



<b>Asignatura</b>	INGENIERIA DE SOFTWARE
<b>Ciclo Lectivo</b>	2010
<b>Vigencia del programa</b>	Ciclo lectivo 2010
<b>Plan</b>	2008
<b>Área</b>	Sistemas de Información
<b>Carga horaria semanal</b>	6 horas
<b>Anual/ cuatrimestral</b>	Cuatrimestral
<b>Coordinador de Cátedra</b>	Ing. Judith Meles
<b>Objetivos de la Materia</b>	<ul style="list-style-type: none"><li>⇒ Conocer los componentes de un proyecto de ingeniería de software.</li><li>⇒ Reconocer la importancia de los conceptos relacionados con la Ingeniería de Software y sus técnicas y herramientas relacionadas.</li><li>⇒ Ser capaz de afrontar la problemática que presenta la planificación y el seguimiento de un proyecto de software.</li><li>⇒ Conocer los estándares asociados a la calidad del proceso de desarrollo de software y de productos de software.</li><li>⇒ Conocer los componentes de los planes de aseguramiento de la calidad y de los planes de prueba.</li><li>⇒ Emplear métricas que se aplican al desarrollo de software.</li><li>⇒ Aplicar los elementos de un proceso de prueba (“testing”) como parte integral del Aseguramiento de Calidad del producto.</li><li>⇒ Integrar por medio de casos prácticos concretos los conocimientos adquiridos en la parte teórica, empleando así las técnicas y herramientas de aplicación de la ingeniería de software.</li></ul>

### Programa Analítico

#### **Unidad Nro. 1: Administración de Proyectos de Software**

##### **Objetivos específicos:**

- ⇒ Introducir a la Administración de Proyectos de Software y describir sus características distintivas.
- ⇒ Introducir los conceptos fundamentales a cerca de la problemática de Administrar Proyectos de Software.
- ⇒ Comprender los motivos que ocasionaron la llamada “crisis del software”.
- ⇒ Comprender la utilidad de los ciclos de vida en el contexto del desarrollo de software y su influencia en la Administración de Proyectos de Software.
- ⇒ Valorar la relación existente entre el Proceso, el Proyecto y el Producto de Software a construir.
- ⇒ Presentar conceptos relacionados a las metodologías ágiles en general y a una metodología de gestión ágil de proyectos de software en particular.



#### Contenidos:

- Introducción a la Ingeniería del Software. ¿Qué es?
- Estado Actual y Antecedentes. La Crisis del Software.
- Ejemplos de grandes proyectos de software fallidos y exitosos.
- Ciclos de vida (Modelos de Proceso) y su influencia en la Administración de Proyectos de Software.
- Ciclos de vida (Modelos de Proceso) y Procesos de Desarrollo de Software
- Ventajas y desventajas de c/u de los ciclos de vida. Criterios para elección de ciclos de vida en función de las necesidades del proyecto y las características del producto.
- Introducción a los Métodos de Desarrollo Ágiles
- Gestión Ágil de Proyectos
- Actividades de Administración de Proyectos de Software.
  - Planificación de Proyectos de Software
  - Gestión de Riesgos
  - Estimación en el contexto del Desarrollo de Software
  - Monitoreo y Control de Proyectos de Software
  - Planificación de Proyectos de Software basada en Procesos de Desarrollo definidos.

#### Bibliografía:

- ❖ **Sommerville, Ian** - INGENIERÍA DE SOFTWARE - Séptima Edición (Editorial Addison-Wesley Año 2005). **Capítulo 1 y 5**
- ❖ **Pressman, Roger** - INGENIERÍA DE SOFTWARE, UN ENFOQUE PRÁCTICO. –(Editorial McGraw Hill – Año 2005) **Capítulo 1, 3, 4, 5, 21, 22, 23, 24 y 25**
- ❖ **Schwaber Ken and Beedle Mike**- AGILE SOFTWARE DEVELOPMENT WITH SCRUM. (Editorial Prentice Hall- Año 2002)- **Capítulos 3 al 7.**

#### Paper:

- ❖ **Deemer Pete and Benefield Gabrielle** – SCRUM PRIMER (Año 2006)

#### Evaluación:

La evaluación de contenidos se hará en el primer parcial como en los ejercicios prácticos de resolución en el aula integrando contenidos de la asignatura entre sí y con otras asignaturas de la carrera.

