



**Tecnicatura Superior Análisis,
 Desarrollo y Programación de
 Aplicaciones**

Plan 2003 -Res: 6175/03

PROGRAMA DE LA MATERIA:

11321 Práctica Profesional

Area: Sistemas

Equipo Docente: Cherencio, Guillermo

Cantidad de horas semanales : 4 hs.

Días de dictado:

| | | | | | |
|---|---|---|---|---|---|
| L | M | M | J | V | S |
| | | 2 | | 2 | |

Correlatividades

Metodología de la Investigación
 Programación Orientada a Objetos
 Bases de Datos
 Análisis de Sistemas

1.0 Objetivos, Expectativas de Logro

Que el alumno sea capaz de:

1. Elaboración de proyectos de software orientados a objetos.
2. Elaboración y administración de planes y proyectos de sistemas de información.
3. Verificación de la calidad de los productos y de los procesos.
4. Resolver problemas a través de un diseño orientado a objetos
5. Implementar un diseño orientado a objetos utilizando un lenguaje orientado a objetos.
6. Representar un diseño orientado a objetos utilizando diagramas UML.
7. Implementar la persistencia de datos a través de un esquema objeto-relacional.
8. Programar triggers y store procedures en un Sistema de Base de Datos.

Contenidos mínimos:

1. Programación avanzada de Sistemas de Bases de Datos: triggers, store procedures, generators, functions, cursores.
2. Diseño Tradicional de Sistemas: Requerimientos, Análisis, Diseño, Especificación, Implementación, Prueba, Distribución, Mantenimiento.
3. Ingeniería de Software.
4. Metodologías Orientadas a Objetos: UML, RUP, XP, Ripple.
5. Diseño de Sistemas utilizando Ripple en un caso real.

Abril 2019



2.0 Fundamentación

El mercado actual exige cada día más a las organizaciones a adaptarse a las nuevas reglas de juego impuestas por el propio mercado, el estado o la competencia cada vez mayor. Esta realidad hace que los sistemas de información deban evolucionar rápidamente para que las organizaciones puedan seguir haciendo un uso efectivo de este recurso. En un primer momento, los sistemas eran herramientas utilizadas únicamente con el afán de automatizar y acelerar los circuitos administrativos vigentes, hoy en día, los sistemas informáticos pasaron a ser mucho más que eso y son un elemento vital que proporciona información resumida a niveles gerenciales para la toma de decisiones. Una empresa vale no sólo por lo bienes tangibles que ésta pueda tener, sino también por su sistema de información.

En este contexto, el desarrollo de sistemas informáticos se ha visto sumamente afectado y no podemos seguir utilizando las mismas herramientas que antaño. Los nuevos proyectos están utilizando cada día más tecnologías orientadas al desarrollo de aplicaciones orientadas a objetos. No se puede negar el impacto que han tenido plataformas como Java de Sun Microsystems aplicables desde dispositivos de mano hasta grandes sistemas distribuidos.

Se cuentan con herramientas que automatizan y aceleran gran parte del desarrollo de las aplicaciones, no obstante ello, aún es imprescindible contar con profesionales que realicen un correcto diseño orientado a objetos y luego una implementación de dicho diseño, programando cada una de las funcionalidades que tendrá el usuario a su disposición. Esta implementación debe estar acorde con ciertas reglas de un buen diseño orientado a objetos para que éste sea sustentable y mantenible en el tiempo.

En la asignatura Programación Orientada a Objetos el alumno adquirió el paradigma de la programación orientada a objetos, aquí se pretende una visión más abstracta, más global, orientada al diseño de sistemas y no orientada al desarrollo de programas puntuales.

Se pretende contrastar el enfoque tradicional de diseño de sistemas con las nuevas metodologías orientadas a objetos. De todas las metodologías existentes se ha optado por utilizar una denominada Ripple (variante de RUP) para el diseño e implementación de un sistema en particular.

Abril 2019



3.0 Programa de Estudio

- 1.0 **Metodologías de Desarrollo de Software**
 - 1.1 El enfoque tradicional: Requerimientos, Análisis, Diseño, Especificación, Implementación, Prueba, Distribución, Mantenimiento.
 - 1.2 Nuevas Metodologías: Espiral, Interactiva, Incremental.
 - 1.3 Combinando Metodologías.
 - 1.4 Metodologías Orientada a Objetos: UML, RUP, XP.
 - 1.5 Metodología Ripple. Introducción.
 - 1.6 TP I: Metodologías de Desarrollo de Software.
- 2.0 **Metodología Ripple**
 - 2.1 Introducción. Fases y Artefactos.
 - 2.2 Diagramas de Casos de Uso.
 - 2.3 Diagramas de Clases.
 - 2.4 Diagramas de Comunicaciones.
 - 2.5 Diagramas de Distribución.
 - 2.6 Diagramas de Secuencia.
 - 2.7 TP II: Metodología Ripple.
- 3.0 **Definiendo el Sistema**
 - 3.1 Obteniendo Requerimientos. Nacimiento del Sistema.
 - 3.2 Casos de Uso.
 - 3.3 Perspectiva de Negocio: Identificando actores, escribiendo glosario, casos de uso, diagramas de comunicación, diagramas de actividad.
 - 3.4 Perspectiva del Desarrollador: especializando actores, relaciones entre casos de uso, detalles de casos de uso de sistema, precondiciones, postcondiciones y herencia.
 - 3.5 TP III: Definición del Sistema.
- 4.0 **Analizando el Sistema**
 - 4.1 Introducción.
 - 4.2 Revisión del proceso de análisis.
 - 4.3 Análisis estático: diagramas de clases, de objetos, relaciones, atributos, asociaciones, objetos tangibles e intangibles
 - 4.4 Análisis dinámico: relaciones de casos de uso, límites, controladores, entidades, operaciones, responsabilidades, diagrama de máquina de estados.
 - 4.5 TP IV: Análisis Dinámico y Estático del Sistema.
- 5.0 **Diseño de la Arquitectura del Sistema**
 - 5.1 Introducción.
 - 5.2 Prioridades de Diseño. Pasos en el Diseño de Sistemas.
 - 5.3 Eligiendo una topología de red: 3-tier, www, client-server.
 - 5.4 Concurrencia.
 - 5.5 Seguridad.
 - 5.6 Particionado de Software: sistemas, subsistemas, capas, flujo de mensajes entre capas.
 - 5.7 TP V: Diseño de la Arquitectura del Sistema.



6.0 **Eligiendo la tecnología apropiada**

- 6.1 Introducción.
- 6.2 HTML/CGI.
- 6.3 RMI.
- 6.4 CORBA.
- 6.5 EJB.
- 6.6 JSP.
- 6.7 Otras tecnologías J2EE.
- 6.8 TP VI: Tecnologías a utilizar en el Sistema.**

7.0 **Diseño de Subsistemas**

- 7.1 Introducción.
- 7.2 Mapeo de operaciones, tipos de datos, visibilidad de campos, métodos de acceso, composiciones, distintos tipos de asociaciones.
- 7.3 Persistencia: mapeo objeto-relacional, mapeo de entidades, asociaciones, estado de objetos.
- 7.4 Diseño de interfaces de usuario.
- 7.5 Diseño de servicios de sistema.
- 7.6 Diseño de transacciones.
- 7.7 Diseño de múltiples actividades (multithreading).
- 7.8 Patrones de diseño reusables.
- 7.9 TP VII: Diseño de Subsistemas.**

8.0 **Programación Avanzada de Bases de Datos Relacionales**

- 8.1 Introducción. Objetos dentro de un SGBD: functions, domains, generators, tables, constraints, index, views, triggers, store procedures.
- 8.2 Implementación de scripts de base de datos. Una metodología posible.
- 8.3 Uso de variables y cursores.
- 8.4 Implementación de triggers: tipos, restricciones, ejemplos.
- 8.5 Implementación de store procedures.
- 8.6 Implementación de select procedures.
- 8.7 Combinaciones.
- 8.8 TP VIII: Implementación de BD Física.**



4.0 Bibliografía

Bibliografía Básica

- McLaughlin Brett, Pollice Gary, West David, "Head First - Object-Oriented Analysis & Design", O'Reilly Media Inc., 2006, ISBN-10:0-596-00867-8
- O'Docherty, Mike, "Object-Oriented Analysis and Design", John Wiley & Sons Ltd, The Atrium, Southern Gate, Chichester, 2005, ISBN-13 978-0-470-09240-8

Bibliografía Adicional

- Goodman, Danny, "Javascript Bible – Gold Edition", Hungry Minds, 2001, ISBN: 0-7645-4718-6
- Ball Jennifer, Carson Debbie, Evans Ian, Fordin Scott, Haase Kim, Jendrock Eric, "The Java™ EE 5 Tutorial", Sun Microsystems, Inc., 2006
- Inderjeet Singh, Beth Stearns, Mark Johnson, and the Enterprise Team, "Designing Enterprise Applications with the J2EE™ Platform, Second Edition", Addison-Wesley, Sun Microsystems, Inc., 2002, ISBN 0-201-78790-3
- Sean Brydon, Greg Murray, Vijay Ramachandran, Inderjeet Singh, Beth Stearns, Thierry Violleau, "Designing Web Services", Sun Microsystems, Inc., 2002
- Sanghera Paul, Ph.D "SCJP Exam for J2SE 5", Apress, 2006, ISBN-13 (pbk): 978-1-59059-697-5
- Eckel Bruce, "Thinking in Enterprise Java", 2003
- Philip Heller, Simon Roberts, " Complete Java® 2 Certification: Study Guide, Fifth Edition", Sybex, San Francisco, London, ISBN: 0-7821-4419-5
- Hanumant Deshmukh, Jignesh Malavia, Matthew Scarpino, "Java Web Component Developer Certification", Manning Publications Co., 2005, ISBN 1-932394-38-9

Apuntes de la Asignatura

- Cherencio, Guillermo, "Programando En Capas Usando Java", CISCO Networking Academy , UTN FRD, 2004

Otras fuentes de Información

- Java Enterprise Edition: <http://java.sun.com/javaee/>
- Java Hispano: <http://www.javahispano.org>
- Hibernate: <http://www.hibernate.org>
- Firebird: <http://www.firebirdsql.org>

Abril 2019



5.0 Evaluación

Se considerara alumnos regulares a aquellos que:

- Aprueben el 1er y 2do parcial con nota mayor igual a 4.
- Tengan aprobados todos los trabajos prácticos propuestos
- Tengan aprobado la primera versión del trabajo práctico final integrador.
- Cumplan con todos los requisitos de promoción indicados en el Plan Institucional de Evaluación, normativas de la DGyCE, DES, Pcia. Bs. As.

Evaluación Final:

- Presentación del Trabajo Práctico Final Individual Integrador (aprobado previamente por el docente en la regularización de la asignatura), software aplicativo funcionando, completo y sobre el cual se solicitarán cambios a realizar. El alumno contará con una 1 hora para implementar los cambios.

Criterio de Evaluación:

- En cuanto a los contenidos teóricos se requerirá la comprensión básica de los mismos y su correcta aplicación.
- Los alumnos tendrán a su disposición la sintaxis básica del lenguaje de programación Java, J2EE, PL/SQL, etc. no siendo este punto de interés en la evaluación.
- Es de interés para la evaluación la correcta combinación de las distintas tecnologías y metodologías requeridas para alcanzar la solución del problema propuesto
- Es de interés para la evaluación que el alumno realice un diseño orientado a objetos que sea aplicable al problema y al mismo tiempo esté acorde con las reglas de un buen diseño
- Es de interés para la evaluación el progreso que el alumno manifieste en cada etapa. La actitud frente a las dificultades, las mejoras introducidas a la aplicación entre la primera versión y la versión final de la misma.
- Es de interés para la evaluación la actitud del alumno frente al trabajo propuesto, las ganas, el interés demostrado en las actividades.
- Es de interés para la evaluación que el alumno adquiera destreza en la utilización de documentación y manuales técnicos en inglés.
- Es de interés para la evaluación que el alumno pueda integrar los conocimientos y el trabajo realizado en las asignaturas Análisis de Sistemas, Base de Datos y Programación Orientada a Objetos.

Abril 2019