



UNIVERSIDAD NACIONAL DE LA PLATA
FACULTAD DE INFORMÁTICA

**TALLER DE TECNOLOGÍAS DE
PRODUCCIÓN DE SOFTWARE
(Opción C)**

Carrera: *Analista Programador
Universitario
Plan 2007/2015*

Año: 3°

Carácter: Obligatoria

Régimen de Cursada: Semestral

Correlativas:

Introducción a las 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

Año 2018

Profesor: *Alejandra Schiavoni*

Hs. semanales: 6 hs.

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 MINIMOS



- Introducir un ambiente de desarrollo de software estandarizado (con herramientas integradas que den una visión homogénea y estandarizada de las aplicaciones, su interfaz grafica, 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 (edit distance)

4.-Procesamiento de figuras geométricas: distancias, intersección, inclusión.

- Envoltura convexa (Convex Hull)
- Algoritmos de triangulación
- Visualización y reconocimiento de patrones

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.

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 e implementen en clase y en su casa. El trabajo se realizará en forma individual o en grupos de hasta 3 integrantes, en las computadoras del aula, dado que los alumnos deben implementar las soluciones a los problemas usando un lenguaje de programación. 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.

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 rendir un parcial que consta de varios problemas, que abarcan los temas dados y de los cuales deben resolver correctamente uno de cada tema.

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.

BIBLIOGRAFÍA OBLIGATORIA

- *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 ICPC 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/>



- Caribbean Online Judge: <http://coj.uci.cu>
- 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

CRONOGRAMA DE CLASES Y EVALUACIONES

Clase	Contenidos/Actividades	Evaluaciones previstas
1 Semana 20/08/2018	Clase de presentación. Explicación del funcionamiento de los Jueces on-line - Registración- Prueba de envío soluciones. Chequeo de errores. Ejemplo de problemas sencillos: <i>Jolly Jumper</i> (Saltos), <i>Hartals</i> y <i>Way too long words</i>	
2		Prueba Diagnóstica
3	1era Parte Conceptos de Lenguaje C++ y Java. Estructuras de Datos básicas. Programación Dinámica: introducción y explicación de problemas de tipo Longest Inc/Decreasing Subsequence. Range Sum Query (RSQ) 2da Parte - Actividad de los alumnos Resolución por parte de los alumnos de los ejercicios de Programación Dinámica (LIS/LDS), RSQ	
4	Programación Dinámica: explicación de problemas de tipo Longest Common Subsequence (LCS) y Edit Distance	
5	Actividad de los alumnos Resolución de ejercicios de los distintos casos de Programación Dinámica vistos en clase	



UNIVERSIDAD NACIONAL DE LA PLATA
FACULTAD DE INFORMÁTICA

6	Explicación de <i>0-1 Knapsack Problem</i> . Problema del cambio	
7	Geometría Analítica: conceptos. Caso de estudio: Plan de vuelo de Superman (Programming Challenges) Ecuación de una recta, pendiente. Punto más cercano	
7	Actividad de los alumnos Resolución de los problemas propuestos sobre los temas dados.	
8		Entrega de las resoluciones de los problemas aceptados sobre Programación Dinámica
9	Geometría Computacional: Definiciones. Triangulación. Producto cruzado. Polígonos Implementación de Convex Hull. Puntos colineales.	
10	1era Parte Teoría de Números: Divisibilidad. Números Primos: Criba de Eratóstenes 2da parte - Actividad de los alumnos Resolución de los problemas propuestos sobre los temas dados.	
11	Actividad de los alumnos Resolución de los problemas propuestos sobre los temas dados.	
12		Entrega de las resoluciones de los problemas aceptados sobre Geometría
13	Teoría de Números: Bases Numéricas. Combinatoria: coeficiente binomial y Teorema de Newton.	
14	1era Parte Teoría de Números: Aritmética Modular. Test de divisibilidad. Congruencias: Teorema Chino del Resto. 2da Parte - Actividad de los alumnos Resolución de los problemas propuestos sobre los temas	



UNIVERSIDAD NACIONAL DE LA PLATA
FACULTAD DE INFORMÁTICA

	datos.	
15	Actividad de los alumnos Resolución de los problemas propuestos sobre los temas dados.	
16	Estructura de Datos Union Find Disjoint Set: definición, representación y operaciones. Ejemplos	
17	Actividad de los alumnos Resolución de los problemas propuestos sobre el tema.	
18	Grafos: Punto de Articulación. Puentes. Camino de Euler. Camino Hamiltoniano.	
19		Entrega de las resoluciones de los problemas aceptados sobre Teoría de Números
20	Grafos: Caminos Mínimos. Algoritmo de Dijkstra. Algoritmo para grafos sin peso. Algoritmo de Bellman-Ford. Árbol abarcador mínimo.	
21	Actividad de los alumnos Resolución de los problemas propuestos sobre los temas dados.	
22	1era Parte Explicación de ejercicios de grafos de resolución no convencional: <i>Almost Shortest Path, Hedge Mazes</i> 2da parte - Actividad de los alumnos Resolución de los problemas propuestos sobre los temas dados.	
23	Resolución y análisis de ejercicios propuestos en las distintas ediciones del Torneo Argentino de Programación	
24		Entrega de las resoluciones de los problemas aceptados sobre Grafos
25		Exposición de la resolución de un problema por parte de los alumnos
26	Consultas	Exposición de la resolución de un problema por parte de



**UNIVERSIDAD NACIONAL DE LA PLATA
FACULTAD DE INFORMÁTICA**

		los alumnos
27		Parcial – 1era Fecha Semana del 19/11/2018
28	Resolución de los problemas del parcial. Consultas	
29	Consultas.	
30		Parcial – 2da Fecha Semana del 03/12/2018
31	Resolución de los problemas del parcial. Consultas	
32	Consultas.	
33		Parcial – 3ra Fecha Semana del 17/12/2018

Contacto de la cátedra (mail, página, plataforma virtual de gestión de cursos):

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

Firmas del/los profesores responsables: