



Nombre de la materia:	Ingeniería de Programación
Clave:	CI7101-T
No. de horas/semana:	4
Total de horas:	64
No. de créditos:	8
Prerrequisitos:	Programación de Computadoras (CI0000-T)

Objetivo general: Este curso presenta un estudio profundo de muchos de los temas de ingeniería de software, incluyendo paradigmas de ingeniería de software, especificación de requisitos, diseño funcional y orientación de objetos, verificación y mantenimiento de software. Se discuten ambientes y herramientas de software y se introduce a los estudiantes a su uso. Se consideran también implicaciones sociales tales como el costo de las fallas y las responsabilidades profesionales. El curso también comprende una introducción a la interacción computadora - humano y una discusión de algunas interfaces vistas desde la perspectiva del software. Este curso requiere de la participación de los estudiantes en proyectos por equipos.

Programa sintético

1. Introducción: Conceptos a tratar	6 hrs.
2. La naturaleza y cualidades del software	6 hrs.
3. Principios de la Ingeniería de Software	4 hrs.
4. Diseño de software	12 hrs.
5. Especificación de requisitos de software	12 hrs.
6. Verificación y Validación	14 hrs.
7. Herramientas y ambientes para la ingeniería de software	8 hrs.
Total: 62 hrs.	

Programa desarrollado

1. Introducción: Conceptos a tratar	6 hrs.
1.1 Objetivos del diseño de software	
1.2 Documentación	
1.3 Control y administración de la configuración	
1.4 Aspectos de la confiabilidad del software: seguridad, responsabilidad, estimación del riesgo	
1.5 Mantenimiento	
1.6 Herramientas de especificación y diseño, herramientas de implementación	
2. La naturaleza y cualidades del software	6 hrs.



- 2.1 Clasificación de las distintas cualidades relativas al software
- 2.2 Cualidades representativas
- 2.3 Cualidades necesarias en distintas Áreas de aplicación
- 2.4 Medición de la calidad
- 3. Principios de la Ingeniería de Software 4 hrs.
 - 3.1 Rigor y formalismo
 - 3.2 Separación de problemas (concerns)
 - 3.3 Modularidad
 - 3.4 Abstracción
 - 3.5 Previsión para el cambio
 - 3.6 Generalidad
 - 3.7 Incrementabilidad
- 4. Diseño de software 12 hrs.
 - 4.1 La actividad de diseño y sus objetivos.
 - 4.2 Técnicas para la modularización
 - 4.3 Diseño funcional/orientado al proceso
 - 4.4 Diseño desde abajo; apoyo para la reutilización
 - 4.5 Estrategias de implementación (desde arriba, desde abajo, desarrollos en equipo)
 - 4.6 Tópicos de implementación; mejoría en el desempeño, depuración, prevención de problemas
 - 4.7 Diseño orientado a objetos
 - 4.8 Manejo de anomalías
 - 4.9 Software concurrente
- 5. Especificación de requisitos de software 12 hrs.
 - 5.1 El uso de las especificaciones
 - 5.2 Cualidades de las especificaciones
 - 5.3 Clasificación de estilos de especificaciones
 - 5.4 Especificaciones y verificación
 - 5.5 Especificaciones operativas
 - 5.6 Especificaciones descriptivas
 - 5.7 Construcción y uso de especificaciones en la práctica
- 6. Verificación y Validación 14 hrs.
 - 6.1 Metas y requisitos para la verificación
 - 6.2 Enfoques para la verificación
 - 6.3 Pruebas (Generación de un plan de pruebas, aceptación de las pruebas, pruebas por unidad, pruebas de integración, pruebas de regresión)



- 6.4 Análisis
- 6.5 Ejecución simbólica
- 6.6 Depuración
- 6.7 Otros aspectos: Desempeño, robustez, métricas del software
- 7. Herramientas y ambientes para la ingeniería de software 8 hrs.
 - 7.1 Desarrollo histórico de herramientas y ambientes
 - 7.2 Clasificación de herramientas y ambientes de software
 - 7.3 Herramientas más representativas: editores, ligadores, intérpretes, generadores de código, depuradores, herramientas para pruebas, analizadores estáticos, herramientas para el manejo de interfaces, configuración y administración; infraestructura para la ingeniería de software.
 - 7.4 El papel del lenguaje de programación en el ambiente
 - 7.5 Algunos ejemplos de herramientas
 - 7.6 Perspectiva

Bibliografía básica:

- Ghezzi, C.; Jazayeri, M.; Mandrioli, D., Fundamentals of Software Engineering, Prentice-Hall Inc., 1991.

Bibliografía complementaria:

- Ledgard, H.; Tauer, J., Professional Software, Software Engineering Concepts, Vol. 1, Addison-Wesley Publishing company, 1988.
- Gilb, T., Principles of Software Engineering Management, Addison-Wesley Publishing Company, 1988.
- Glass, R.L., Software Conflict, Essays on the Art and Science of Software Engineering, Yourdon Press Computing Series, 1991.
- Hrietz, J.; Jackson, K., Computer Security Solutions, Blackwell Scientific Publications, 1990.
- Naur, P., Computing: A Human Activity, ACM Press, Addison Wesley Publishing Company, 1992.
- Schulmeyer, G. G., Zero Defect Software, McGraw-Hill, Inc., Software Engineering Series, 1990.
- Thimbleby, H., User Interface Design, ACM Press, Addison-Wesley Publishing Company, 1990.
- Nutt, G.J., Centralized and Distributed Operating Systems, Prentice-Hall, 1991.
- Klerer, M., Design of Very High-Level Computer Languages, A User-Oriented Approach, 2nd. Edition, McGraw-Hill, Inc., 1991.
- Tracz, W.; Editor, Software Reuse-Emerging Technology, IEEE Computer Society Press, 1988.

Metodologías de enseñanza-aprendizaje:



Metodologías de evaluación:

