



**UNIVERSIDAD NACIONAL DE INGENIERÍA
FACULTAD DE CIENCIAS
ESCUELA PROFESIONAL DE FÍSICA**

SÍLABO

INFORMACIÓN GENERAL

ASIGNATURA	:	INGENIERIA DE SOFTWARE I
CÓDIGO	:	CC472
CRÉDITOS	:	04 (CUATRO)
PRE-REQUISITO	:	CC302 – LENGUAJE PARALELO DISTRIBUIDO ORIENTADO A OBJETOS
CONDICIÓN	:	OBLIGATORIO
HORAS POR SEMANA	:	06 (TEORÍA: 03, LABORATORIO: 03)
SISTEMA DE EVALUACIÓN	:	G

OBJETIVO

Familiarizar al alumno con los procesos de software que se presentan en el desarrollo del ciclo de vida del software.

Presentar a los alumnos los diferentes modelos de evaluación de procesos y las métricas del proceso de software

Los alumnos deben identificar los requerimientos Funcionales y no Funcionales de la construcción o de un software

Orientar a los alumnos en el modelamiento de análisis de requerimientos y prototipos de un a software.

Los alumnos debe ser capaces de seleccionar y aplicar patrones de diseño apropiados en la construcción de una aplicación de software

Aplicar el diseño de componentes y el diseño de reuso en las aplicaciones presentadas por los alumnos.

PROGRAMA ANALÍTICO

1. Procesos de Software

I: Ciclo de vida del software y modelos de procesos. II: Modelos de evaluación de procesos. III: Métricas del proceso de software.

2. Representación Avanzada del Conocimiento y Razonamiento

I: Elicitación de requerimientos. II: Técnicas de modelamiento del análisis de requerimientos. III: Requerimientos Funcionales y no Funcionales. IV: Prototipeo. V: Conceptos básicos de técnicas de especificación formal.

3. Tópicos Fundamentales en Sistemas Inteligentes

I: Conceptos fundamentales de diseño y principios. II: Patrones de diseño. III: Arquitectura de software. IV: Diseño estructural. V: Análisis y diseño orientado a objetos. VI: Diseño a nivel componente. VII: Diseño para reuso.

BIBLIOGRAFÍA

1. Blum, B. I. Software Engineering: A Holistic View. Oxford University Press US, 7th edition., 1992
2. Pressman, R. S. Software Engineering: A Practitioner's Approach. McGrawHill, 6th edition. 2004.
3. Schach, S. R. Object-Oriented and Classical Software Engineering. McGrawHill., 2004.