---

#### Unidad Nro. 2: Métricas de Software

#### Objetivos específicos:

- ↳ Saber utilizar las técnicas de medición y estimación de software.
- ↳ Comprender la importancia de medir para obtener visibilidad de los proyectos de desarrollo de software.
- ↳ Reconocer la importancia de las métricas para la mejora de procesos, proyectos y productos.



### Contenidos

- Métricas, Medidas e Indicadores
- Métricas en el Proceso y Proyecto
- Mediciones del Software
- Orientadas al tamaño
- Orientadas a la función
- Métricas de Calidad
- Integración de Métricas
- Aplicación de las métricas en Administración de Proyectos.

### Bibliografía:

- ❖ **Pressman, Roger** - INGENIERÍA DE SOFTWARE, UN ENFOQUE PRÁCTICO. –(Editorial McGraw Hill – Año 1998) – **Capítulo 15 y 22**

### Evaluación:

La evaluación de contenidos se hará en el segundo parcial como en los ejercicios prácticos de resolución en el aula integrando los contenidos de toda la materia y de materias anteriores de la carrera

---

### Unidad Nro. 3: Gestión de Configuración del Software

#### Objetivos específicos:

- ↷ Reconocer la importancia de la Gestión de Configuración de Software.
- ↷ Describir las actividades de planificación de la Gestión de Configuración de Software.
- ↷ Conocer los componentes de un Plan de Gestión de Configuración de Software.
- ↷ Discutir el uso de algunas herramientas utilizadas para la Gestión de Configuración de Software.

#### Contenidos.

- Conceptos Introdutorias de la Gestión de Configuración
- Versiones, variantes, release
- Planificación de la Gestión de Configuración de Software
- Actividades relacionadas con la Gestión de Configuración
- El rol de las líneas base y su administración.
- Elementos de con figuración del Software
- Identificación de Objetos en la Configuración de Software
- Control de Versiones
- Control de Cambios
- Auditoria de Configuración

#### Bibliografía:

- ❖ **Pressman, Roger** - INGENIERÍA DE SOFTWARE, UN ENFOQUE PRÁCTICO. –(Editorial McGraw Hill – Año 1998) **Capítulo 27**
- ❖ **Sommerville, Ian** - INGENIERÍA DE SOFTWARE - Séptima Edición (Editorial Addison-



Wesley Año 2005). **Capítulo 29**

- ❖ **Bersoff, Edgard** – Elements of Software Configuration Management
- ❖ **Software Program Manager Network** - The Little Book of Software Configuration Management, (AirLie Software Council, 1998)

**Evaluación:**

La evaluación de contenidos se hará en el segundo parcial como en los ejercicios prácticos de resolución en el aula en los que se integrarán los contenidos de toda la materia y de materias anteriores de la carrera

**Unidad Nro. 4: Aseguramiento de Calidad de Proceso y de Producto**

**Objetivos específicos:**

- ↪ Identificar las principales tendencias respecto a la calidad y su forma de incorporarla.
- ↪ Conocer la importancia y la influencia que tienen el trabajo para y con calidad en las organizaciones.
- ↪ Analizar críticamente los principales modelos de calidad de software existentes en el mercado.
- ↪ Identificar técnicas y herramientas para realizar aseguramiento de calidad de software.
- ↪ Descubrir la importancia de la prueba del sistema como una herramienta que controlará la calidad del producto construido.
- ↪ Reconocer y utilizar las diferentes técnicas (auditorías, revisión e inspecciones de software) relacionadas con el aseguramiento de la calidad del proceso y del producto.
- ↪ Conocer técnicas y herramientas para realizar pruebas e inspecciones de software.
- ↪ Poder planificar actividades relacionadas al aseguramiento de calidad de software e insertarlas en el proyecto.

**Contenidos**

- Conceptos generales sobre calidad.
- Importancia de trabajar para y con Calidad. Ventajas y Desventajas.
- Tendencias de la Calidad.
- Actividades relacionadas con el Aseguramiento de la Calidad del Software.
- Modelo de Mejora Procesos de Desarrollo, estructura, definición y características.
- Principales Modelos de Calidad existentes (CMMI – SPICE – ISO)
- Lineamientos para la implementación de modelos de calidad en las organizaciones.
- Métodos de Evaluación de modelos de calidad.
- Calidad de Producto: Planificación de pruebas para el software- Técnicas y herramientas para probar software. Técnicas y Herramientas para Inspección de Software.
- Diferentes tipos de Auditorías:
  1. Auditorías de Proyecto
  2. Auditorías al Grupo de Calidad.
- Proceso de Auditorías: Responsabilidades. Preparación y ejecución. Reporte y seguimiento.



**Bibliografía:**

- ❖ **Sommerville, Ian** - INGENIERÍA DE SOFTWARE - Sexta Edición (Editorial Addison-Wesley Año 2002). **Capítulo 27 y 28**
- ❖ **Pressman, Roger** - INGENIERÍA DE SOFTWARE, UN ENFOQUE PRÁCTICO. –(Editorial McGraw Hill – Año 1998) **Capítulo 26**
- ❖ **Myers, Glenford**- El arte de Probar el Software. (Editorial El Ateneo, 1983).- **Capítulos 1 al 8**
- ❖ IEEE STD 730 STANDARD FOR SOFTWARE QUALITY ASSURANCE PLANS
- ❖ IEEE STD 1028-1997 STANDARD FOR SOFTWARE REVIEWS
- ❖ IEEE STD 1012-1998 (REVISION OF IEEE STD 1012-1986) IEEE STANDARD FOR SOFTWARE VERIFICATION AND VALIDATION

**Evaluación:**

La evaluación de contenidos se hará en el segundo parcial como en los ejercicios prácticos de resolución en el aula y del desarrollo de un Trabajo Práctico en el que se integrarán los contenidos de toda la materia y de materias anteriores de la carrera.



<b>Metodología de enseñanza y aprendizaje</b>	<p>Destacando el hecho que el currículo no solo se manifiesta en la especificación de una serie de contenidos en un programa, sino por el contrario, abarca cuestiones mucho más profundas tales como: bibliografía, priorización de algunos contenidos sobre otros, proceso de enseñanza – aprendizaje, formas de evaluación, entre otras; es que se considera importante poner de manifiesto algunos de estos aspectos con el propósito de mejorar el nivel académico y fomentar la integración de la cátedra, sin interferir, por supuesto, en la libertad de cada uno de los docentes que la integren.</p> <p>La selección de los contenidos incluidos en el programa se realizó considerando la integración de esta nueva asignatura al resto de las asignaturas de la carrera, lo que fundamenta en gran medida la priorización y el nivel de profundidad elegido para cada tema.</p> <p>Dentro de las cuestiones que se expondrán para el desarrollo de la Metodología se tendrán en cuenta los siguientes aspectos:</p> <p><b>Dictado de la materia:</b> el contenido temático está organizado lógicamente y situado coherentemente según su grado de dificultad de manera que permita al estudiante ir asimilando los contenidos propios de la materia en forma gradual y a la vez integrar los contenidos de otras asignaturas.</p> <p>Para el desarrollo del programa se considerará un sistema de clases que combine: clases explicativas, clases prácticas tipo taller, la realización de ejercicios prácticos grupales y el uso del laboratorio. A través de estas diferentes formas organizativas de la enseñanza se proponen los siguientes objetivos educativos:</p> <ul style="list-style-type: none"><li>⇒ Transmitir los conocimientos a través de un proceso de enseñanza-aprendizaje que permita la apropiación de los contenidos.</li><li>⇒ Desarrollar el hábito de la lectura, el análisis y la interpretación de textos, invitando a los alumnos a trabajar con las fuentes bibliográficas originales, posibilitando que elaboren sus propias interpretaciones y realicen sus propias conclusiones.</li><li>⇒ Promover el espíritu investigativo para buscar siempre la verdad auténtica y la rigurosidad de la ciencia en la búsqueda de las soluciones a las situaciones de aprendizaje que se propongan.</li><li>⇒ Valorar el uso de bibliografía como fuente original de los conceptos desarrollados en la asignatura.</li><li>⇒ Fomentar la habilidad para aplicar los conocimientos adquiridos a situaciones concretas.</li></ul>
---	---



<b>Sistema de evaluación</b>	<p><b>Para obtener la aprobación de la asignatura se proponen diferentes actividades, algunas para desarrollar individualmente y otras para desarrollar en forma grupal.</b></p> <p><b><u>Condiciones de Regularidad</u></b> <b>Nivel Grupal:</b></p> <p>1. <i>Ejercicios Prácticos:</i> en cada una de las unidades se le planteará a los estudiantes la realización de uno o más ejercicios de este tipo, cuyo objetivo es lograr la apropiación de los contenidos abordados en cada unidad. Debe aprobar el 70 % de los trabajos prácticos. La no presentación del Ejercicio Práctico lo deja en condición de NO Aprobado.</p> <p>En caso de no alcanzar el porcentaje mínimo de aprobación, cada uno de los integrantes del grupo quedará en condición de <i>libre</i>.</p> <p>Los prácticos deben respetar la fecha de presentación definida por los docentes, el no cumplimiento de las mismas equivaldrá a la no aprobación de la presentación correspondiente.</p>
<b>Condiciones de regularidad</b>	<p>Además de lo descripto en el apartado anterior respecto de los Ejercicios Prácticos, para obtener la regularidad el estudiante deberá rendir y aprobar 2(dos) parciales teórico-prácticos, teniendo la posibilidad de realizar <b>1 (un) recuperatorio al final del cuatrimestre</b>. La aprobación se obtiene con nota mayor o igual a 4 (cuatro), tanto en los parciales como en los Recuperatorios. La nota 4 (cuatro) se corresponde con el 60 % de los contenidos evaluados en el parcial.</p> <p>Debe tener al menos un 50 % del puntaje asignado a cada parte (teórico-práctico) además de cumplir con el puntaje mínimo, para que el parcial se considere aprobado.</p> <p>Tanto los parciales como los Recuperatorios se toman de forma unificada para todos los cursos que conforman la cátedra.</p>
<b>Modalidad de examen final</b>	<p>Para obtener la aprobación de la materia el estudiante deberá aprobar el examen final en el que se evaluarán todos los contenidos del <i>último programa vigente</i> para la asignatura. Se evaluarán aspectos teóricos de la materia.</p> <p><b>El examen final se aprueba con nota mínima no menor a 4 (cuatro), correspondiendo también al 60 % de los contenidos evaluados, al igual que en los exámenes parciales.</b></p> <p><b><u>Importante:</u></b> La cátedra tomará los exámenes finales en forma conjunta para todos los alumnos, esto permitirá la nivelación e integración de todos los cursos que la integran</p>
<b>Actividades en laboratorio</b>	<p><b>Práctica deseable:</b> Utilización de un software para planificación de proyectos, y de un software para gestión de configuración.</p>



<b>Horas/año totales de la asignatura</b>	96 horas
<b>Cantidad de horas prácticas totales</b>	La carga horaria afectada a la formación práctica se corresponde con el 50 % de las horas totales de la asignatura, es decir 48 horas. De este 50 % de carga horaria al menos el 15 % se empleará específicamente en la Resolución de Problemas de Ingeniería. Este 15 % de tiempo se traduce en términos de clases en 5 clases de 3 módulos cada una.
<b>Cantidad de horas teóricas totales</b>	48 horas
<b>Tipo de formación práctica</b> (marque la que corresponde y si es asignatura curricular -no electiva-)	<input type="checkbox"/> Formación experimental <input checked="" type="checkbox"/> Resolución de problemas de ingeniería <input type="checkbox"/> Actividades de proyecto y diseño <input type="checkbox"/> Prácticas supervisadas en los sectores productivos y /o de servicios
<b>Cantidad de horas afectadas a la formación práctica indicada</b>	La carga horaria afectada a la formación práctica se corresponde con el 50 % de las horas totales de la asignatura. De este 50 % de carga horaria al menos el 15 % se empleará específicamente en la Resolución de Problemas de Ingeniería. Este 15 % de tiempo se traduce en términos de clases en 5 clases de 3 módulos cada una.
<b>Descripción de los prácticos</b>	La parte práctica de la materia está basada en el desarrollo de ejercicios prácticos de resolución de problemas de ingeniería. Sobre estos ejercicios prácticos se trabajará cada uno de los contenidos prácticos que se desarrollan en la materia. Adicionalmente se entregará a los alumnos Ejercicios Prácticos resueltos, sobre los que se explicará en clase la forma de resolución de los mismos.
<b>Criterios de evaluación de los prácticos</b>	Sobre los ejercicios prácticos que la cátedra utilizará se evaluarán los siguientes aspectos, vinculados al cumplimiento de los objetivos de la asignatura: <ol style="list-style-type: none"><li>1. Trabajo acorde a las consignas presentadas</li><li>2. Que resuelva correctamente el problema que el ejercicio presenta y cumpla los objetivos definidos para éste</li><li>3. Consistencia entre los diferentes aspectos a resolver en el ejercicio y con el resto de los ejercicios si correspondiera.</li><li>4. Aspecto formal de la presentación del práctico</li><li>5. Cumplimiento de la fecha acordada</li><li>6. Integración del grupo en la realización del trabajo</li></ol>
<b>Formato de presentación de los prácticos</b>	<b>Cada uno de los Proyectos Prácticos de Aplicación estará descrito considerando los siguientes aspectos:</b> <ol style="list-style-type: none"><li>1. Tema que cubre el Ejercicio Práctico</li><li>2. Objetivo del Ejercicio.</li><li>3. Propósito del Ejercicio</li><li>4. Entradas requeridas para su ejecución</li><li>5. Salidas esperadas</li><li>6. Consigna asociada Ejercicio Práctico</li></ol>



	7. Instrucciones, si correspondiera.																																																									
<b>Cronograma de actividades de la asignatura,</b> incluyendo semana prevista para cada práctico	<p>Si bien el Calendario Académico de la Universidad tiene planificadas 16 semanas de clase para el cuatrimestre, si va a plantear un cronograma de 15 semanas de clase debido a los feriados y asuetos que en forma diferente afectan a cada uno de los cursos que conforman la cátedra.</p> <table border="1"> <thead> <tr> <th>Semana</th> <th>Clase Teórica</th> <th>Clase Práctica</th> </tr> </thead> <tbody> <tr> <td>1</td> <td>Presentación de la materia, de los docentes- Encuesta Teórico: Ingeniería de Software – Organizar la lectura del Robot Asesino</td> <td>Práctico: Robot Asesino</td> </tr> <tr> <td>2</td> <td>Ciclo de Vida y Proceso</td> <td>Práctico de Proceso</td> </tr> <tr> <td>3</td> <td>Métodos Ágiles y SCRUM</td> <td>Práctico de SCRUM</td> </tr> <tr> <td>4</td> <td>Planificación de Proyectos – WBS – Calendarización</td> <td>WBS - Cronograma</td> </tr> <tr> <td>5</td> <td>Riesgos</td> <td>Práctico de Riesgos</td> </tr> <tr> <td>6</td> <td>Estimación</td> <td>Práctico de Estimación: Pócker y Wide Band Delphi</td> </tr> <tr> <td>7</td> <td>Métricas</td> <td>Práctico de Métricas</td> </tr> <tr> <td>8</td> <td>Monitoreo y control de Proyectos</td> <td>Práctico de Monitoreo con SCRUM</td> </tr> <tr> <td>9</td> <td>Administración de Configuración de Software</td> <td>Práctico de CM Parte I</td> </tr> <tr> <td colspan="3" style="background-color: #00FF00; text-align: center;"><b>Sábado 9:00 hs Primer Parcial Teórico Práctico</b></td> </tr> <tr> <td>10</td> <td>Teórico QA- Modelos de Mejora</td> <td>Práctico de CM Parte II</td> </tr> <tr> <td>11</td> <td>Teórico CMMi – Foco en nivel II</td> <td>Práctico de CMMi</td> </tr> <tr> <td>12</td> <td>Teórico de Auditorías de Software</td> <td>Práctico de Auditorías</td> </tr> <tr> <td>13</td> <td>Teórico de Calidad de Producto: Inspecciones</td> <td>Práctico de Inspecciones de Código</td> </tr> <tr> <td>14</td> <td>Teórico de Calidad de Producto: Testing</td> <td>Práctico de Testing</td> </tr> <tr> <td colspan="3" style="background-color: #00FF00; text-align: center;"><b>Sábado 9:00 hs Segundo Parcial Teórico y Práctico</b></td> </tr> <tr> <td>15</td> <td colspan="2" style="text-align: center;">Clase de Consulta</td> </tr> <tr> <td colspan="3" style="background-color: #00FF00; text-align: center;"><b>Sábado 9:00 hs Recuperatorios Teórico Práctico</b></td> </tr> </tbody> </table>	Semana	Clase Teórica	Clase Práctica	1	Presentación de la materia, de los docentes- Encuesta Teórico: Ingeniería de Software – Organizar la lectura del Robot Asesino	Práctico: Robot Asesino	2	Ciclo de Vida y Proceso	Práctico de Proceso	3	Métodos Ágiles y SCRUM	Práctico de SCRUM	4	Planificación de Proyectos – WBS – Calendarización	WBS - Cronograma	5	Riesgos	Práctico de Riesgos	6	Estimación	Práctico de Estimación: Pócker y Wide Band Delphi	7	Métricas	Práctico de Métricas	8	Monitoreo y control de Proyectos	Práctico de Monitoreo con SCRUM	9	Administración de Configuración de Software	Práctico de CM Parte I	<b>Sábado 9:00 hs Primer Parcial Teórico Práctico</b>			10	Teórico QA- Modelos de Mejora	Práctico de CM Parte II	11	Teórico CMMi – Foco en nivel II	Práctico de CMMi	12	Teórico de Auditorías de Software	Práctico de Auditorías	13	Teórico de Calidad de Producto: Inspecciones	Práctico de Inspecciones de Código	14	Teórico de Calidad de Producto: Testing	Práctico de Testing	<b>Sábado 9:00 hs Segundo Parcial Teórico y Práctico</b>			15	Clase de Consulta		<b>Sábado 9:00 hs Recuperatorios Teórico Práctico</b>		
Semana	Clase Teórica	Clase Práctica																																																								
1	Presentación de la materia, de los docentes- Encuesta Teórico: Ingeniería de Software – Organizar la lectura del Robot Asesino	Práctico: Robot Asesino																																																								
2	Ciclo de Vida y Proceso	Práctico de Proceso																																																								
3	Métodos Ágiles y SCRUM	Práctico de SCRUM																																																								
4	Planificación de Proyectos – WBS – Calendarización	WBS - Cronograma																																																								
5	Riesgos	Práctico de Riesgos																																																								
6	Estimación	Práctico de Estimación: Pócker y Wide Band Delphi																																																								
7	Métricas	Práctico de Métricas																																																								
8	Monitoreo y control de Proyectos	Práctico de Monitoreo con SCRUM																																																								
9	Administración de Configuración de Software	Práctico de CM Parte I																																																								
<b>Sábado 9:00 hs Primer Parcial Teórico Práctico</b>																																																										
10	Teórico QA- Modelos de Mejora	Práctico de CM Parte II																																																								
11	Teórico CMMi – Foco en nivel II	Práctico de CMMi																																																								
12	Teórico de Auditorías de Software	Práctico de Auditorías																																																								
13	Teórico de Calidad de Producto: Inspecciones	Práctico de Inspecciones de Código																																																								
14	Teórico de Calidad de Producto: Testing	Práctico de Testing																																																								
<b>Sábado 9:00 hs Segundo Parcial Teórico y Práctico</b>																																																										
15	Clase de Consulta																																																									
<b>Sábado 9:00 hs Recuperatorios Teórico Práctico</b>																																																										
<b>Descripción de metodología propuesta de consultas y cronograma de consultas</b>	<p>La cátedra tiene planificada una clase de consulta la clase previa a cada uno de los parciales, inclusive los recuperatorios.</p> <p>Los horarios de consulta deberán convenirlos en cada curso con sus docentes.</p> <p>A todo efecto la cátedra dispone de la siguiente dirección de correo para que los alumnos se puedan comunicar directamente con el Coordinador de Cátedra: <b>jmeles@gmail.com</b>.</p>																																																									
<b>Plan de integración con otras asignaturas</b>	<p>Esta materia utilizará en gran medida lo aprendido en ASI (Análisis de Sistemas) en el segundo nivel, en DSI (Diseño de Sistemas), directa continuadora de los contenidos de la materia ASI, en Gestión de Datos y las materias del área de programación, dado que se asume que el alumno ya maneja las herramientas y técnicas necesarias para construir un software. En este sentido, el foco principal de la Asignatura Ingeniería de Software es fortalecer las disciplinas denominadas “protectoras”, que transversalmente van apoyando el desarrollo del producto desde el inicio hasta el final, me refiero a</p>																																																									



