



Ministerio de Cultura y Educación
 Universidad Nacional de San Luis
 Facultad de Ciencias Físico-Matemáticas y Naturales
 Departamento: Informática
 Area: Area IV: Pr. y Met. de Des. del Soft.

(Programa del año 2006)
 (Programa en trámite de aprobación)
 (Presentado el 14/03/2006 12:07:13)

I - Oferta Académica

Materia	Carrera	Plan	Año	Período
OPTATIVA(TECNOLOGIAS CASE)	LIC. CS. COMP.	01/03	5	1c

II - Equipo Docente

Docente	Función	Cargo	Dedicación
RIESCO, DANIEL EDGARDO	Prof. Responsable	P.ADJ EXC	40 Hs
MONTEJANO, GERMAN ANTONIO	Prof. Co-Responsable	P.ADJ EXC	40 Hs

III - Características del Curso

Credito Horario Semanal				
Teórico/Práctico	Teóricas	Prácticas de Aula	Práct. de lab/ camp/ Resid/ PIP, etc.	Total
11 Hs	Hs	Hs	Hs	11 Hs

Tipificación	Periodo
B - Teoria con prácticas de aula y laboratorio	1 Cuatrimestre

Duración			
Desde	Hasta	Cantidad de Semanas	Cantidad de Horas
13/03/2006	16/06/2006	14	154

IV - Fundamentación

Actualmente es imprescindible que un profesional de Informática tenga capacidad de aplicar la ingeniería de software a la propia ingeniería de software de forma de automatizar actividades asociadas al proceso de producción de software, es decir generar herramientas CASE (Ingeniería de Software Asistida por Computadora). Esta capacidad garantiza la aplicación del conocimiento necesario al desarrollo de software sofisticado. Las herramientas CASE ayudan al ingeniero de software en la producción de resultados de alta calidad. Además, disponer de automatización permite que el usuario CASE elabore resultados adicionales y personalizados que no serán fáciles ni prácticos de producir sin el soporte de las herramientas.

V - Objetivos

Habilitar al profesional en Ciencias de la Computación utilizar y producir herramientas básicas en la producción de software.

VI - Contenidos

Unidad I: CASE

¿Qué significa CASE?. Quién lo hace?. Importancia.

Construcción de Bloques Básicos para CASE.

Arquitectura de Entorno. Plataforma. Servicios de Portabilidad.

Marco de Integración.

Unidad II: Taxonomía de Herramientas CASE

Herramientas de Ingeniería de Procesos de Negocio. Modelado de Procesos y Herramientas de Gestión. Herramientas de Planificación de Proyectos. Herramientas de Análisis de Riesgo. Herramientas de Gestión de Proyectos. Herramientas de Seguimiento de Requisitos. Herramientas de Métricas y de Gestión. Herramientas de Documentación. Herramientas de Control de Calidad. Herramientas de Gestión de Configuración de Software. Herramientas de Análisis y Diseño. Herramientas PRO/SIM. Herramientas de Desarrollo y diseño de Interfaz. Herramientas de Construcción de Prototipos. Herramientas de desarrollo Web. Herramientas de Integración y Pruebas. Herramientas de Análisis Estáticos. Herramientas de Análisis Dinámico. Herramientas de Gestión de Pruebas. Herramientas de Reingeniería.

Unidad III: I-CASE

Entornos CASE Integrados. Introducción. Beneficios. Arquitectura de Integración.

Repositorio CASE. Características Especiales de los Repositorios. Versiones.
Seguimiento de Dependencias. Gestión de Cambios. Seguimiento de Requisitos.
Gestión de Configuración. Seguimiento de Auditoría.

Unidad IV: Ingeniería del Software Orientada a Objetos

Conceptos y Principios Orientados a Objetos. Modelado. Herramientas.

UML (Unified Modeling Object). Diagramas. Análisis Orientado a Objetos.
Análisis de Dominio. Modelo Objeto Relación. Modelo Objeto Comportamiento.
Proceso de Diseño de Objetos. Herramientas. Patrones de Diseño.

Unidad V: Arquitectura UML

Artefactos Primarios de UML. Definición. Desarrollo de Artefactos del Proyecto.

Porqué Modelamos. Metas UML. Arquitectura del Lenguaje. Arquitectura del Metamodelo en Capas. Meta-metamodelos.
Metamodelos. Modelos.

MOF Meta-metamodelo. Estructura de Paquetes. Formalismo del Lenguaje. Niveles de Formalismo. Estructura de Especificación de Paquetes. Sintaxis Abstracta. Reglas Bien Formadas. Semántica. Elementos Estándar. OCL (Object Constraint Language).

Unidad VI. Foundation

Núcleo. Clasificador. Asociación. Atributos. Operaciones. Mecanismos de Extensión. Estereotipos. Valores Rotulados. Restricciones. Elementos del Comportamiento. Gestión del Modelo. Metamodelo. Semánticas. Reglas de Buena Formación en OCL.

VII - Plan de Trabajos Prácticos

Los alumnos deben entregar un Diagrama de Objetos que muestre diferentes instancias de algunos Metamodelos definidos en

la arquitectura en capas.

VIII - Regimen de Aprobación

- Para regularizar la asignatura el alumno debe realizar una presentación oral y pública en base a un trabajo escrito por un autor elegido por la cátedra, aprobar un examen parcial o su correspondiente recuperación, y presentar en forma y tiempo los prácticos de aula y de laboratorio y aprobarlos.
- Para promocionar la asignatura el alumno debe cumplir con las condiciones de regularización y aprobar con nivel superior o igual a siete puntos sobre un total de diez.
- No se aceptan alumnos libres debido al trabajo práctico de laboratorio que deben realizar.

IX - Bibliografía Básica

- [1] · Pressman, R. "Software Engineering", Mc Graw Hill, 5ta edición.
- [2] · UML 1.5. Estandar definido por la OMG (Object Management Group), www.omg.org.
- [3] · Sharon, D. R. Bell, "Tools that Bind: Creating Integrated Environments". IEEE Software, Marzo de 1995.
- [4] · Brown, W, Carney, D. Morris, E. "Principles of CASE Tool Integration", Oxford University Press, 1994.
- [5] · Booch, Rumbaugh, Jacobson, et al. "UML Semantics", www.omg.gov
- [6] · Jackson, M. "Software Requirement & Specifications", Addison Wesley, 1995.
- [7] · "Software Engineering", Ian Sommerville, Addison-Wesley, 5ta Edición. 1996
- [8] · Apuntes de la Cátedra.

X - Bibliografía Complementaria

[1]

XI - Resumen de Objetivos

Actualmente es imprescindible que un profesional de Informática tenga capacidad de aplicar la ingeniería de software a la propia ingeniería de software de forma de automatizar actividades asociadas al proceso de producción de software, es decir generar herramientas CASE (Ingeniería de Software Asistida por Computadora). Esta capacidad garantiza la aplicación del conocimiento necesario al desarrollo de software sofisticado. Las herramientas CASE ayudan al ingeniero de software en la producción de resultados de alta calidad. Además, disponer de automatización permite que el usuario CASE elabore resultados adicionales y personalizados que no serán fáciles ni prácticos de producir sin el soporte de las herramientas. Se pretende en el dictado de este curso habilitar al profesional en Ciencias de la Computación utilizar y producir herramientas básicas en la producción de software.

XII - Resumen del Programa

Unidad I: CASE

¿Qué significa CASE?. Quién lo hace?. Importancia.

Construcción de Bloques Básicos para CASE.

Arquitectura de Entorno. Plataforma. Servicios de Portabilidad.

Marco de Integración.

Unidad II: Taxonomía de Herramientas CASE

Herramientas de Ingeniería de Procesos de Negocio. Modelado de Procesos y Herramientas de Gestión. Herramientas de Planificación de Proyectos. Herramientas de Análisis de Riesgo. Herramientas de Gestión de Proyectos. Herramientas de Seguimiento de Requisitos. Herramientas de Métricas y de Gestión. Herramientas de Documentación. Herramientas de Control de Calidad. Herramientas de Gestión de Configuración de Software. Herramientas de Análisis y Diseño. Herramientas PRO/SIM. Herramientas de Desarrollo y diseño de Interfaz. Herramientas de Construcción de Prototipos. Herramientas de desarrollo Web. Herramientas de Integración y Pruebas. Herramientas de Análisis Estáticos. Herramientas de Análisis Dinámico. Herramientas de Gestión de Pruebas. Herramientas de Reingeniería.

Unidad III: I-CASE

Entornos CASE Integrados. Introducción. Beneficios. Arquitectura de Integración. Repositorio CASE. Características Especiales de los Repositorios. Versiones. Seguimiento de Dependencias. Gestión de Cambios. Seguimiento de Requisitos. Gestión de Configuración. Seguimiento de Auditoría.

Unidad IV: Ingeniería del Software Orientada a Objetos

Conceptos y Principios Orientados a Objetos. Modelado. Herramientas. UML (Unified Modeling Object). Diagramas. Análisis Orientado a Objetos. Análisis de Dominio. Modelo Objeto Relación. Modelo Objeto Comportamiento. Proceso de Diseño de Objetos. Herramientas. Patrones de Diseño.

Unidad V: Arquitectura UML

Artefactos Primarios de UML. Definición. Desarrollo de Artefactos del Proyecto. Porqué Modelamos. Metas UML. Arquitectura del Lenguaje. Arquitectura del Metamodelo en Capas. Meta-metamodelos. Metamodelos. Modelos. MOF Meta-metamodelo. Estructura de Paquetes. Formalismo del Lenguaje. Niveles de Formalismo. Estructura de Especificación de Paquetes. Sintaxis Abstracta. Reglas Bien Formadas. Semántica. Elementos Estándar. OCL (Object Constraint Language).

Unidad VI. Foundation

Núcleo. Clasificador. Asociación. Atributos. Operaciones. Mecanismos de Extensión. Estereotipos. Valores Rotulados. Restricciones. Elementos del Comportamiento. Gestión del Modelo. Metamodelo. Semánticas. Reglas de Buena Formación en OCL.

XIII - Imprevistos

ELEVACIÓN y APROBACIÓN DE ESTE PROGRAMA**Profesor Responsable**

Firma:

Aclaración:

Fecha: