

Programa de Asignatura

00 Código:	F	C	Asign.
	T	109	29
	T	108	24

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

02. Asignatura: ***T109 – 29 - TRABAJO DE DIPLOMA***
T108 – 24 - TRABAJO DE DIPLOMA

03. Año lectivo: **2018**

04. Año de Cursada: **3º**

05. Cuatrimestre: **2º**

06. Hs. Semanales: **6.**

Hs. Tutoriales: **100.**

07.

Profesor Titular: Ing. Cardacci Darío Guillermo	
Buenos Aires	Rosario
Titular a cargo: Ing. Cardacci Darío Guillermo	Asociado a cargo: Ing. Vaquero Marcelo
Adjuntos: Domenech Carlos Defino Guillermo Principale Lisandro Battaglia Nicolás Weingan Gastón Ghigliani Leonardo Ayudante Marcos Merlico	Adjunto: Audoglio, Pablo Auxiliares Juzefiszyn, Mauro Tesone, Federico
Eje socioprofesional: Los Modelos como representación abstracta de la realidad (Análisis y Lenguajes)	
Coordinador de eje: Ing. Vilaboa Pablo	Coordinador de eje: Ing. Vaquero Marcelo
Eje Epistémico: Administración de Recursos Tecnológicos.	

08. Fundamentación:

Es fundamental para la formación del futuro analista de sistemas integrar los conocimientos adquiridos así como implementar las técnicas y buenas prácticas que impone la ingeniería de software respecto al desarrollo de sistemas informáticos.

La refactorización de piezas de código, la aplicación de técnicas avanzadas de programación y patrones de diseño, aplicados en un caso práctico testigo, hacen que los desarrollos posean una estructura constructiva más apta a los requerimientos profesionales.

En particular las técnicas avanzadas abordadas en la asignatura, potencian al modelo orientado a objetos trabajado en asignaturas previas. Por medio del subprocesamiento, se logra construir sistemas que funcionen en paralelo y con distintos niveles de prioridad. La reflexión aporta una característica que permite observar la estructura dinámica y estática de las clases y objetos en tiempo de ejecución, lo cual le da a los diseños una versatilidad altamente buscada en el software actual. A través del aprovechamiento de la serialización se puede almacenar un objeto con el objetivo, entre otros, de poder transmitirlo a través de una conexión lo cual facilita notablemente el desarrollo de software para que funcione en entornos distribuidos.

09. Ítems del perfil que se desarrollarán:

La tarea de los analistas de sistemas excede el mero desarrollo de software. Dentro del área profesional que les incumbe, deben dominar las técnicas y metodologías que le permitirán realizar una correcta administración de los recursos informáticos.

Los profesionales preparados para desarrollar esta actividad deben manifestarse con soltura en las actividades que involucran los aspectos que se indican a continuación.

Habilidad y predisposición para lograr crear ámbitos virtuales de solución a problemas reales, utilizando y administrando eficientemente los recursos informáticos.

Incorporar las destrezas para lograr transformar la observación de la realidad compleja a una representación simplificada y sistémica.

Desarrollar las habilidades que permitan implementar soluciones probadas y estándar que otorgan seguridad y eficiencia en las actividades llevadas a cabo.

Capacidad y motivación para innovar en los conceptos referidos a la ingeniería del software.

Conocimiento para la toma de decisiones que permitan comprender, adaptar y aplicar conceptos altamente calificados para el desarrollo de software.

Desarrollar la capacidad para trabajar en forma individual y/o grupal.

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

10. Correlativa previa:

Trabajo de Campo I.

Posteriores: Práctica Profesional Supervisada

11. Articulación con asignaturas:

Articulación vertical:

La articulación con las asignaturas Metodología de Desarrollo de Sistemas I y Metodología de Desarrollo de Sistemas II se produce a partir del aporte que éstas realizan respecto al análisis y la modelización de sistemas así como el uso de una simbología estándar para representación de

los mismos. De las asignaturas Lenguajes de Última Generación y Programación Orientada a Objetos se obtienen los conocimientos de programación y acceso a datos para desarrollar un sistema de información.

A su vez esta asignatura le provee los saberes referidos a técnicas avanzadas de diseño, refactorización y aspectos avanzados de programación a las asignaturas Seminario de Aplicación Profesional y Trabajo Final de Ingeniería.

Articula también con la asignatura Trabajo de Campo I ya que toma los contenidos de ingeniería de software y el desarrollo del trabajo práctico (una carpeta de proyecto de desarrollo de software) y la implementa en un producto de software. Para la implementación también se construye un modelo de datos y la base de datos asociada, temas tratados en la asignatura Base de Datos.

12. Objetivos:

- Comprender como analistas la complejidad de la administración de un proyecto de sistemas.
- Mejorar las técnicas de construcción de software que permitirán obtener productos de calidad.
- Reafirmar los conocimientos de técnicas de desarrollo de software orientada a generar sistemas de información para las organizaciones.
- Integrar los conocimientos sobre diseño arquitectónico avanzado y desarrollo de software.
- Utilizar técnicas avanzadas de gestión de proyectos.
- Comprender los elementos que se utilizan para lograr una óptima gestión de la configuración de los sistemas de información.
- Implementar los elementos que permitan incorporar los conceptos de ingeniería de software orientado a objetos.
- Desarrollar las habilidades que se necesitan para construir sistemas de información con componentes distribuidos.
- Asociar los conocimientos sobre tecnologías de desarrollo que permitan implementar sistemas de información cliente-servidor administrando subprocesos simples y múltiples.
- Utilizar técnicas avanzadas en la implementación de sistemas de información.
- Adaptar los sistemas de información al contexto variable y a la web.
- 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 Campo.

13. Unidades de desarrollo de los contenidos:

UNIDAD DE REPASO Y CONOCIMIENTOS NECESARIOS

Evaluación Diagnóstica.

Repaso:

Teoría y práctica sobre aspectos relevantes relacionados con el desarrollo de software.

Conocimientos necesarios:

Técnicas y estrategias de programación estructurada.
Técnicas y estrategias de programación orientada a objetos.
Análisis y diseño de software orientado a objetos.
Acceso a base de datos.

Tiempo: 6 hs

Unidad 1: DESARROLLO DE SOFTWARE.

El Software.

Complejidad del software. La construcción del software como negocio. Costo – Beneficio en la migración de software a nuevas tecnologías. Ciclo del proyecto (involucrar componente humano y económico). Análisis de actualizar, comprar, adoptar vs desarrollar.

Tiempo: 6 hs

Unidad 2: CONSTRUCCIÓN 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. Bitácora de Log in – Log out. Bitácora de seguimiento. Concepto de parámetro. Introducción a la Criptografía. Seguridad para la integridad de datos. Administración de múltiples lenguajes. Backup. Qué consideraciones se deben tener para crear código seguro en las aplicaciones.

Tiempo: 12 hs

Unidad 3: PATRONES DE DISEÑO II.

Patrones de creación

Builder. Introducción y desarrollo para separar la construcción de un objeto complejo de su representación, de forma que el mismo proceso de construcción permita crear diferentes representaciones.

Prototype. Introducción y desarrollo para establecer la especificación de los tipos de objetos a crear por medio de una instancia prototípica. Creación de nuevos objetos copiando el prototipo.

Patrones estructurales

Bridge. Introducción y desarrollo para poder desacoplar una abstracción de su implementación para que ambas puedan variar de forma independiente.

Facade. Introducción y desarrollo para proporcionar una interfaz unificada para un conjunto de interfaces de un subsistema. Construcción de interfaces de alto nivel para facilitar el uso del sistema.

Flyweight. Introducción y desarrollo para obtener pequeños objetos compartidos utilizables en distintos contextos.

Proxy. Introducción y desarrollo para generar una estructura que presente un objeto sustituto o “representante” para acceder al objeto principal.

Patrones de comportamiento

Chain of Responsibility. Introducción y desarrollo para evitar acoplar el emisor de una petición a su receptor. Como lograr una estructura de comportamiento que permita a más de un objeto responder a una misma petición.

Command. Introducción y desarrollo para encapsular una petición en un objeto.

Parametrización de peticiones encapsuladas en objetos. Establecimiento de colas y registro de peticiones.

State. Introducción y desarrollo para lograr que un objeto cambie su comportamiento cada vez que cambie su estado interno.

Strategy. Introducción y desarrollo para Definir una familia de algoritmos, cada uno de ellos encapsulados e intercambiables.

Template Method. Introducción y desarrollo para la delegación en subclases de algunos pasos de un algoritmo sin cambiar la estructura de este.

Visitor. Introducción y desarrollo para definir nuevas operaciones sin cambiar las clases de los elementos sobre los que opera.

Concepto de Refactorización.

Tiempo: 24 hs

Unidad 4: SERIALIZACIÓN.

Conceptos básicos sobre serialización. Serialización binaria. Serialización SOAP. Atributos Serializable y NonSerialized. Gráficos de objetos. Clonación profunda de objetos.

Serialización personalizada. La interfaz ISerializable. La estructura StreamingContext. La interfaz IDeserializationCallback.

Serialización XML. La clase XmlSerializer. Atributos de Serialización. El objeto XmlSerializerNamespace. Sucesos de deserialización. Sobreescritura del comportamiento. Manejo de esquemas XSD.

Tiempo: 12 hs

Unidad 5: REFLEXIÓN.

Concepto. Utilización de los ensamblados y los módulos. La clase Assembly. La clase AssemblyName. La clase Module.

Manejo de tipos. Obtención de un objeto Type. Recuperación de atributos Type.

Enumeración de miembros. Recuperación de información de miembros. Enumeración de parámetros. Ejecución de miembros. Creación dinámica de objetos. Aspectos de seguridad a considerar.

Tiempo: 18 hs

Unidad 6: SUBPROCESAMIENTO, CONCURRENCIA, PARALELISMO E INGENIERIA DE SOFTWARE CLIENTE SERVIDOR.

Concepto. Introducción al subprocesamiento. Ventajas y desventajas del subprocesamiento. Determinación de la necesidad de utilizar subprocesos. Creación de subprocesos. Manejo de subprocesos. Propiedades de los subprocesos. Depuración de los subprocesos.

Almacenamiento y compartición de datos. Variables locales, de clases y estáticas. El atributo ThreadStatic. Almacenamiento local de subproceso. Cómo pasar y obtener datos de un subproceso.

Sincronización de subprocesos. Utilización de SyncLock. La clase Monitor. La clase Interlocked. La clase Mutex. La clase ReaderWriterLock. La clase ManualResetEvent. La clase AutoResetEvent.

La clase ThreadPool. La clase parallel.

Temporizadores. La clase System.Timers.Timer. La clase System.Threading.Timer.

Asincronismo. Operaciones asíncronas. Delegados asíncronos. Operaciones asíncronas con

archivos.

Conceptos de concurrencia y paralelismo. Multithreading Ejecución de tareas en forma asincrónica. Async. Await.

Sistemas distribuidos. Software intermedio (middleware). Arquitecturas estratificadas. Concepto de protocolo, su impacto en los desarrollos cliente – servidor. Aspectos a considerar en la construcción de sistemas cliente servidor.

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 de 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. Interconsultas. 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 indagación que incluirán la exposición oral de los mismos.

Los medios audiovisuales (Presentaciones / Webcast) 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. Comprende la finalización del proyecto de desarrollo de un sistema que se trabajó en Trabajo de Campo 1 y Metodología de Desarrollo de Sistemas II como parte

de las estrategias de articulación. Los alumnos utilizarán las 100 hs de tutoría para desarrollar las distintas etapas del proyecto. El proyecto se realizará una parte en el tiempo áulico dónde se abordarán los aspectos más representativos del trabajo y otra en el que el alumno designe fuera del tiempo de asistencia a la asignatura. Se inducirá a que el alumno desarrolle el mismo de una manera iterativa e incremental. Se dispondrán de dos puntos de control denominados entregas parciales del TP y la entrega final del mismo. El objetivo de las entregas parciales es que el alumno pueda obtener una devolución de su avance en el trabajo práctico. Los elementos que conforman al proyecto son los que se muestran en la siguiente lista.

- 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 Indagación. Los trabajos de indagación 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:

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

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

Balena, Francesco. **Programación avanzada con Microsoft Visual Basic.Net.**-- México, DF: McGraw-Hill, c2003. XXV, 1238 páginas, 1 CD-Rom

Ampliatoria:

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

Hopcroft, John E.; Motwani, Rajeev; Ullman, Jeffrey D. **Introducción a la teoría de autómatas, lenguajes y computación.** 3a. ed. -- Madrid: Pearson Educación, 2007. XVI, 440 páginas

Laudon, Kenneth C.; Laudon, Jane Price. **Administración de los sistemas de información: organización y tecnología.** 3a. ed.-- México, 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

Martin, James; Odell, James J. **Análisis y diseño orientado a objetos.**-- México, DF: Prentice Hall Hispanoamericana, c1994. 546 páginas

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

Palma Méndez, José Tomás; Garrido Carrera, María del Carmen; Sánchez Figueroa, Fernando; y otros. **Programación concurrente.**-- Madrid: Thomson Editores, 2003. XIII, 380 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, 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.

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

Pressman, Roger S. **Ingeniería del software: un enfoque práctico.** 6a.ed.-- México, DF: McGraw-Hill, c2005. XXXIV, 958 páginas

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.-- México, 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

Swartzfager, Gene; Chandak, Ramesh; Chandak, Purshottam y otros. **Visual Basic 6 : programación orientada a objetos.**-- Madrid : Paraninfo, c1999. XXVII, 454 páginas, 1 CD-Rom

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

Yourdon, Edward. **Análisis estructurado moderno.**-- México, DF: Prentice Hall Hispanoamericana, c1993. 735 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.

Los exámenes parciales y recuperatorios incluirán 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 15 (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.

La nota conceptual estará sustentada 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.

Objetivos de la evaluación.

Conocer el grado de internalización de los contenidos.
Verificar los errores y cambios conceptuales en las estructuras cognitivas de los alumnos.
Garantizar la transferencia del conocimiento.
Identificar dificultades y contradicciones en el proceso de aprendizaje.
Reconocer la integración de aprendizajes significativos.

Contenidos de la evaluación.

Se tendrán en cuenta los contenidos conceptuales, procedimentales y actitudinales.

