



Planificaciones

7510 - Técnicas de Diseño

Docente responsable: PANTALEO GUILLERMO GUSTAVO

OBJETIVOS

En este curso se busca introducir a los alumnos en el concepto de diseño de software. Para lograrlo se presentan además de técnicas de diseño, los aspectos claves que determinan la validez de las mismas como solución a un problema de implementación. Se analiza el contexto en el que se deben aplicar las técnicas y se fijan criterios de selección. Se estudia la evolución del software en el tiempo, lo que hace que algunas de las técnicas que en determinado momento fueron válidas, hoy ya no lo sean y en cambio otras continúen vigentes. Se introduce el concepto de arquitectura de software, como marco a todas las actividades del diseño de software de un sistema en desarrollo.

Ayudar a los estudiantes a validar sus objetivos de estudio para generar las condiciones necesarias de esfuerzo y motivación que faciliten la aprobación de la materia.

Ordenar y relacionar conocimientos de diseño de software que los alumnos traen de estudios anteriores para evitar que se sacrifique el concepto de "interfaz" y "herencia" y abrir la cabeza a todos los conceptos del diseño de software.

Proveer a los alumnos de herramientas que contribuyan a formar un criterio que les permita solucionar problemas de diseño de software para que lo utilicen cuando lo necesiten y lo perfeccionen con otros estudios y con la experiencia.

Entrenar a los alumnos en el proceso de resolver problemas de diseño de software a partir de una clara comprensión del problema planteado y una solución derivada del análisis del mismo; evitando así sistemas sobredimensionados por el uso injustificado de herramientas.

CONTENIDOS MÍNIMOS

PROGRAMA SINTÉTICO

Conceptos, paradigmas y modelos de programación. Evolución histórica de los modelos de programación y el concepto de diseño.

Enfoque lógico, funcional, orientado a objetos y por prototipos. Diseño de software en modelos declarativos e imperativos.

Criterios de buen diseño.

Lenguajes procedurales. Módulos, interfaces y funciones. Cohesión y acoplamiento. Eficiencia de procesamiento como restricción de diseño.

Lenguajes orientados a objetos. Inversión en la cadena de dependencia, criterio de substitución de Liskov, código clausurado ante cambios, segregación de interfaces, simple responsabilidad. Ley de Demeter.

El proceso de diseño. Requerimientos funcionales y no funcionales como conductores y restricciones de diseño.

Entender el problema, patrones de colaboración. Modelo de dominio.

Implementar una solución, patrones de diseño. Modelo de diseño.

Diseño conducido por el dominio. Diseño conducido por las pruebas.

Arquitectura de software, conceptos, vistas y enfoques. Patrones de arquitectura. Sistemas de procesamiento de información, sistemas de tiempo real, sistemas enterprise.

Diseño de componentes, patrones de diseño y buenas prácticas de diseño.

Calidad del diseño de productos de software, calidad interna, caracterización de los diseños, métricas y herramientas. Sesiones de revisión de diseño.

PROGRAMA ANALÍTICO

Conceptos, paradigmas y modelos de programación. Evolución histórica de los modelos de programación y el concepto de diseño.

Enfoque lógico, funcional, orientado a objetos y por prototipos. Diseño de software en modelos declarativos e imperativos.

Criterios de buen diseño.

Lenguajes procedurales. Módulos, interfaces y funciones. Cohesión y acoplamiento. Eficiencia de procesamiento como restricción de diseño.

Lenguajes orientados a objetos. Inversión en la cadena de dependencia, criterio de substitución de Liskov, código clausurado ante cambios, segregación de interfaces, simple responsabilidad. Ley de Demeter.

El proceso de diseño. Requerimientos funcionales y no funcionales como conductores y restricciones de diseño.

Entender el problema, patrones de colaboración. Modelo de dominio.

Implementar una solución, patrones de diseño. Modelo de diseño.

Diseño conducido por el dominio. Diseño conducido por las pruebas.

Arquitectura de software, conceptos, vistas y enfoques. Patrones de arquitectura. Sistemas de procesamiento

de información, sistemas de tiempo real, sistemas enterprise.

Diseño de componentes, patrones de diseño y buenas prácticas de diseño.

Calidad del diseño de productos de software, calidad interna, caracterización de los diseños, métricas y herramientas. Sesiones de revisión de diseño.

BIBLIOGRAFÍA

1. Ingeniería de Software, Guillermo Pantaleo, Ludmila Rinaudo, Alfaomega, 2015.
2. Streamlined Object Modelling, Jill Nicola et al., Prentice Hall, 2002.
3. Clean Code: A Handbook of Agile Software Craftsmanship, Robert C. Martin, Prentice Hall, 2008.
4. Software systems architecture : working with stakeholders using viewpoints and perspectives, Nick Rozanski, Eoin Woods, Pearson Education, 2005.
5. Pattern Oriented Software Architecture, vol 1, Michael Stal, Hans Rohnet, Frank Buschmann, Regine Meunier, Peter Sommerlad, Wiley, 1996.
6. Real-Time Design Patterns: Robust Scalable Architecture for Real-Time Systems, Bruce Powel Douglass, Addison Wesley, 2002.
7. Patterns of Enterprise Application Architecture, Martin Fowler, Addison Wesley, 2003.
8. Design Patterns, E. Gamma et al., Addison Wesley, 1995.
9. Object-Oriented Metrics in Practice: Using Software Metrics to Characterize, Evaluate, and Improve the Design of Object-Oriented Systems, Lanza, Marinescu, Ducasse, Springer, 2006.
10. Domain-Driven Design: Tackling Complexity in the Heart of Software, Eric Evans, Pearson Education, 2005.
11. Test Driven Development: By Example, Kent Beck, Pearson Education, 2003.

RÉGIMEN DE CURSADA

Metodología de enseñanza

Clases teóricas y prácticas.

Desarrollo de un trabajo práctico con asistencia de los docentes auxiliares.

Modalidad de Evaluación Parcial

Exámenes Parciales

TP aprobado y publicado

Promoción con nota siete (7) o superior o Coloquio

CALENDARIO DE CLASES

Semana	Temas de teoría	Resolución de problemas	Laboratorio	Otro tipo	Fecha entrega Informe TP	Bibliografía básica
<1> 09/03 al 14/03	Criterios de diseño					
<2> 16/03 al 21/03	Proceso de diseño - Presentación					
<3> 23/03 al 28/03	Proceso de diseño – desarrollo de un ejemplo					
<4> 30/03 al 04/04	Modelo de dominio					
<5> 06/04 al 11/04	Modelo de dominio					
<6> 13/04 al 18/04	Arquitectura de Software					
<7> 20/04 al 25/04	Patrones de Arquitectura					
<8> 27/04 al 02/05	Resolución de ejercitación de proceso de diseño				Presentación parcial del avance de los TP para grupos impares	
<9> 04/05 al 09/05	Patrones creacionales				Presentación parcial del avance de los TP para grupos impares	
<10> 11/05 al 16/05	Patrones Organización del trabajo Patrones Extensión de Servicio					
<11> 18/05 al 23/05	Patrones Variación de Servicio					
<12> 25/05 al 30/05	Programación Funcional					
<13> 01/06 al 06/06	Diseño Funcional					
<14> 08/06 al 13/06	Programación por Prototipos					
<15> 15/06 al 20/06	Diseño por Prototipos				Entrega de los TP de los grupos impares	
<16> 22/06 al 27/06	Métricas de diseño y código				Entrega de los TP de los grupos pares	

CALENDARIO DE EVALUACIONES

Evaluación Parcial

Oportunidad	Semana	Fecha	Hora	Aula
1º	7	20/04	19:00	
2º	11	18/05	19:00	
3º	15	22/06	19:00	
4º	16	29/06	19:00	
Observaciones sobre el Temario de la Evaluación Parcial				
Se evalúa a los alumnos acerca de todos los temas dados sobre un problema en el cual deben reconocer contextos, problemas a resolver y proponer soluciones basados en los patrones de arquitectura y diseño vistos				