**Facultad de Ingeniería y Ciencias  
Escuela de Informática y Telecomunicaciones**

**PROGRAMA DE ASIGNATURA**Fundamentos de Ingeniería y Arquitectura de Software

1. **Identificación**
   * Nombre : Fundamentos de Ingeniería y Arquitectura de Software
   * Código :
   * Créditos : 6
   * Duración : Semestral
   * Ubicación en plan de estudio: Semestre 2
   * Requisitos :
   * Sesiones semanales : 2 cátedras, 1 ayudantía
2. **Descripción del curso**

Este curso entrega los fundamentos necesarios para el diseño, evaluación y optimización de soluciones de software a través de metodologías de ingeniería de software y principios de arquitectura moderna. Se abordan técnicas de recolección de requerimientos, diseño modular, patrones arquitectónicos y tecnologías como microservicios y contenedores. El curso culmina con un proyecto colaborativo aplicado, reforzando las competencias necesarias para enfrentar desafíos reales en la industria del software.

1. **Resultados de aprendizaje**

Al finalizar el curso el alumno será capaz de:

* Aplicar metodologías modernas para el análisis, diseño y evaluación de software.
* Diseñar arquitecturas de software escalables y seguras basadas en principios de calidad.
* Formular y analizar requerimientos utilizando herramientas actuales.
* Desplegar soluciones de software siguiendo buenas prácticas, a partir de una correcta documentación e implementación.
* Participar en proyectos colaborativos aplicando principios de arquitectura y diseño.

1. **Unidades Temáticas**
2. ***Introducción a la Ingeniería y Arquitectura de Software***
   1. ***Importancia de la arquitectura en el ciclo de vida del software***
   2. ***Rol del/la ingeniero/a de software en equipos modernos***
3. ***Ingeniería de Requerimientos***
   1. ***Requerimientos funcionales y no funcionales***
   2. ***Historias de usuario y criterios de aceptación***
   3. ***Técnicas de análisis y priorización***
4. ***Estilos y Principios de Arquitectura de Software***
   1. ***Arquitecturas en capas, cliente/servidor, orientada a servicios (SOA)***
   2. ***Atributos de calidad: escalabilidad, seguridad, rendimiento, mantenibilidad***
   3. ***Principios de diseño arquitectónico***
5. ***Diseño Modular y APIs***
   1. ***Descomposición de sistemas en módulos cohesivos***
   2. ***Definición de interfaces claras y uso de APIs***
6. ***Patrones de Diseño y Arquitectura***
   1. ***MVC, Microkernel, Pipeline***
   2. ***Patrones comunes y antipatrones***
7. ***Introducción a Microservicios y Contenedores***
   1. ***Comparación con arquitectura monolítica***
   2. ***Contenedores y orquestación***
8. ***Evaluación, Planificación y Análisis de Brechas***
   1. ***Evaluación integral del proyecto: revisión de objetivos, procesos, requerimientos y soluciones actuales.***
   2. ***Identificación de brechas entre el estado actual y el estado deseado del sistema y del proyecto.***
   3. ***Planificación de mejoras: arquitectura, procesos, tecnología, organización***
9. **Metodología**

Las clases se imparten en modalidad teórico-práctica. Se espera que el alumno adquiera los conocimientos técnicos y metodológicos necesarios mediante clases expositivas, y su aplicación a problemas realistas. Las clases presentan los principios, métodos y técnicas utilizadas en Ingeniería de Software. Para poner en práctica los conceptos aprendidos, se realizará un proyecto semestral en contextos reales de manera colaborativa con equipos formados por los alumnos.

A partir de las actividades antes mencionadas se desarrollará la capacidad para modelar formalmente sistemas informáticos. La disciplina de Ingeniería de Software tiene una relación estrecha con la habilidad de un ingeniero para reducir los riesgos del software. Planificar, analizar y diseñar son actividades orientadas a esos aspectos y serán reforzadas/evaluadas durante la ejecución del proyecto antes descrito y en las evaluaciones formales del curso.

1. **Evaluación**

Se contempla la realización de evaluaciones parciales (controles y trabajos), un proyecto semestral, dos pruebas solemnes de igual valor y un examen final. Si la nota asociada a la ejecución del proyecto semestral es inferior a 4.0, el alumno reprobará la asignatura con nota final igual a la nota obtenida en el proyecto. En caso contrario, la nota de presentación se calcula a través de la siguiente formula.

Si promedio solemnes es mayor o igual a 4.0:

Nota de Presentación = (20% Solemne 1 + 20% Solemne 2 + 20% Proyecto+ 10% Parciales)/0.7

Si promedio solemnes es menor a 4.0:

Nota de Presentación = (30% Solemne 1 + 30% Solemne 2 + 10% Parciales)/0.7

Nota Final= 70% Nota de Presentación + 30% Examen.

Podrán eximirse aquellos alumnos que cumplan con los siguientes requisitos:

1. Nota de presentación mayor o igual a 5.0.
2. Solemne 1, Solemne 2 y Proyecto mayor o igual 4.0.
3. Todas las evaluaciones rendidas, incluyendo las dos solemnes, las 3 entregas del proyecto y todas las evaluaciones parciales.
4. **Bibliografía Básica**

 Sommerville, Ian. *Ingeniería de Software*, 10ª ed., Prentice-Hall, 2015.

 Pressman, Roger. *Ingeniería del Software: Enfoque práctico*, 8ª ed., McGraw Hill, 2014.

 Gamma, E. et al. *Patrones de diseño*, Addison-Wesley.

 Craig Larman. *UML y Patrones*, 3ª ed.

 Apuntes y recursos del curso disponibles en línea.



Elaborado por: Jonathan Frez

Fecha revisión: Marzo de 2025

Fecha vigencia: Marzo de 2026