



UNIVERSIDAD NACIONAL DE LA PLATA
FACULTAD DE INFORMÁTICA

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

Año 2012

Carrera: **Analista Programador
Universitario**

Año: **3°**

Carácter: **Optativa**

Régimen de Cursada: **Semestral**

Correlativas:

Profesor:

Alejandra Schiavoni

Catalina Mostaccio

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 leer y entender el enunciado, abstraer el problema, encontrar una solución, implementarla, y corroborar 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
- 2.- Cálculo y uso de números primos
 - Divisibilidad (máximo común divisor, mínimo común múltiplo), aritmética modular, congruencias. Teorema chino del resto. Criba de Eratóstenes
 - Cálculos de Combinatoria
- 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 de tres horas cada una, dos veces por semana, en día y horario a confirmar.

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 de la sala de PC o de alguna otra aula equipada de la Facultad, 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.

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:



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

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 grupo de trabajo deberá implementar las soluciones y enviarlas a un Jurado electrónico, que se utiliza para determinar la correctitud de las mismas. Cada grupo debe resolver correctamente el 80% de los problemas propuestos y cada integrante del grupo debe exponer la resolución de un problema.

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

- *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/>

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	Clase de presentación. Jueces-Registración-Envío soluciones- Errores. Ejemplo de Problemas sencillos: Jolly Jumper (Saltos) y WERTYU	
2		Prueba Diagnóstica
3	1era Parte Actividad de los alumnos. Resolución de ejercicios de grafos repasando algoritmos conocidos, para familiarizarse con el ambiente de trabajo. Problemas: Playing_with_wheels, Degrees_of_Separation, Tower_of_Cubes, Fire_Station, Lazy_Jumping_Frog Explicación de alguno de los problemas planteados, si fuera necesario 2da Parte Programación Dinámica: introducción y explicación de problemas de tipo Longest Inc/Decreasing Subsequence	
4	1era Parte - Actividad de los alumnos Resolución por parte de los alumnos de los ejercicios de Programación Dinámica (LIS/LDS) Problemas: Coin change Is Bigger Smarter 2da Parte Conceptos de Lenguaje C++ y Java	



UNIVERSIDAD NACIONAL DE LA PLATA
FACULTAD DE INFORMÁTICA

5	1era Parte Explicación de un ejercicio de grafos de resolución no convencional: Fire Station 2da Parte Programación Dinámica : continuación con problemas de tipo Longest Common Subsequence (LCS)	
6	1era Parte Explicación de un ejercicio de grafos de resolución no convencional: Almost Shortest Path 2da parte - Actividad de los alumnos Resolución de los problemas propuestos sobre los temas dados.	
7	1era Parte Teoría de Números: Divisibilidad. Números Primos: Criba de Eratóstenes 2da Parte Programación Dinámica: continuación con problemas de tipo Edit Distance	
8	Actividad de los alumnos Resolución de los problemas propuestos sobre los temas dados.	Entrega de las resoluciones de los problemas aceptados
9	1era Parte Geometría Analítica: conceptos. Ejemplo: Plan de vuelo de Superman 2da Parte Actividad de los alumnos Resolución de los problemas propuestos sobre los temas dados.	
10	Actividad de los alumnos Resolución de los problemas propuestos sobre los temas dados.	
	1era Parte Teoría de Números: Aritmética Modular. Test de divisibilidad.	



UNIVERSIDAD NACIONAL DE LA PLATA
FACULTAD DE INFORMÁTICA

11	Congruencias: Teorema Chino del Resto. 2da Parte Programación Dinámica: continuación con problemas de tipo: 0-1 Knapsack Problem	
12	Actividad de los alumnos Resolución de los problemas propuestos sobre los temas dados.	Entrega de las resoluciones de los problemas aceptados
13	Se explican los problemas: Light more light (Teoría de Números) The knights of the round table (Geometría Análítica) History Grading (Programación Dinámica)	
14	1era Parte Grafo: Algoritmo de búsqueda A* Ejemplo: Resolución Problema del 8-Puzzle 2da Parte Actividad de los alumnos Resolución de los problemas propuestos sobre los temas dados.	
15	1era Parte Teoría de Números: Bases Numéricas. Combinatoria: coeficiente binomial y Teorema de Newton. 2da Parte Actividad de los alumnos Resolución de los problemas propuestos sobre los temas dados.	
16	Problema del Ancestro Común más Cercano. Ejemplo: Resolución Problema Ants Colony.	
17	1era Parte Geometría Computacional: Definiciones e implementación de Convex Hull 2da Parte	



UNIVERSIDAD NACIONAL DE LA PLATA
FACULTAD DE INFORMÁTICA

	Programación Dinámica: Resolución Problema Elevator Optimization.	
18	1era Parte Grafo: Punto de Articulación. 2da Parte Actividad de los alumnos Resolución de los problemas propuestos sobre los temas dados.	Entrega de las resoluciones de los problemas aceptados
19		Exposición de la resolución de un problema por parte de los alumnos
20	Consultas	Exposición de la resolución de un problema por parte de los alumnos
21	Consultas	
22		Parcial – 1era Fecha
23	Resolución de los problemas del parcial Consultas	
24		Parcial – 2da Fecha
25	Resolución de los problemas del parcial Consultas	
26		Parcial – 3ra Fecha
27	Resolución de los problemas del parcial	

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: