

**TALLER DE TECNOLOGÍAS DE  
PRODUCCIÓN DE SOFTWARE  
(Opción C - Técnicas y Estrategias  
para la resolución de problemas)**

**Año 2025**

**Carrera/ Plan:**

*Analista Programador Universitario Plan 2021/Plan 2015*

**Año:** 3°

**Régimen de Cursada:** Semestral

**Carácter (Obligatoria/Optativa):** Obligatoria

**Correlativas:** *Diseño de Bases de Datos -*

*Algoritmos y Estructuras de Datos –*

*Introducción a los Sistemas Operativos -*

*Orientación a Objetos 1 -*

*Taller de lecto comprensión y traducción en  
Inglés*

*Ingeniería de Software 2*

**Profesor/es:** *Alejandra Schiavoni*

**Hs. semanales:** *6 hs. (clases teórico/prácticas)*

**FUNDAMENTACIÓN**

Dentro del marco de la carrera, esta opción propone ampliar los conocimientos con los que cuentan los alumnos, para ayudarlos a perfeccionar sus habilidades en la resolución de problemas complejos, que actualmente se presentan con mayor frecuencia en el desempeño de la actividad de un profesional informático. En su contenido se incluyen técnicas de diseño de algoritmos proveyendo herramientas y metodologías, que les permiten resolver una amplia gama de problemas e implementar algoritmos eficientes aplicados a problemas de gran escala. Se tienen en cuenta todas las etapas que van desde la lectura e interpretación del enunciado del problema, la abstracción del problema en sí mismo, la búsqueda de una solución, su implementación, y la verificación de su viabilidad en cuanto a correctitud y requerimientos de tiempo y/o espacio teórica y experimentalmente.

**OBJETIVOS GENERALES**

Introducir a los alumnos en un esquema de organización de producción de software, utilizando metodologías, prácticas y herramientas actualizadas y acordes con los estándares actuales.

Fomentar la práctica del alumno en esquemas de trabajo similares a los que se utilizan en las empresas de desarrollo de productos de software.

Ofrecer a los alumnos alternativas tecnológicas, siempre en base a herramientas de utilización actual en el mercado laboral.

**CONTENIDOS MÍNIMOS (de acuerdo al Plan de Estudios)**

- Introducir un ambiente de desarrollo de software estandarizado (con herramientas integradas que den una visión homogénea y estandarizada de las aplicaciones, su interfaz gráfica, el acceso a las bases de datos y la interconexión entre aplicaciones), enfocado a un organismo o “clase” de empresa usuaria.
- Practicar cómo usar el ambiente de desarrollo y las diferencias que tiene con el ambiente de producción, ilustrando la metodología organizacional del pasaje de desarrollo a producción.
- Practicar con documentación estandarizada (por ej. Casos de Uso con UML) mostrando cómo se pasa de una especificación al código ejecutable.
- Ejemplificar la actividad del tester de aplicaciones. Metodología de trabajo y ambiente de prueba (diferencia con los otros ambientes)

- Plantear el proceso estandarizado de desarrollo de software en una tecnología de uso en el mercado. Rol de la documentación en cada etapa.
- Plantear el desarrollo de una solución a un problema real y que ilustre todas las problemáticas antes descriptas
- Describir cuales son las principales características de un proceso de desarrollo de software con calidad (introduciendo los principios básicos de CMM o CMMI)

## **PROGRAMA ANALÍTICO**

El contenido del curso está basado en los conceptos teóricos y herramientas para el análisis y la resolución de diferentes clases de problemas y situaciones. Además, se repasarán las estructuras de datos, los algoritmos vistos anteriormente en la carrera que se aplican en la solución de problemas concretos y conceptos de los lenguajes que se utilizarán en la implementación de las soluciones.

Los temas apuntan a resolver situaciones tales como:

- 1.- Búsqueda de caminos particulares dentro de un grafo que cumplan con ciertas condiciones
  - Análisis de conectividad y procesamiento de redes de flujos
  - Camino de Euler y camino Hamiltoniano
  - Búsqueda de caminos mínimos con algunas variantes
- 3.- Manejo de conjuntos disjuntos
  - Uso de la estructura Union-Find: particionamiento de conjuntos, componentes conexas, árbol abarcador mínimo
- 2.- Cálculo y uso de números primos y manipulación de números muy grandes
  - Divisibilidad (máximo común divisor, mínimo común múltiplo), aritmética modular, congruencias. Teorema chino del resto. Criba de Eratóstenes
  - Cálculos de Combinatoria
  - Coeficientes binomiales. Teorema de Newton.
  - Bases numéricas
- 3.- Asignación de recursos
  - Procesamiento sobre strings: búsqueda de patrones, subsecuencia común más larga, similitud de strings (editdistance)
- 4.-Procesamiento de figuras geométricas: distancias, intersección, inclusión.
  - Envoltura convexa (Convex Hull)
  - Algoritmos de triangulación
  - Visualización y reconocimiento de patrones

## **BIBLIOGRAFÍA**

- *Competitive Programming 3*. Steven Halim, Felix Halim. Standard Copyright License, Third Edition, ISBN: 9781312796645.
- *Programming Challenges. The Programming Contest Training Manual*. Steven S. Skiena, Miguel A. Revilla. Springer-Verlag New York, Inc., 2003 ISBN 0-387-00163-8.
- *Programming Pearls*. Second Edition. Jon Bentley. Addison-Wesley, Inc., 2000. ISBN 0-201-65788-0.
- Conjuntos de Problemas de las Regionales y Finales de las ACM IPCP International Collegiate Programming Contest., <http://cm.baylor.edu/ICPCWiki/Wiki.jsp?page=Problems>
- *Data Structures and Algorithm Analysis in Java*. Mark Allen Weiss. Addison Wesley; 2nd edition (March 3, 2006). ISBN : 0-321-37013-9
- Jueces on-line
  - Juez Programming Challenges: <http://www.programming-challenges.com>
  - Juez de la Universidad de Valladolid: <http://uva.onlinejudge.org/>
  - Sphere Online Judge: <http://www.spoj.com/>

- 
- ACM-ICPC Live Archive: <https://icpcarchive.ecs.baylor.edu/>
  - Codeforces: <http://codeforces.com/>
  - Light Oj: <http://www.lightoj.com/index.php>

## **BIBLIOGRAFÍA COMPLEMENTARIA**

- *Introduction to algorithm*; third edition. Thomas H. Cormen, The MIT Press. 2009
- *Algorithms*. S. Dasgupta, C. H. Papadimitriou, and U. V. Vazirani. McGraw-Hill Science/Engineering/Math; 1 edition (September 13, 2006). ISBN-13: 978-0073523408
- *Competitive Programming 3, The New Lower Bound of Programming Contests*. S. Halim, F. Halim, Lulu.com, 2013. ISBN: 8392212355, 9788392212355

## **METODOLOGÍA DE ENSEÑANZA**

Las clases son dos veces por semana, en día y horario a confirmar y se desarrollan en un aula equipada con computadoras. En la carga semanal se incluye el desarrollo de los temas teóricos, y una explicación de las técnicas y estrategias a utilizar en la resolución de los diferentes tipos de problemas relacionados. Se incluye además la explicación de un ejercicio sobre el tema desarrollado. Se podrán hacer virtual un porcentaje menor de algunas actividades, si es pertinente con la temática abordada.

Respecto a la actividad de los alumnos, la modalidad de trabajo consistirá en plantear problemas de dificultad gradual sobre los temas abordados en el curso, para que los alumnos resuelvan. El trabajo se realizará en forma individual o en grupos de hasta 3 integrantes, Para la implementación se usa un ambiente de desarrollo que permite la utilización de todas sus herramientas integradas destinadas a facilitar la escritura, prueba y puesta a punto de los algoritmos.

Durante la cursada se usarán jueces electrónicos, donde los alumnos deben enviar las soluciones a los problemas para que sean evaluadas.

En las clases, se distribuirán ejercicios a los grupos. Cada grupo resuelve el ejercicio asignado previamente y define casos de prueba para alguno de los ejercicios resueltos por los otros grupos.

Cada grupo, además, explicará a los restantes la solución encontrada, exponiendo los inconvenientes que fueron hallando en la resolución. Se discutirán otras posibles soluciones. Se realizará un testeo de las distintas soluciones con diferentes casos de prueba.

Esta manera de trabajo alienta la interacción entre los grupos y el desarrollo de una tarea participativa.

El desarrollo de los temas teóricos y la explicación de ejercicios ejemplificadores serán a través de presentaciones electrónicas. Además, se utiliza la pizarra para realizar una explicación más detallada en caso de ser necesario.

Se utiliza también Moodle (plataforma de aprendizaje virtual), que ofrece una funcionalidad muy útil para la organización del curso. A través de ella, se publican las explicaciones teóricas, los ejercicios a resolver, material adicional de consulta y la bibliografía. Además, se usan los *Foros* para realizar consultas, anuncios, discusiones, etc, las *Wikis* para que los alumnos trabajen en grupo, las *Tareas* para que los alumnos realicen entregas, las *Encuestas* cuando es necesario realizar una consulta a los alumnos. También cuenta con la herramienta de video conferencia Big Blue Button, para la realización de actividades virtuales.

## **EVALUACIÓN**

Con respecto a la evaluación de los resultados alcanzados por los alumnos se propone la siguiente modalidad para acreditar la aprobación de la materia:

Cada alumno deberá cumplir con el 80% de asistencia a las clases.

Durante la cursada, los alumnos deben resolver los ejercicios propuestos por la cátedra, según lo explicado en la Modalidad de Enseñanza presentada en el ítem anterior. Cada alumno deberá implementar las soluciones y enviarlas al juez electrónico correspondiente, que se utiliza para determinar la correctitud de las mismas. Cada alumno debe resolver correctamente al menos un problema propuesto por cada tema.

Antes de finalizar el dictado de la materia los alumnos deben realizar una exposición sobre la solución obtenida de alguno de los problemas resueltos durante la cursada.

Para la aprobación de la cursada, además deben resolver un problema propuesto por la cátedra y exponer su resolución dentro del contexto del tema en el que se encuadra.

Para la aprobación de la materia habrá un trabajo final que consistirá en una investigación de las características y distintas alternativas de solución de un problema modelo que es aplicado a distintas situaciones.

**CRONOGRAMA DE CLASES Y EVALUACIONES**

Clase	Fecha	Contenidos/Actividades
Semana 1	25/08/2025	Clase de presentación. Explicación del funcionamiento de los Jueces on-line -Registración- Prueba de envío soluciones. Chequeo de errores. Manejo de datos y operaciones de entrada/salida Ejemplo de problemas sencillos: <i>Jolly Jumper</i> (Saltos), <i>Hartals</i> y <i>ACM ContestScoring</i>  Conceptos de lenguaje y manejo de estructuras de datos en C++, Python y Java
Semana 2	01/09/2025	Programación Dinámica: introducción. Explicación de los casos: LongestInc/DecreasingSubsequence (LIS) Range Sum Query (RSQ)  Resolución por parte de los alumnos de los ejercicios de Programación Dinámica LIS/LDS y RSQ
Semana 3	08/09/2025	Programación Dinámica Explicación de los casos: Knapsack Longest Common Subsequence (LCS) y Edit Distance  Resolución por parte de los alumnos de los ejercicios de Programación Dinámica Knapsack, LCS y EditDistance
Semana 4	15/09/2025	UFDS: UnionFindDisjoint Set Grafos: recorridos y aplicaciones. Componentes conexas. Puntos de articulación. Puentes. Camino de Euler. Camino Hamiltoniano.
Semana 5	22/09/2025	Resolución por parte de los alumnos de ejercicios de los distintos casos de Programación Dinámica.  Resolución por parte de los alumnos de los ejercicios de recorridos de grafos
Semana 6	29/09/2025	Grafos: caminos mínimos. Algoritmo de Dijkstra. Algoritmo para grafos sin peso. Algoritmo de Bellman-Ford.  Árbol de expansión mínima (MST). Casos: MiniMax/MaxiMin, Segundo mejor árbol de expansión  Resolución por parte de los alumnos de los ejercicios de caminos mínimos y MST de grafos
Semana 7	06/10/2025	Geometría Analítica: conceptos. Caso de estudio: Plan de vuelo de Superman (ProgrammingChallenges)  Ecuación de una recta, pendiente. Punto más cercano

Semana 8	13/10/2025	Geometría Computacional: Definiciones. Triangulación. Producto cruzado. Polígonos Implementación de Convex Hull. Puntos colineales.
Semana 9	20/10/2025	Resolución por parte de los alumnos de los ejercicios de geometría computacional
Semana 10	27/10/2025	Teoría de números: aplicaciones. Números primos. Criba de Eratóstenes. Divisibilidad. Mínimo común múltiplo. Máximo común divisor.
Semana 11	03/11/2025	Teoría de aritmética modular
Semana 12	10/11/2025	Resolución por parte de los alumnos de los ejercicios sobre teoría de números.
Semana 13	17/11/2025	Resolución y análisis de ejercicios propuestos en las distintas ediciones del Torneo Argentino de Programación
Semana 14	24/11/2025	Actividad de los alumnos  Exposición de la resolución de un problema por parte de los alumnos.
Semana 15	01/12/2025	Clases de consulta.
Semana 16	08/12/2025	Evaluación: exposición del problema asignado y coloquio sobre los temas vistos en la materia
Semana 17	15/12/2025	Evaluación: exposición del problema asignado y coloquio sobre los temas vistos en la materia

Evaluaciones previstas	Fecha
Prueba diagnóstica	Semana del 25/08/2025
Entrega de las resoluciones de los problemas aceptados sobre Programación Dinámica	Semana del 15/09/2025
Entrega de las resoluciones de los problemas aceptados sobre Grafos	Semana del 06/10/2025
Entrega de las resoluciones de los problemas aceptados sobre Geometría	Semana del 27/10/2025
Entrega de las resoluciones de los problemas aceptados sobre Teoría de Números	Semana del 17/11/2025
Evaluación: exposición del problema asignado y coloquio sobre los temas vistos en la materia	Semana del 01/12/2025



Evaluación: exposición del problema asignado y coloquio sobre los temas vistos en la materia	Semana del 08/12/2025
Evaluación: exposición del problema asignado y coloquio sobre los temas vistos en la materia	Semana del 15/12/2025

**Contacto de la cátedra (mail, sitio WEB, plataforma virtual de gestión de cursos):**

Plataforma virtual: <https://catedras.info.unlp.edu.ar/>

María Alejandra Schiavoni  
Prof. Titular