



Universidad Nacional Experimental  
de los Llanos Occidentales  
"Ezequiel Zamora"

CONTENIDO PROGRAMÁTICO DEL SUBPROYECTO:

## PRINCIPIOS DE INGENIERÍA DEL SOFTWARE

**VICERRECTORADO:** Planificación y Desarrollo Social  
**PROGRAMA:** Ingeniería, Arquitectura y Tecnología  
**SUBPROGRAMA:** Ingeniería en Informática  
**CARRERA:** Ingeniería en Informática  
**ÁREA DE**  
**CONOCIMIENTO:** Formación Profesional Específica  
**PROYECTO:** Desarrollo de Software  
**CÓDIGO:** II51002070601  
**PRELACIÓN:** Bases de Datos I y Algoritmos y Programación II  
**HORAS SEMANALES:** 6 Horas: 4 Horas Teóricas y 2 Horas Prácticas  
**UNIDADES CRÉDITO:** 05  
**SEMESTRE:** VI  
**CONDICIÓN:** Obligatoria  
**PERFIL DEL**  
**DOCENTE:** Ingeniero en Informática, Ingeniero de Sistemas, Licenciado en  
Computación, Ingeniero en Información, Licenciado en Informática,  
Ingeniero de Sistemas, Ingeniero en Computación, o afines  
**PROFESOR(ES)**  
**DISEÑADOR(ES):** Prof. José Rafael Quintero A.

**Barinas, Julio 2008**

## JUSTIFICACIÓN

Los grandes sistemas software constituyen actualmente un elemento común en nuestra sociedad, convirtiéndose día a día en imprescindibles para la industria, el comercio, la banca, las finanzas y la vida personal en general. La producción del software, como cualquier otro producto industrial, necesita de la aplicación de los conceptos, técnicas, teorías y métodos de la ingeniería que posibiliten el desarrollo de sistemas software de calidad y de aplicación efectiva dentro de los procesos de la actividad humana; de allí lo fundamental que es para el ingeniero en informática conocer y manejar los principios, normalmente aceptados, para el desarrollo de software.

Luego de adquirir los elementos teóricos básicos sobre la ingeniería de software, los estudiantes deben conocer temas relacionados con el proceso del software, los fundamentos de los métodos convencionales y de los métodos orientados a objeto de la ingeniería del software; además, es indispensable profundizar sus conocimientos en la construcción práctica de modelos con el lenguaje unificado de modelado.

## OBJETIVO GENERAL

Al finalizar el sub-proyecto, el estudiante estará en capacidad de manejar los conceptos básicos sobre la ingeniería del software; podrá definir y contextualizar los procesos del software incluyendo el proceso Unificado de Rational (RUP) su definición y características; utilizará el lenguaje unificado de modelado - UML como una herramienta dentro de la orientación a objeto de la ingeniería de software. Conocerá y manejará la ingeniería de requisitos y las herramientas estructuradas como bases de los métodos convencionales de la ingeniería del

software, manejará los elementos básicos para la verificación y prueba del software.

## MÓDULO I: INTRODUCCIÓN A LA INGENIERÍA DEL SOFTWARE

Valor 15%

### **Objetivo Específico**

Conocer los conceptos básicos relacionados con la ingeniería del software.

### **Contenido**

Software: definición y características, evolución del software hacia la ingeniería.

Progreso de las técnicas en la construcción de software. Crisis del software: causas, síntomas, consecuencias, planteamiento de soluciones.

Definición de la Ingeniería del Software, objetivos de la Ingeniería del Software, disciplinas que componen la Ingeniería del Software.

Fundamentos y técnicas de apoyo a la Ingeniería del Software.

## MÓDULO II: LOS PROCESOS DEL SOFTWARE

Valor 20%

### Objetivos Específicos

1. Conocer y manejar las actividades necesarias para la producción de software.
2. Entender el concepto de procesos del software y sus modelos asociados.

### Contenido

Definición, importancia y características del proceso de software.

Los modelos de procesos del software definiciones y característica para: el modelo de construcción de prototipos, el modelo en cascada y en cascada modificado, el modelo RAD, los modelos evolutivos, el modelo basado en componentes, el modelo de los métodos formales, técnicas de cuarta generación.

Actividades del proceso de software, definiciones de: especificación de requisitos del software, diseño e implementación del software, validación y evolución del software.

El proceso Unificado de Rational (RUP): definición y características de RUP, elementos y fases, las 6 reglas para la buena práctica en la ingeniería del software, elementos estáticos y dinámicos.

## MÓDULO III: INGENIERÍA DE REQUISITOS

Valor 20%

### Objetivos Específicos

1. Presentar al estudiante los elementos básicos sobre la ingeniería de requisitos.
2. Conocer y comprender las principales actividades implicadas en la ingeniería de requisitos para la ingeniería del software.

### Contenido

Definición, objetivos e importancia de los requisitos de software.

Definición y características para: requisitos funcionales y no funcionales, requisitos del usuario, requisitos del sistema.

Obtención y análisis de requisitos: uso de entrevistas, escenarios y casos de uso.

Técnicas estructuradas para el modelado y la representación de requisitos: diagramas de flujo, de control, de contexto, de estado, de transición de estados, diagramas de entidad - relación y diccionario de datos.

## MÓDULO IV: INTRODUCCIÓN A LA VERIFICACIÓN Y PRUEBAS DEL SOFTWARE

Valor 20%

### Objetivos Específicos

1. Presentar al estudiante las técnicas de verificación de software.
2. Conocer y manejar los conceptos relativos a las pruebas del software.
3. Comprender los elementos necesarios para generar casos de pruebas del software.

### Contenido

Objetivo de la verificación de software, planificación y características de la verificación.

Pruebas del software: objetivos y principios.

Técnicas y diseño de casos de prueba: diseño de casos de pruebas de la caja blanca, prueba del camino básico, prueba de condiciones, prueba de bucles, diseño de casos de pruebas de la caja negra, partición equivalente, análisis de valores límites, pruebas de las interfaces, pruebas de la documentación.

Estrategias de pruebas del software, definición y características para: prueba de unidad, prueba de integración, prueba de validación y prueba del sistema.

## MÓDULO V: ANÁLISIS ORIENTADO A OBJETOS: EL LENGUAJE UNIFICADO DE MODELADO (UML)

Valor 25%

### Objetivos Específicos

1. Definir los elementos que conforman el lenguaje unificado de modelado - UML.
2. Entender la sintaxis del UML y utilizar sus diversos componentes para el proceso de desarrollo de software.
3. Presentar casos prácticos de modelado con UML.

### Contenido

Introducción al UML: definición, ámbito y concepción del lenguaje.

Introducción a los casos de uso: definición, importancia, inclusión y extensión de los casos de uso.

Diagramas de casos de uso: representación y concepción de las relaciones.

Definición, características y funcionalidad de: Diagramas de estados, diagramas de secuencias, diagramas de colaboración, diagramas de actividades, diagramas de componentes, diagramas de distribución.

Presentación de casos teórico - prácticos de modelado en UML, desarrollo de ejemplos de casos prácticos de modelado con UML.



## ESTRATEGIAS INSTRUCCIONALES

- Exposiciones orales por parte del profesor.
- Talleres de ejercicios.
- Exposición de casos para el análisis.

## EVALUACION

Módulos	Tipo de Evaluación
I	Prueba Escrita
II	Prueba Escrita
III	Prueba Escrita
IV	Prueba Escrita
V	Prueba Teórico - Práctica

## BIBLIOGRAFÍA

- **KIMMEL Peter**, Manual de UML, Ed. McGraw Hill Interamericana, 2ra edición, México 2006.
- **KRUCHTEN Philippe**, "The Rational Unified Process", Ed. Addison - Wesley, 3ra Edición, USA 2003.
- **PRESSMAN Roger**, "Ingeniería del Software, un Enfoque Práctico", Ed. McGraw Hill, 5ta edición, España 2002.
- **SCHACH Sue**, "Ingeniería de Software Clásica y Orientada a Objetos", Ed. McGraw Hill Interamericana, 3ra edición, México 2006.
- **SOMMERVILLE Ian**, "Ingeniería del Software", Ed. Addison Wesley, 7ma edición, España 2005.
- **WEITZENFELD Alfredo**, "Ingeniería de Software Orientada a Objetos con UML, Java e Internet", Ed. Thomson, USA 2005.