	<p>las disciplinas de Gestión de Configuración, Aseguramiento de Calidad, Verificación y Validación y Administración de Proyectos.</p> <p>Además se prevé la integración con (ARE) Administración de Recursos, complementará a Ingeniería de Software con el dictado de contenidos vinculados a Auditoría Informática y Peritaje.</p>
<b>Bibliografía Obligatoria</b>	<ul style="list-style-type: none"><li>❖ <b>Sommerville, Ian</b> - INGENIERÍA DE SOFTWARE - Sexta Edición (Editorial Addison-Wesley Año 2005).</li><li>❖ <b>Pressman, Roger</b> - INGENIERÍA DE SOFTWARE, UN ENFOQUE PRÁCTICO. -(Editorial McGraw Hill – Año 2005)</li><li>❖ <b>Myers, Glenford</b>- El arte de Probar el Software. (Editorial El Ateneo, 1983).- <b>Paper:</b><ul style="list-style-type: none"><li>❖ <b>Deemer Pete and Benefield Gabrielle</b> – SCRUM PRIMER (Año 2006)- <b>Sitio:</b> <a href="http://www.rallydev.com">http://www.rallydev.com</a></li></ul></li></ul>
<b>Bibliografía Complementaria</b>	<ul style="list-style-type: none"><li>❖ <b>Steve Mc Connell.</b>, DESARROLLO Y GESTIÓN DE PROYECTOS INFORMÁTICOS (Editorial McGraw Hill – Año 1996).</li><li>❖ <b>Humphrey, W.S.</b> - MANAGING THE SOFTWARE PROCESS, Addison-Wesley, Reading, MA, 1989.</li><li>❖ <b>Jacobson, Ivar</b> - OBJECT-ORIENTED SOFTWARE ENGINEERING. (Editorial Addison-Wesley Año 1994).</li><li>❖ <b>SPICE PROJECT</b>, CONSOLIDATED PRODUCT. SOFTWARE PROCESS ASSESSMENT – PART 1: CONCEPTS AND INTRODUCTORY GUIDE. VERSION 1.00. Site de SPICE: <a href="http://www.esi.es/Projects/SPICE">www.esi.es/Projects/SPICE</a></li><li>❖ <b>McFeeley, Bob</b> - IDEAL: A USER GUIDE FOR SOFTWARE PROCESS IMPROVEMENT – CMU/SEI-96-HB-001. <a href="http://WWW.SEI.CMU.EDU">WWW.SEI.CMU.EDU</a></li><li>❖ <b>Sitio de la IEEE:</b> <a href="http://www.ieee.org">http://www.ieee.org</a><ul style="list-style-type: none"><li>❖ IEEE STD 730 STANDARD FOR SOFTWARE QUALITY ASSURANCE PLANS</li><li>❖ IEEE STD 1028-1997 STANDARD FOR SOFTWARE REVIEWS</li><li>❖ IEEE STD 1012-1998 (REVISION OF IEEE STD 1012-1986) IEEE STANDARD FOR SOFTWARE VERIFICATION AND VALIDATION</li><li>❖ IEEE STD 1028-1997 STANDARD FOR SOFTWARE REVIEWS</li><li>❖ ISO/IEC FDIS 17799 INFORMATION TECHNOLOGY — SECURITY TECHNIQUES — CODE OF PRACTICE FOR INFORMATION SECURITY MANAGEMENT</li></ul></li><li>❖ <b>Schwaber Ken and Beedle Mike</b>- AGILE SOFTWARE DEVELOPMENT WITH SCRUM. (Editorial Prentice Hall- Año 2002)</li><li>❖ <b>Bersoff, Edgard</b> – Elements of Software Configuration Management – <b>Sitio:</b> <a href="http://portal.acm.org">http://portal.acm.org</a></li><li>❖ <b>Software Program Manager Network</b> - The Little Book of Software Configuration Management, (AirLie Software Council, 1998)- <b>Sitio:</b> <a href="http://www.spmn.com">http://www.spmn.com</a></li></ul>



<b>Distribución de docentes por curso</b>	<b>Curso</b>	<b>Día y Horas</b>	<b>Turno</b>	<b>Profesor</b>	<b>J.T.P.</b>	<b>Ayudantes</b>
	4K1	Mar 3-4-5-6 Jue 4-5	M	Meles, Judith	Abdala, Valeria	Massano, Cecilia
	4K2	Mar 1-2-3-4 Vie 3-4	T	Battistelli, Daniel	Abdala, Valeria	López Consuelo
	4K3	Mie 3-4-5-6 Vie 5-6	N	Battistelli, Daniel	Covaro, Laura	López Consuelo Farías Facundo
	4K90	Mar 3-4-5-6 Vie 1-2	N	Meles, Judith	Massano, Cecilia	-