

UNIVERSIDAD DE BUENOS AIRES

FACULTAD DE CIENCIAS EXACTAS Y NATURALES

DEPARTAMENTO: Computación.....

ASIGNATURA: ~~.....~~.....

CARRERA/S:..Licenciatura en Cs. de la Computación...(Plan 1987)..  
.orientación informática.....

CARACTER:..optativa.....(indicar si es obligatoria u optativa)

PUNTAJE:..4.....(en caso de ser optativa)

DURACION DE LA MATERIA:.....(indicar si es cuatri-  
mestral o anual).

HORAS DE CLASE: a) TEORICAS..6... HS. b) PROBLEMAS ..... HS.  
c) LABORATORIO... HS. d) SEMINARIOS..... HS.  
e) TOTALES..6.... HS.

ASIGNATURAS CORRELATIVAS:..Especificaciones y diseño de software.  
Análisis de sistemas.....

Objetivos generales:

- que al finalizar el curso, el asistente sea capaz de:
  - conocer los términos básicos, las definiciones y las técnicas de la Ingeniería de software,
  - Aplicar modelos de la IS a organizaciones arbitrarias para diagnosticar su madurez tecnológica,
  - elaborar un plan para instalar las técnicas de la IS en una organización,
  - diseñar métricas adecuadas para medir calidad y estimar variables de proyecto de un proceso de desarrollo de software,
  - calcular variables económicas de un proyecto de desarrollo de software.

-En función de esos objetivos generales, se proponen los siguientes módulos:

Módulo 1: definición y justificación de la ingeniería de software.

Objetivos: al finalizar el módulo el asistente estará capacitado para: comprender la necesidad de una Ingeniería de software, conocer la definición de Ingeniería de software, comprender el significado económico de la Ingeniería de software.

Objetivos específicos: al finalizar el módulo, el alumno estará capacitado para:

dado un proyecto de desarrollo de software, entender las necesidades económicas que lo motivan,

dada la situación problemática que plantea el desarrollo de software, conocer el valor de las soluciones estructuradas.

Contenido: definición de Boehm de la ingeniería de software. Estadísticas de costos, demanda, mantenimiento y efectividad en el desarrollo de software. La crisis del software.

Módulo 2: sistemas y software en las organizaciones.

Objetivos: al finalizar el módulo el asistente estrá capacitado para: comprender el porqué y el para qué de la IS, distinguir entre hacer software y hacer programas, evaluar el rol de la IS en las organizaciones.

Objetivos específicos: al finalizar el módulo, el alumno estará capacitado para:

dada una organización, reconocer sus subsistemas desde el punto de vista de la Ingeniería de software,

dado un sistema de información de una organización dada, reconocer la relación entre los datos y la información,

dado un sistema de información de una organización dada, reconocer sus componentes formales e informales,  
dada una organización cualquiera, analizar su madurez tecnológica,

dado un proyecto de sistemas reconocer la relación entre éste y el proyecto de software,

dado un proyecto de software, reconocer las etapas que lo componen, según distintos modelos que se pueden aplicar.  
Contenido: propósito y definición de software. Sistemas producto. Los sistemas en las organizaciones: componentes. Ciclo de información. Información y datos. Modelo de crecimiento de Nolan y Norton. Modelo de Madurez tecnológica de Humphrey. El proyecto de sistemas y el proyecto de software. Modelos del proceso de desarrollo de software: Code-and-Fix, Royce, Metzger, Yourdon, Balzer, Connel y Schafer, Boehm.

### Módulo 3: Métricas en software.

Objetivos: al finalizar el módulo el asistente estará capacitado para construir métricas de software, evaluar métricas de software.

Objetivos específicos: al finalizar el módulo, el alumno estará capacitado para:

dado un proyecto de software bajo un modelo del proceso de desarrollo, determinar métricas adecuadas de estimación de resultados,

dado un modelo de proceso de desarrollo con técnicas estructuradas de desarrollo, determinar métricas de calidad adecuadas,

dada una organización de desarrollo de software, sugerir la estructura organizativa que permita el aprovechamiento de las métricas,

dadas ciertas métricas previamente seleccionadas por el Ingeniero de software y los valores observados para ellas, determinar el valor de los resultados según algún modelo econométrico.

Contenido: Definición de métrica. Características deseables de una métrica. Uso de las métricas. Proyección de resultados. Medición de la calidad de la estimación. Personal afectado a la estimación. Actividades relacionadas. Métricas de volumen. Métricas de estructuras de datos. Métricas de lógica. Ciencia de software. Métricas de esfuerzo y costo. Micro-modelos de esfuerzo: Halstead, Woodfield, modularidad, Basili, Gordon, Curtis. Macro-Modelos: Walston Felix, Cocomo, Softcost, Copmo.

### Módulo 4: Construcción de modelos de sistema.

Objetivos: al finalizar el módulo el asistente estará capacitado para: comprender la necesidad y utilidad de los modelos,

comprender las ventajas y desventajas de los lenguajes textuales y gráficos,

conocer ASML y sus herramientas de descripción de modelos.

Objetivos específicos: al finalizar el módulo, el alumno estará capacitado para:

dado un proyecto de desarrollo de software encarado con ASML, describir la secuencia de construcción de una especificación estructurada completa, enumerando los modelos que la componen y señalando la relación entre ellos,

dado un modelo particular que forma parte de la documentación de un proyecto de desarrollo de software, describir la secuencia de construcción del mismo, enumerando los documentos que lo componen y señalando la relación entre ellos,

dado un modelo particular que forma parte de la documentación de un proyecto de desarrollo de software, describir la secuencia de construcción del mismo, enumerando los documentos que lo componen y señalando la relación entre ellos,

dada una lista de eventos, conocer su función en la construcción del modelo,

dado un diagrama de contexto, describir sus componentes y su función

dado un diagrama de flujo de datos, nombrar sus componentes e identificar sus funciones, evaluando su calidad,

dado un diagrama de entidades y relaciones, enumerar sus partes, nombrar sus funciones y evaluar su calidad,

dada una miniespecificación por pre y post condición, explicar su función en el modelo e interpretar su significado,

dado un diccionario de datos, explicar su uso en la construcción de un modelo y evaluar su calidad,

dado un diagrama de transición de estados, expresar los cambios de estado que simboliza y evaluar su calidad,

dado una carta de estructura, explicar su uso, interpretar su contenido y evaluar su calidad.

Contenido: modelos y sistemas. Modelos gráficos y modelos de texto. Herramientas, lista de eventos. Diagramas del contexto. Diccionario de datos. Diagramas de entidades y relaciones. Diagramas de flujo de datos. Explosiones de nivel. Miniespecificaciones. Cartas de estructura. Cohesión y acoplamiento. Diagramas de transición de estados. Hojas de ruta de pantallas.

#### Módulo 5: Actividades del relevamiento.

Objetivos: Al finalizar el módulo el asistente estará capacitado para:

comprender el rol del relevamiento en un proyecto de software,  
construir los documentos propios del relevamiento.

Objetivos específicos: al finalizar el módulo, el alumno estará capacitado para:

dado un proyecto de software, establecer los alcances del sistema,

dado un proyecto de software, fijar los beneficios esperados,

dado un proyecto de software, definir los usuarios esenciales,

dado un proyecto de software, definir los objetivos funcionales, de eficiencia, de seguridad y de mantenimiento.

Contenido: Las primeras actividades. Modelo del ambiente. Alcances. Usuarios. Beneficios esperados. Determinación de objetivos. Funcionales, de seguridad, de eficiencia y de mantenimiento.

#### Módulo 6: Factibilidad del proyecto.

Objetivos: al finalizar el módulo el asistente estará capacitado para:

establecer la factibilidad técnica de un proyecto de software,  
establecer la factibilidad operativa de un proyecto de software,

establecer el riesgo de un proyecto de software,  
analizar financieramente el proyecto de software.

Objetivos específicos: al finalizar el módulo, el alumno estará capacitado para:

dado un proyecto de software, identificar los procesos que puedan resultar en problemas no computables,

dado un proyecto de software, identificar los procesos que puedan resultar en problemas de complejidad exponencial,

dado un proyecto de software, identificar los problemas operativos que puedan derivarse de su instalación,

dado un proyecto de software, establecer los parámetros micro-económicos del mismo,

dado un proyecto de software, analizar el riesgo que entraña encararlo, establecer si ese riesgo es aceptable y determinar las medidas que permiten conducir el proyecto manteniendo el riesgo bajo control.

Contenido: computabilidad. Problema de la parada. Complejidad algorítmica. Problemas duros. Operatividad. Impacto del software sobre el sistema global. Análisis financiero. Valor presente. Tasa de retorno. Flujo de caja. Vida útil. Retorno de la inversión. Período de repago. Concepto de portfolio. Definición de riesgo. La gerencia de proyectos dirigidos por el riesgo. Variables que intervienen en el riesgo. Modelo de Mac. Farlane. Modelo de Boehm. Medidas de control.

Metodología: El curso se dictará en clases teóricas semanales de dos horas, seguidas de el estudio de un caso. Los alumnos deberán completar el material del mismo como parte de la evaluación. El examen final consistirá en preguntas relacionadas con el desarrollo del caso. Se habilitarán turnos de consulta para evacuar dudas resultantes de la resolución de los problemas del caso.

Bibliografía:

Los Clásicos

- Block, "The politic of projects",
- Boddie, "Crunch Model: Building Effective System on a Tight Schedule",
- Boehm, "Software Engineering Economics",
- Brooks, "The Mythical Man-Month",
- De Marco, "Peopleware",
- Grindley & Humble, "The Effective Computer: an MBO Approach",
- Nolan, "Managing the Data Resource Function",
- Yourdon, "Managing the System Life-Cycle",

Novedades:

- Pressman, "Making Software Engineering Happen",

Prototipación:

- Connel & Schafer, "Structured Rapid Prototyping",
- Wood & Silver, "Joint Application Design",

Métricas:

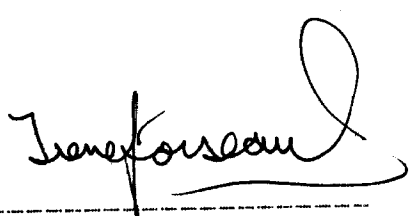
- Conte, Dunsmore & Shen, "Software Engineering Metrics and Techniques",
- De Marco, "Controlling Software Projects",
- Grady & Caswell, "Software Metrics: Establishing a company-wide Program",

Plan de contingencias:

- Toigo, "Disaster Recovery Planning",

FECHA: 1/10/91

  
-----  
DOCENTE RESPONSABLE  
ING. JORGE BORJA

  
-----  
AUTORIDAD DEPARTAMENTAL  
I. LOISEAU