



**TALLER DE TECNOLOGÍAS DE  
PRODUCCIÓN DE SOFTWARE**  
*Opción B*

**Año 2014**

**Carrera/ Plan:**

*Analista Programador Universitario  
Plan 2007*

**Año:** 3°

**Régimen de Cursada:** *Semestral*

**Carácter:** Obligatoria

**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*

**Profesor:** *Mg. Pablo Thomas*

**Hs. semanales:** *6 hs.*

**FUNDAMENTACIÓN**

El desarrollo de software como solución a un problema simulado, contribuye a la formación del alumno y lo estimula a resolver problemas. No obstante, el desarrollo de software en entornos reales es sustancialmente diferente, particularmente en la definición del problema a resolver y en todas las circunstancias e imprevistos que en una simulación no surgen.

Por ende, esta asignatura y particularmente esta opción, ofrece al alumno la posibilidad de realizar una práctica real, es decir, resolver un problema real con las complicaciones adicionales que aparecen en este entorno, todo esto con la asistencia de los docentes de la asignatura. De este modo se articula la salida laboral del alumno al mercado.

**OBJETIVOS GENERALES:**

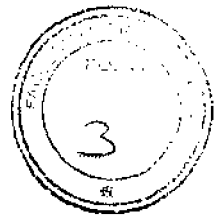
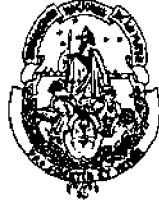
Introducir a los alumnos en un esquema de organización de producción de software, utilizando metodologías, técnicas y herramientas actualizadas, acordes con las prácticas actuales.

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.

Es propósito esencial ejecutar un proyecto real, donde se cumplan todas las etapas del ciclo de vida de desarrollo de Software, y se transfiera un Sistema de Software concreto a una Organización.

**CONTENIDOS MINIMOS:**



UNIVERSIDAD NACIONAL DE LA PLATA  
FACULTAD DE INFORMÁTICA

- Ambientes 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.
- Ambiente de desarrollo y las diferencias que tiene con el ambiente de producción. Metodología organizacional del pasaje de desarrollo a producción.
- Documentación estandarizada en el ciclo de vida de desarrollo de software.
- Actividades del tester de aplicaciones. Metodología de trabajo y ambiente de prueba (*diferencia con los otros ambientes*)
- Proceso estandarizado de desarrollo de software en una tecnología de uso en el mercado. Rol de la documentación en cada etapa.
- Desarrollo de una solución a un problema real que ilustre todas las problemáticas antes descriptas.
- Principales características de un proceso de desarrollo de software con calidad.

### PROGRAMA ANALÍTICO

1. Software a Medida vs. Software enlatado. Análisis comparativo. Ventajas y Desventajas.
2. Modelos de Ciclos de Vida de Desarrollo de Software. Aplicación de cada modelo a casos reales. Análisis comparativo.
3. Ingeniería de Requerimientos de Software. Proceso para la Ingeniería de Requerimientos. Técnicas utilizadas. Aplicación de herramientas CASE para la Gestión de Requerimientos. Especificación de Requerimientos según estándares actuales.
4. Diseño de Software. Construcción de Modelo de Datos. Técnicas utilizadas. Aplicación de herramientas CASE para especificaciones de diseño. Prototipación con Mockups.
5. Desarrollo. Criterios metodológicos. Uso de versionado. Uso de Frameworks de desarrollo. Ventajas.
6. Testeo. Ambientes de Testeo. Casos de Testeo.
7. Puesta en producción. Análisis de posibles incidencias.
8. Mantenimiento correctivo y perfectivo.



## METODOLOGÍA DE ENSEÑANZA

El dictado de la asignatura tiene modalidad de Taller lo que permite a los alumnos aplicar las estrategias propuestas en la resolución de problemas concretos a medida que se desarrollan las clases teórico/prácticas.

Las clases son guiadas a través de la proyección de transparencias utilizando el cañón y la PC, disponibles en el aula.

Los alumnos se organizarán en grupos.

Se plantearán casos prácticos **reales** a resolver. Por ende, cada grupo desarrollará un sistema de software en un entorno real, como solución a un caso real. El desarrollo del sistema implica afrontar los problemas de todo el ciclo de vida de desarrollo de software, inclusive la puesta en producción del sistema.

Cada grupo será coordinado y supervisado por los docentes de la asignatura.

Todo el material del curso estará disponible a través de la plataforma de educación a distancia WebUNLP. Tanto alumnos como docentes deberán contar con un usuario y una clave para poder acceder.

## EVALUACIÓN

La asignatura establece un régimen el cual es detallado a continuación.

Se deben realizar entregas parciales del proyecto en desarrollo (Especificación de Requerimientos, Modelo de Datos, Diseño, Implementación Final, entre otras). Las entregas son evaluadas por los docentes de la asignatura.

Para aprobar la cursada, el alumno debe haber aprobado todas las entregas solicitadas.

Para aprobar la asignatura el alumno deberá inscribirse en una mesa de final, donde expondrá a través de un coloquio el trabajo realizado para resolver el problema real planteado.

## BIBLIOGRAFÍA OBLIGATORIA

- I. Sommerville, Ingeniería de Software. Pearson-Addison Wesley.
- P. Loucopoulos, V. Karakostas; System Requirements Engineering; McGraw-Hill International series in Software Engineering

## BIBLIOGRAFÍA COMPLEMENTARIA

- Desarrollo web ágil con Symfony2, Javier Eguiluz, versión electrónica.
- IEEE, *IEEE Recommended Practice for Software Requirements Specifications*, IEEE Std 830-1998 (Revision of IEEE Std 830-1993)



- Building application frameworks: object-oriented foundations of framework design. Mohamed Fayad, Douglas Schmidt, Ralph Johnson. John Wiley & sons. Editorial Wiley. 1999
- Project Management Institute, A Guide to the Project Management Body of Knowledge, PMI, Upper Darby, 1996.
- S.L. Pfleeger, Software Engineering: Theory and Practice, second ed., Prentice Hall, 2001.
- Introducción a las Bases de Datos. Fundamentos y Diseño. Bertone, Thomas. Pearson Education 2011.
- Elmasri, Navathe. Fundamentos de Sistemas de Bases de Datos. Pearson-Addison Wesley. 2002.

### **CLASES**

Las clases se desarrollarán durante el segundo semestre. El inicio está previsto en la semana del 11 de Agosto del 2014.

Se estima dictar el contenido del programa analítico previamente detallado en 13 clases.

### **CRONOGRAMA TENTATIVO DE CLASES Y EVALUACIONES**

#### **Clase 1**

- Presentación de la asignatura
- Presentación de la forma de trabajo
- Software a Medida vs. Software enlatado. Análisis comparativo. Ventajas y Desventajas.

#### **Clase 2**

- Modelos de Ciclos de Vida de Desarrollo de Software. Aplicación de cada modelo a casos prácticos. Análisis comparativo.
- Presentación de Proyectos

#### **Clase 3**

- Introducción a la Ingeniería de Requerimientos
- Entrega de planificación por proyectos

#### **Clase 4**

- Procesos de Ingeniería de Requerimientos

#### **Clase 5**

- Presentación de subsistema de seguridad
- Diseño de Software. Construcción de Modelo de Datos. Técnicas utilizadas. Aplicación de herramientas CASE para especificaciones de diseño. Prototipación con Mockups.

#### **Clase 6**

- Desarrollo. Criterios metodológicos. Uso de versionado. Uso de Frameworks de desarrollo. Ventajas.



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

---

- Presentación de framework específico
- Clase 7**
  - Estudio de framework específico
- Clase 8**
  - Estudio de framework específico
  - Seguimiento de proyectos
- Clase 9**
  - Estudio de framework específico
  - Seguimiento de proyectos
- Clase 10**
  - Testeo. Ambientes de Testeo. Casos de Testeo.
- Clase 11**
  - Puesta en producción. Análisis de posibles incidencias.
- Clase 12**
  - Mantenimiento correctivo y perfectivo.
- Clase 13**
  - Conclusiones finales. Discusiones sobre experiencias en los proyectos realizados.

Dada la modalidad de Taller de la asignatura las evaluaciones de trabajos prácticos serán realizadas a partir de entregas de productos de software a través del ciclo de vida de desarrollo.

Se prevé la entrega de los productos de software, en un cronograma específico a definir de acuerdo a cada proyecto real.

Los productos a entregar serán:

- Minutas de reunión
- Especificación de Requerimientos de Software
- Modelo de Datos
- Diseño de Software. Casos de Uso. Mockups.
- Implementación final

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

Se utiliza como plataforma virtual WebUNLP,

Firmas del/los profesores responsables: