiempo de la clase se utilizará para reforzar los conocimientos teóricos y solucionar los posibles inconvenientes que hayan surgido durante las actividades extra-aula.
2. Finalizado el proceso de la discusión y refuerzo de conocimientos, se destinará el resto de la clase al desarrollo de diferentes actividades orientadas a aplicar y afianzar los conocimientos adquiridos durante las etapas previas. Estas actividades pueden incluir:
  1. **Trabajo dirigido (TD):** se provee al alumno de un conjunto de ejercicios simples que deberán ser resueltos aplicando mayormente los puntos expuestos durante la clase. Los TD pueden tomar la forma de cuestionarios de opción múltiple, o también permitir desarrollo de los diferentes algoritmos expuestos.
  2. **Trabajo Práctico de laboratorio (TP):** se trata de problemas que deben ser resueltos en computadora. El objetivo de esta actividad acercar al alumno a implementación concretas de los algoritmos a partir de los algoritmos en pseudocódigo y las estructuras abstractas de datos.
  3. **Actividad de investigación (AI):** se trata de una actividad que apunta a guiar al alumno en el desarrollo de soluciones sobre temas no desarrollados con anterioridad. Para llegar a la solución, el alumno requerirá la aplicación de varios de los conocimientos expuestos del programa.
  4. **Evaluación:** Se trata de una evaluación integradora realizada al final de cada unidad. La misma se realiza de manera individual y tiene por objetivo medir el nivel de conocimiento de los alumnos a fin de tomar las acciones correctivas necesarias.
3. Antes de dar por finalizado el tema desarrollado, se procederá a realizar una puesta en común en donde tanto los alumnos como los profesores de la cátedra elevarán sus opiniones sobre los temas tratados.

La siguiente tabla presenta la carga horaria de las actividades realizadas dentro del aula. Se puede observar que tal cual se indicó al comienzo de la sección las actividades relacionadas con la teoría están en el orden del 25%, mientras que la práctica alcanza el restante 75%.

<b>Actividad</b>	<b>Carga horaria por semestre</b>
Teoría y resolución de ejercicios simples	24
<b>Formación práctica</b>	
Formación Experimental – Laboratorio	27
Formación Experimental - Trabajo Dirigido	27
Actividad de Investigación	18
<b>Total</b>	<b>96</b>

### **BIBLIOGRAFÍA**

#### **Bibliografía básica**

Autor	Título	Editorial	Año	Ejemplares en biblioteca
Brassard, G.; Bratley, P.	Fundamentos de algoritmia	Prentice-Hall	1998	
Weiss, M. A	Estructuras de datos y algoritmos.	Addison- Wesley	1995	
Aho, A. V.; Hopcroft, J. E.; Ullman, J. D.	Estructura de datos y algoritmos.	Addison-Wesley	2001	

#### **Bibliografía complementaria**

Autor	Título	Editorial	Año	Ejemplares en biblioteca
T. H. Cormen, C. E. Leiserson, R L. Rivest, C. Stein	Introduction to Algorithms, 3º ed.	The MIT Press	2010	
Martí, M.; Ortega, Y. Verdejo	A. Estructuras de datos y métodos algorítmicos: ejercicios resueltos	Prentice-Hall	2004	