

## Programa de Asignatura

<b>00 Código:</b>	F	C	Asign.
	T	109	26
	T	108	21

**01. Facultad: Tecnología Informática / Carrera: Ingeniería en Sistemas Informáticos**

**02. Asignatura:** T109 – 26 - TRABAJO DE CAMPO I  
 T108 – 21 - TRABAJO DE CAMPO I

**03. Año lectivo: 2015 04. Año de Cursada: 3° 05. Cuatrimestre: 1°**

**06. Hs. Semanales: 6.**  
**Hs. Tutoriales: 100.**

**07.**

<b>Profesor Titular:</b> Pons Claudia	
<b>Buenos Aires</b>	<b>Rosario</b>
<b>Titular a cargo:</b> Pons Claudia	<b>Asociado a cargo:</b> Vaquero Marcelo
<b>Adjuntos:</b> Cardacci, Darío Domenech, Carlos Contreras, Victor Calabria, Esteban Defino, Guillermo Barnech, Diego	<b>Adjunto:</b> Audoglio, Pablo
<b>Eje socioprofesional: Los Modelos como representación abstracta de la realidad (Análisis y Lenguajes)</b>	
<b>Coordinador de eje:</b> Ing. Vilaboa Pablo	<b>Coordinador de eje:</b> Ing. Vaquero Marcelo
<b>Eje Epistémico: Administración de Recursos Tecnológicos.</b>	

### 08. Fundamentación:

La producción de un sistema de información es una regla de medida importante que permite evaluar el grado de asimilación de los conocimientos. La práctica guiada permite fortalecer el perfil profesional del futuro egresado, aportando un criterio analítico fusionando los conocimientos y canalizando el saber.

La materia promueve con el uso de las técnicas avanzadas de modelización la construcción de sistemas de información adaptativos. Estimula las habilidades personales para construir

componentes de software y facilita las técnicas necesarias para aprovechar diferentes métricas de calidad con el fin de evaluar el rendimiento de su producción.

Por último le permite complementar los saberes previos con nuevos conocimientos que lo llevarán a un estadio mayor en la interpretación y uso de nuevos paradigmas en la ingeniería del software.

#### **09. Ítems del perfil que se desarrollarán:**

Los sistemas de información que le brindan a las organizaciones la posibilidad de operar de manera eficiente y eficaz en contextos permanentemente cambiantes, deben estar contruidos bajo las pautas de la ingeniería del software. Los profesionales preparados para desarrollar esta actividad tienen que manifestarse con soltura en las actividades que se indican a continuación. Habilidad y predisposición para la lectura crítica de la realidad, la investigación y la innovación aplicada a la invención.

Incorporar las destrezas para transferir los avances de las ciencias informáticas a los diversos servicios.

Desarrollar las habilidades que permitan establecer una clara planificación y administración de los recursos y las actividades de documentación, que ayudarán a desarrollar software reutilizable y le dará al producido un marco de estabilidad y escalabilidad superior.

Brindarle a la actividad de construcción de software un enfoque que se adecue a los lineamientos de la ingeniería del software.

Obtener el equilibrio para discernir sobre los aportes que otorga aplicar métricas en el desarrollo de software.

Perfeccionar las capacidades para llevar a cabo el trabajo de producción tanto en forma individual como grupal.

Poder tomar decisiones bajo estrictas valoraciones técnicas, éticas y morales.

**10. Correlativas previas:** Metodología de Desarrollo de Sistemas II. Lenguajes de Última Generación.

**Posteriores:** Trabajo de Diploma.

#### **11. Articulación con materias del mismo año:**

Articulación vertical:

Las materias Metodologías de desarrollo de sistemas I y Metodologías de desarrollo de sistemas II aportan los conocimientos necesarios para que el alumno logre alcanzar el nivel de abstracción suficiente para crear los modelos representativos de un sistema de información. Las asignaturas lenguaje de última generación y programación orientada a objetos contribuyen el saber preciso para comprender el paradigma de objetos y con ello alcanzar la cuota de interpretación teórica y practica para modelar la solución tecnológica propuesta. Al culminar el alumno obtendrá las habilidades técnicas y profesionales para nutrir las asignaturas Seminario de Aplicación Profesional y Trabajo Final de Ingeniería.

Articulación horizontal:

Para implementar un modelo de datos y la base de datos asociada, aplica los temas tratados en la asignatura Base de Datos.

## 12. Objetivos:

- Reafirmar los conocimientos sobre técnicas de desarrollo de software orientada a generar sistemas de información para las organizaciones.
- Integrar los conocimientos sobre relevamiento, análisis y diseño de software.
- Utilizar técnicas avanzadas para gestión de proyectos.
- Comprender los elementos que se utilizan para lograr una óptima gestión del riesgo.
- Construir aplicaciones que posean sólidos mecanismos de control y gestión de calidad.
- Implementar los elementos que permitan incorporar los conceptos de ingeniería del software orientado a objetos.
- Adquirir conocimientos referidos al ciclo de vida del producto (software) así como el análisis estratégico sobre la conveniencia en tiempo y forma del mantenimiento del mismo.
- Desarrollar las habilidades que se necesitan para proporcionar una planificación y seguimiento del proyecto acorde a las necesidades del mercado.
- Asociar los conocimientos de producción de software a las métricas referidas, a fin de obtener indicadores que mejoren la rentabilidad en la producción.
- Comprender el aporte que la ingeniería basada en componentes al desarrollo de software.
- Utilizar técnicas avanzadas en la modelización de sistemas de información.
- Adaptar los sistemas de información al contexto variable.
- Elaborar hipótesis de investigación sobre el desarrollo de sistemas de información.
- Desarrollar un sistema de información articulando e integrando desde un proyecto práctico común con las asignaturas Metodologías de desarrollo de sistemas II y Trabajo de Diploma.

## 13. Unidades de desarrollo de los contenidos:

### UNIDAD DE REPASO Y CONOCIMIENTOS NECESARIOS

#### Evaluación Diagnóstica.

##### Repaso:

- Aspectos esenciales a tener en cuenta en el desarrollo de software.
- Análisis y diseño de sistemas.
- Análisis y diseño de sistemas orientados a objetos.

##### Conocimientos necesarios:

- Análisis y diseño de sistemas.
- Técnicas y estrategias de programación estructurada.
- Técnicas y estrategias de programación orientada a objetos.
- Acceso a base de datos.

Tiempo: 6 hs

### Unidad 1: PROCESOS PARA LA CREACIÓN DE SOFTWARE.

#### El Software.

Producto. Definición y alcance de un sistema. Selección de herramientas. Análisis estratégico sobre la vida del producto (software). Pautas de Mantenimiento. Reconvertir Vs. Mantener.

Proceso. Métodos y herramientas. Etapas en la construcción de un software. Ciclo económico del software. Modelos de procesos. Modelo lineal secuencial. Modelo de prototipos. El modelo DRA.

Modelos evolutivos: modelo incremental, modelo espiral, modelo espiral winwin, modelo de desarrollo concurrente. Modelo de ensamblaje de componentes. Modelo de métodos formales. Formación y gestión de equipos de trabajo. Conceptos de planificación y seguimiento de proyectos de software.

Desarrollo ágil de aplicaciones. Conceptos generales. Programación extrema XP. Scrum. Otros modelos ágiles de desarrollo.

Tiempo: 12 hs

## **Unidad 2: ANÁLISIS Y DISEÑO DE UN MÓDULO DE SEGURIDAD Y SERVICIOS.**

Módulo de Seguridad.

Aspectos a considerar en un módulo de seguridad. Conceptos. Log in – Log out. Gestión de perfiles. Administración de la bitácora de trabajo. Concepto de parametrización. Introducción a la encriptación. Seguridad para la integridad de datos. Administración de múltiples idiomas. Gestión del backup. Qué consideraciones se deben tener para crear código seguro en las aplicaciones.

Mapa de navegación. Interfaces de usuario. Administración de errores.

Tiempo: 12 hs

## **Unidad 3: METRICAS**

Métricas de proyectos. Medida, métrica e indicadores. Mediciones del software. Integración de las métricas al proceso de ingeniería del software. Adaptación de las métricas a pequeñas y medianas empresas.

Métricas técnicas del software. Métricas para la calidad. Métricas del modelo de análisis. Métricas del modelo de diseño. Métricas del código fuente. Métricas para la prueba. Métricas de Mantenimiento.

Métricas técnicas para sistemas orientados a objetos. Características distintivas. Métricas para el diseño orientado a objetos. Métricas orientadas a las clases. Métricas orientadas a operaciones. Métricas para pruebas orientadas a objetos. Métricas para proyectos orientados a objetos.

Tiempo: 12 hs

## **Unidad 4: GESTIÓN DEL RIESGO Y GARANTÍA DE CALIDAD**

Riesgo del software. Identificación del riesgo. Proyección del riesgo. Tablas para la evaluación del riesgo. Evaluación del impacto. Refinamiento del riesgo. Reducción y gestión del riesgo. Riesgos y peligros para la seguridad. Plan de reducción, supervisión y gestión del riesgo RSGR.

Concepto de calidad. Garantía de calidad del software. Revisiones del software. Concepto de error y defecto. Amplificación y eliminación de errores. Revisiones técnicas formales. Fiabilidad del software. Fiabilidad y disponibilidad. Estándares de Calidad. ISO-9000. ISO-9001. ISO-9002. ISO-9003. ISO-12207. ISO-15504. Mejora del proceso de software y determinación de la capacidad SPICE (Software Process Improvement and Capability Determination). CMM (Modelo de madurez de capacidades). CMMI (Integración del modelo de madurez de capacidades). El plan SQA (Plan de garantía de calidad del software).

Tiempo: 12 hs

## **Unidad 5: INTRODUCCIÓN A LOS PATRONES DE DISEÑO.**

Concepto de patrón de diseño. Consideraciones a tener en cuenta cuando se trabaja con patrones de diseño. Relación entre los patrones de diseño y el desarrollo de software. Análisis y detección de los problemas que pueden ser resueltos por patrones. Aspectos a considerar para seleccionar un patrón de diseño. Formas adecuadas y buenas prácticas para la utilización de un patrón de diseño. Antipatrones.

### **Patrones de creación:**

**Singleton.** Introducción y desarrollo para garantizar que una clase tenga solo una instancia y proporcionar un punto de acceso global a la misma.

**Abstract Factory.** Introducción y desarrollo para proporcionar una interfaz que permita crear familias de objetos relacionados o que dependan entre sí, sin especificar sus clases concretas.

**Factory Method.** Introducción y desarrollo para definir una interfaz que permite crear un objeto, dejando que sean las subclases quienes decidan que clase instanciar. Delegación de la creación de objetos de una clase en sus subclases.

### **Patrones estructurales:**

**Adapter.** Introducción y desarrollo para convertir la interfaz de una clase en otra, que es la que esperan los clientes. Colaboración entre clases que poseen interfaces incompatibles.

**Composite.** Introducción y desarrollo para componer objetos en forma de árbol para representar jerarquías del tipo parte – todo.

**Decorator.** Introducción y desarrollo para asignar responsabilidades adicionales a un objeto dinámicamente. Extensión de las funcionalidades sin utilizar herencia.

### **Patrones de comportamiento:**

**Interpreter.** Introducción y desarrollo para lograr que dado un lenguaje se pueda definir una representación de su gramática junto con un intérprete.

**Iterator.** Introducción y desarrollo para proporcionar un modo de acceder secuencialmente a los elementos de un objeto agregado sin exponer su representación interna.

**Mediator.** Introducción y desarrollo para definir un objeto que encapsula como interactúan una serie de objetos.

**Memento.** Introducción y desarrollo para recordar, representar y externalizar el estado de un objeto sin violar la encapsulación. Retorno a un estado anterior.

**Observer.** Introducción y desarrollo para definir una dependencia de uno a muchos entre objetos. Notificación de cambio de estado entre objetos dependientes.

Tiempo: 24 hs

## **Unidad 6 PRUEBA DE SOFTWARE**

Técnicas de prueba.

Fundamento de las pruebas del software. Diseño de casos de prueba. Pruebas de caja blanca. Pruebas del camino básico. Complejidad ciclomática. Prueba de la estructura de control. Pruebas de caja negra. Prueba de entornos especializados, arquitecturas y aplicaciones.

Estrategias de prueba.

Enfoqué estratégico para la prueba de software. Aspectos estratégicos. Prueba de unidad. Pruebas de integración. Integración ascendente. Integración descendente. Prueba de regresión. Prueba de humo. Pruebas de validación. Pruebas alfa. Pruebas beta. Prueba del sistema. Prueba de recuperación.

Prueba de seguridad. Prueba de resistencia. Prueba de rendimiento. Depuración. Enfoques de la depuración.

Pruebas orientadas a objetos

Pruebas de los modelos de análisis orientado a objetos y diseño orientado a objetos. Estrategias de pruebas orientadas a objetos. Pruebas de unidad, integración y validación en el contexto orientado a objetos. Diseño de casos de pruebas para software orientado a objetos. Métodos de prueba aplicables al nivel de clases.

Tiempo: 18 hs

#### **14. Metodología de trabajo:**

La forma de desarrollar la clase permitirá la articulación de los conocimientos teóricos y prácticos. Las clases poseen dos momentos marcados, el del abordaje hacia el marco teórico y el desarrollo e implementación de la práctica.

Se tendrá especial atención en la participación activa de los alumnos, para poder rescatar la elaboración conceptual que construyan de los temas tratados.

Se aplicarán diversas técnicas y metodologías en el plano teórico y práctico entre las cuales podemos mencionar:

Teoría: 50% del tiempo de la clase. Exposición docente. Diálogo con los alumnos. Puesta en común.

Conclusiones grupales. Inteconsultas. Revisión de temas. Relación de temas nuevos con conocimientos adquiridos de la asignatura y de otras asignaturas relacionadas.

Práctica: 50% del tiempo de la clase. La ejercitación deberá reflejar los conceptos transmitidos en la clase teórica, más los conocimientos de las clases anteriores. Se propondrán ejercicios cuya resolución será grupal en algunos casos e individual en otros. Los alumnos prepararán trabajos de investigación que incluirán la exposición oral de los mismos.

Los medios audiovisuales ocuparán un importante lugar en el desarrollo de las clases para lograr un mayor estímulo en el alumno y de esa forma fijar mejor los contenidos directrices.

Los textos de actualidad y las notas periodísticas sobre tecnología de punta, serán propuestos para ser leídos y tratados en clase.

Se fomentará la participación de los alumnos en la formulación de soluciones a los problemas de la práctica profesional.

Se desarrollarán ejercicios que denoten un alto grado de relación grupal para aprovechar la sinergia de la relación interpersonal.

Herramientas:

- Software de modelado de datos.
- Software de modelado de sistemas.
- Software de documentación.
- Lenguaje de programación orientado a objetos.
- Laboratorio.

#### **15. Trabajos prácticos:**

**a. Trabajo práctico final.** Refinar la carpeta de proyecto que se desarrolló en Metodología de Desarrollo de Sistemas II. Los alumnos utilizarán las 100 hs de tutoría para desarrollar las distintas etapas del proyecto. Se realizan dos entregas parciales del trabajo antes de la entrega final con el objetivo de efectuar un seguimiento incremental del mismo.

Deberá incorporar los elementos que se muestran en la siguiente lista y refinar los que se desarrollaron en materias anteriores.

- 1            Introducción
- 2            Requerimientos mínimos
- 3            Artefactos entregables
  - 3.1           Documento Visión
  - 3.2           Descripción Global del Producto
  - 3.3           Descripción Detallada del Producto
  - 3.4           Restricciones
  - 3.5           Requisitos de Documentación
  - 3.6           Especificaciones de Casos de Uso
  - 3.7           Especificaciones Adicionales
  - 3.8           Mapa de Navegación
  - 3.9           Prototipos de Interfaces de Usuario
  - 3.10          Diagrama de Clases Global del Sistema
  - 3.11          Diagrama de Componentes
  - 3.12          DER Global del Sistema
  - 3.13          Especificación de Casos de Prueba
  - 3.14          Manual de Instalación
  - 3.15          Ayuda en Línea
  - 3.16          Guías de Instalación, Configuración, y Fichero Léame

### 3.17 Material de Apoyo al Usuario Final

El trabajo es individual y se deberá entregar una copia impresa y otra en soporte magnético al docente.

**b. Trabajo de Investigación.** Los trabajos de investigación estarán referidos a temas de la asignatura. Se realizarán en forma grupal y el número de sus integrantes será de cuatro a seis alumnos. La entrega de los mismos será fijada oportunamente por el docente dentro del plazo en que se desarrolla el cuatrimestre.

#### **16. Bibliografía:**

##### Obligatoria:

Gamma, Erich; Helm, Richard; Johnson, Ralph y otros. **Patrones de diseño : elementos de software orientado a objetos reutilizable.**-- Madrid : Pearson Educación, c2003. xvii, 364 páginas

Pressman, Roger S. **Ingeniería del software : un enfoque práctico.** 7a.ed.-- México,DF : McGraw-Hill, c2005. xxxiv, 958 páginas

##### Ampliatoria:

Cardacci Dario y Booch, Grady. **Orientación a Objetos. Teoría y Práctica.** -- Buenos Aires, Argentina. Pearson Argentina, 2013. 400 páginas

Braude, Eric J. **Ingeniería de software : una perspectiva orientada a objetos.**-- México, DF : Alfaomega, c2003. xxiii, 539 páginas

Date , C. J. **Introducción a los sistemas de bases de datos.** 7a.ed.-- México, DF : Addison Wesley Longman, c2001. xxi, 936 páginas

Gómez Vieites, Alvaro; Suárez Rey, Carlos. **Sistemas de información : herramientas prácticas para la gestión empresarial.** 2a. ed.-- México, DF : Alfaomega Grupo Editor, 2007. xx, 233 páginas

Hay Group. **Gestión de proyectos.** – Madrid : McGraw-Hill, 2002.

Jacobson, Ivar; Booch, Grady; Rumbaugh, James y otros. **El proceso unificado de desarrollo de software.**-- Madrid : Pearson Educación, c2000. xiv, 438 páginas

Jiménez Murillo, José Alfredo. **Matemáticas para la computación.**-- México, DF : Alfaomega, c2009. xv, 496 páginas

Kendall, Kenneth E.; Kendall, Julie E. **Análisis y diseño de sistemas.** 6a.ed.-- México, DF : Pearson Educación de México, 2005. xxvi, 726 páginas



Laudon, Kenneth C.; Laudon, Jane Price. **Administración de los sistemas de información : organización y tecnología**. 3a. ed.-- Méxiico, DF : Prentice Hall Hispanoamericana, c1996. xxxviii, 885 páginas

Martin, James; Odell, James J. **Métodos orientados a objetos : conceptos fundamentales**-- México, DF : Prentice Hall Hispanoamericana, c1997. xv, 412 páginas

Meyer, Bertrand. **Construcción de Software Orientado a Objetos**. -- 2ª ed.-- Madrid : Prentice Hall, 1999.

O'Brien, James A.; Marakas, George M.; Herrero Díaz, María Jesús (Traductor) y otros. **Sistemas de información gerencial**. 7a. ed.-- México, DF : McGraw-Hill Interamericana, 2006. xxx, 592 páginas

Pfleeger, Shari Lawrence; Ruíz de Mendarozqueta, Alvaro; Quiroga, Elvira(Traductor). **Ingeniería de software : teoría y práctica**-- Buenos Aires : Pearson Education, c2002. xxv, 760 páginas

Piattini Velthuis, Mario G.; García Rubio, Félix O. **Calidad en el desarrollo y mantenimiento del software**-- México, DF : Alfaomega, c2003. xxv, 310 páginas

Piattini, Mario; Villalba, José; Ruiz, Francisco; Bastanchury, Teresa; Polo, Macario; Martínez, Miguel Ángel ; Nistal, César. **Mantenimiento del software : modelos, técnicas y métodos para la gestión del cambio**. Alfaomega, 2001.

Schach, Stephen R.; Fernández, Esther (Traductor); Guerrero, Ekaterina (Traductor) y otros. **Ingeniería de software clásica y orientada a objetos**. 6a. ed.-- México, D.F. : McGraw-Hill Interamericana de México, 2006. xxi, 581 páginas

Senn, James A. **Análisis y diseño de sistemas de información**. 2a. ed.-- Mexico, DF : McGraw-Hill, 1998. xviii, 942 páginas

Sommerville, Ian; Alfonso Galipienso, María Isabel (Traductor); Botía Martínez, Antonio (Traductor) y otros. **Ingeniería del software**. 7a. ed.-- Madrid : Pearson Educación, 2005. xviii, 687 páginas

Weitzenfeld, Alfredo. **Ingeniería de software orientada a objetos con UML, Java e Internet**-- Bogotá : Thomson Editores, c2005. xx, 678 páginas

## **17. Procedimiento de evaluación y criterio de promoción:**

Parciales.

Los exámenes parciales serán como mínimo dos, cada uno con teoría y práctica. Ambos son obligatorios, individuales y escritos. El primero de ellos se efectuará entre la quinta y séptima semana del cuatrimestre y el segundo entre la semana doce y catorce. Los parciales deberán tener un 60% de contenido correcto para estar aprobados. En este último caso la nota deberá quedar establecida entre 4 (cuatro) y 10 (diez) puntos. Si el alumno no se presenta a un examen parcial se calificará con una nota insuficiente.

#### Recuperatorios.

Los exámenes recuperatorios se desarrollarán a razón de uno por cada parcial con teoría y práctica. Se podrá optar por tomar un recuperatorio integral en aquellos casos que se justifique y el mismo deberá ser equivalente al alcance de los recuperatorios individuales. Recuperarán los parciales que correspondan, aquellos alumnos que obtuvieron una calificación inferior a 4 (cuatro) puntos en alguno de sus exámenes parciales. Los recuperatorios son individuales y por escrito. El docente dispondrá la fecha para realizar esta actividad que podrá ser con posterioridad a cada uno de los parciales o en una fecha común para todos. Las notas de los exámenes recuperatorios no reemplazan las calificaciones insuficientes obtenidas en los exámenes regulares que se están recuperando, las mismas serán tomadas como parte de las calificaciones que intervienen en el promedio.

#### Aspectos comunes a parciales y recuperatorios.

La confección de los exámenes parciales y recuperatorios deberán contemplar diversas técnicas para poder observar desde distintos ángulos el objeto de evaluación. Se deberán intercalar preguntas a desarrollar, del tipo múltiples casos, guiadas y lógicas. Se evaluará la capacidad de razonamiento del alumno y el marco teórico que lo sustenta. Se recomienda ser objetivo y extremar la claridad de la pregunta no dejando lugar a duda de lo que se está intentando evaluar.

#### Trabajos prácticos.

Se desarrollarán los trabajos prácticos enunciados en el punto 14 (trabajos prácticos). El trabajo práctico final deberá ser defendido por el alumno. El trabajo y la defensa será individual y la evaluación del mismo estará conformada por un promedio de notas que involucren la producción y el desempeño. Para ambos trabajos prácticos los aspectos a tener en cuenta en dicha evaluación son: calidad del trabajo, preparación y clasificación del material, producción del material, calidad de la defensa, aprovechamiento de los medios y administración del tiempo.

#### Nota conceptual.

Una nota conceptual basada en la participación en clase del alumno, el interés demostrado en la asignatura y el empeño puesto de manifiesto en las actividades de estudio.

#### Otros aspectos generales de la evaluación.

También se confeccionarán grillas de seguimiento las cuales deberán reflejar la evolución de los alumnos en los aspectos evaluados.

Los cuestionarios de autoevaluación elaborados para el curso poseen como objetivo el desarrollo de un espíritu crítico en el alumno sobre su desempeño en la futura práctica profesional.