



ANÁLISIS Y DISEÑO DE SISTEMAS 2

MARTES 7:10 A 8:50 (T-3 216) – JUEVES 17:20 A 19:00 (T-3 212)

PROGRAMA DEL CURSO

ING. RICARDO MORALES

EMAIL: JRMPRADO@PROTONMAIL.COM

DESCRIPCIÓN GENERAL

¿ACERCA DE QUÉ ES EL CURSO?

¿Has pensado en que existe más de una forma de definir la estructura interna de un sistema de software? En un mundo donde se busca obtener resultados más rápidos, manteniendo la calidad de un producto, los conceptos de patrones de diseño y arquitectura de software son herramientas de gran utilidad para alcanzar esos objetivos

El curso es acerca de patrones aplicables para el diseño de componentes de una aplicación, es decir patrones de diseño. El uso de dichos patrones, de patrones de arquitectura, técnicas para definir una arquitectura que satisfaga los requerimientos funcionales y no funcionales, comprenden el resto del curso. Se aplicarán los elementos anteriores en un proyecto de curso para poder contar con una experiencia real en el uso de los patrones y técnicas presentados.

¿POR QUÉ ES IMPORTANTE EL CURSO?

Dada la presencia del software en casi todos los aspectos de la sociedad, la confiabilidad, disponibilidad, seguridad y calidad en general del mismo es un factor crítico de éxito, tanto para un producto de software, como para la comunidad de usuarios que lo utilice.

Para alcanzar niveles de calidad acorde a las expectativas de los clientes, es necesario aplicar procesos en el diseño del software, de manera que los mismos consideren los diferentes tipos de requerimientos que pueden impactar en la definición de la estructura de un sistema de software.

OBJETIVOS

¿QUÉ DEBO SABER Y PODER HACER AL FINAL DEL CURSO?

Identificará y describirá los conceptos de arquitectura de software para el desarrollo de sistemas informáticos

Identificará y describirá los conceptos de atributos de calidad y tácticas aplicables para alcanzarlos

Evaluará y adaptará el diseño de un sistema de software de acuerdo con los diferentes tipos de requerimientos de un proyecto

Diseñará la arquitectura de software de un proyecto real, con base en diferentes puntos de vista y perspectivas para alcanzar los atributos de calidad de dicho proyecto.

Aplicará técnicas para medición cuantificable del rendimiento y escalabilidad de una aplicación

CONTENIDO

Conceptos generales de arquitectura de software

Elementos de diseño

Patrones de diseño

Principios de diseño

Arquitectura de software

Vistas, puntos de vista y perspectivas

Atributos de calidad

Estilos de arquitectura

Proceso de arquitectura (ADD, Microsoft, ACDM, ADM)

Temas transversales

Descripciones Estructurales (Vistas

Estáticas)

Descripciones de Comportamiento

(Vistas Dinámicas)

Optimización

Uso de perspectivas

Pruebas cuantitativas de rendimiento y escalabilidad

Profiling

METODOLOGÍA

Se impartirá clase magistral 2 veces por semana.

Así mismo se formarán grupos para el desarrollo de un proyecto donde se deben aplicar los conceptos aprendidos en el curso, de forma práctica, dando solución a un problema real, lo cual se complementará con tareas y evaluaciones.

Se desarrollarán investigaciones y exposiciones de conferencias recientes relacionadas con el curso.

EVALUACIÓN

EVALUACIONES PARCIALES

Se realizarán 2 evaluaciones parciales individuales (26 febrero y 2 de abril). Las evaluaciones se realizarán en clase, de forma escrita o electrónica y evaluarán el contenido cubierto hasta la clase anterior a la evaluación.

PROYECTO

Los estudiantes deben formar grupos para el desarrollo de un proyecto de curso que consistirá en la construcción de un sistema de software utilizando las prácticas que están en el contenido del curso.

El proyecto está dividido en 3 fases:

- Fase 1 (11 febrero, 6 puntos), evaluar framework de desarrollo, herramientas de apoyo al proyecto, sistema a desarrollar, evaluar plataformas en la nube y arquitectura a utilizar.
- Fase 2 (3 marzo, 7 puntos), hacer un prototipo de una aplicación utilizando cada uno de los patrones de arquitectura (monolítico, orientado a eventos, orientado a servicios, serverless), incluyendo un análisis de los atributos de calidad asociados a cada patrón

- Fase 3 (26 marzo, 8 puntos), definición de arquitectura, desarrollo, profiling y pruebas de rendimiento.
- Fase 3 (30 abril, 9 puntos), mejora de arquitectura y aplicación de perspectivas.

Para la evaluación de cada fase se definirá una rúbrica que contendrá detalle los aspectos a calificar.

EXPOSICIÓN

Cada grupo debe realizar una exposición de una conferencia de InfoQ, que será asignada al inicio del curso

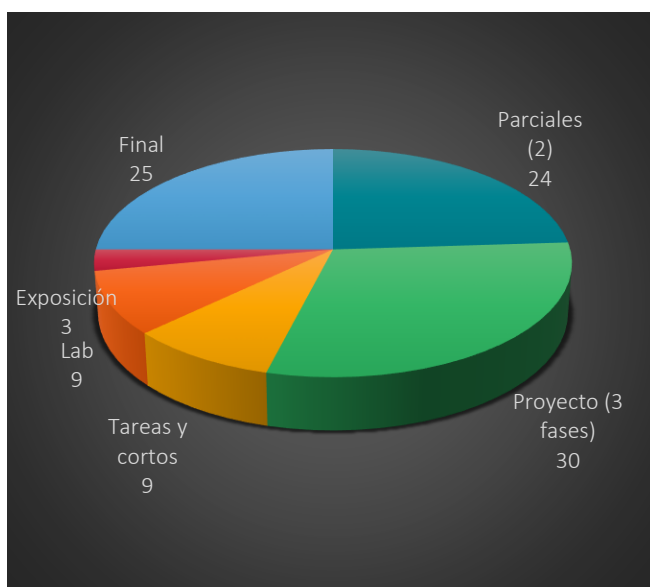
LABORATORIO

El auxiliar del curso llevará a cabo actividades de apoyo al desarrollo de las diferentes fases del proyecto y aplicación de las prácticas del curso.

TAREAS Y CORTOS

En las clases durante el semestre se asignarán lecturas para resumen y discusión, así como exámenes cortos.

DISTRIBUCIÓN DE LA EVALUACIÓN



POLÍTICAS

Las entregas fuera de fecha no son aceptadas.

No es permitido el uso de celulares durante la clase y exámenes parciales.

Debe existir respeto por las opiniones de los demás.

Como estudiantes universitarios, se espera que sepan y entiendan las guías de ética y plagio relacionadas con trabajos de otros autores.

Al final del semestre, no se asignarán trabajos extra para recuperar puntos de zona.

Se debe aprobar (61/100) el laboratorio del curso para tener derecho a examen final.

BIBLIOGRAFÍA

1. Essential software architecture, Ian Gorton
2. Just enough software architecture, George Fairbanks
3. Head first design patterns, Elisabeth Freeman, Eric Freeman, Bert Bates, Kathy Sierra
4. Software systems architecture, working with stakeholders using viewpoints and perspectives, Nick Rozanski, Eoin Woods
5. Software performance and scalability: a quantitative approach, Henry H. Liu

CALENDARIO

FECHA	EN CLASE	TAREA / ENTREGAS
Martes Ene 21	Bienvenida Revisión programa del curso Conceptos generales de arquitectura de software Definición Importancia de arquitectura de software Ejercicio requerimientos diferentes sistemas	Lectura capítulos 1 y 2, bibliografía 2 Ejercicio requerimientos diferentes sistemas
Jueves Ene 23	Patrones de diseño Conceptos generales Patrón strategy Patrón observer	Entrega ejercicio requerimientos diferentes sistemas Lectura artículo Five Things Every Developer Should Know about Software Architecture, https://www.infoq.com/articles/architecture-five-things
Martes Ene 28	Patrón factory	
Jueves Ene 30	Principios de diseño Single Responsibility Principle (SRP) Open-Closed Principle (OCP) Liskov Substitution Principle (LSP)	
Martes Feb 04	Interface Segregation Principle (ISP) Dependency Inversión Principle (DIP) Inversión del Control (IoC)	
Jueves Feb 06	Vistas, puntos de vista y perspectivas Vista Punto de vista Perspectiva	
Martes Feb 11	Atributos de calidad	Entrega fase 1 proyecto
Jueves Feb 13	Exposición fase 1 proyecto	
Martes Feb 18	Estilos de arquitectura Escenarios Estilos de arquitectura	Exposición infoq, grupo 1

Jueves Feb 20	Estilos de arquitectura	Exposición infoq, grupo 2
Martes Feb 25	Primer examen parcial	
Jueves Feb 27	Proceso de arquitectura Definir contexto y alcance inicial Involucrar stakeholders Capturar primera versión de temas de interés Definir la arquitectura	Exposición infoq, grupo 3
Martes Mar 03	Catálogo de puntos de vista Punto de vista de contexto Punto de vista funcional Punto de vista de información	Exposición infoq, grupo 4 Entrega fase 2 proyecto
Jueves Mar 05	Presentación y calificación fase 2	
Martes Mar 10	Punto de vista de concurrencia Punto de vista de desarrollo	Exposición infoq, grupo 5
Jueves Mar 12	Punto de vista de deployment Punto de vista operacional	Exposición infoq, grupo 6
Martes Mar 17	Catálogo de perspectivas Perspectiva de seguridad Perspectiva de rendimiento y escalabilidad	Exposición infoq, grupo 7
Jueves Mar 19	Perspectiva de rendimiento y escalabilidad	
Martes Mar 24	Perspectiva de disponibilidad y resistencia	Exposición infoq, grupo 8
Jueves Mar 26	Entrega fase 3 proyecto	Entrega fase 3 proyecto
Martes Abr 14	Pruebas cuantitativas de rendimiento y escalabilidad Introducción Stack de software	

Jueves Abr 16	Probando rendimiento y escalabilidad de software	
Martes Abr 21	Segundo examen parcial	
Jueves Abr 23	Probando rendimiento y escalabilidad de software	
Martes Abr 28	Profiling	
Jueves Abr 30	Teoría de colas	Entrega fase 4 proyecto
Martes May 05	Calificación fase 4 proyecto	
Jueves May 07	Calificación fase 4 proyecto	