



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: RMORALES88@LYCOS.COM

DESCRIPCIÓN GENERAL

¿ACERCA DE QUÉ ES EL CURSO?

¿Has pensado en que existe mas de una forma de definir la estructura interna de un sistema de software? En un mundo donde se busca obtener resultados mas rápido, 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.

CONTENIDO

Conceptos generales de arquitectura de software

Patrones de diseño

Principios de diseño

Estilos de arquitectura

Vistas, puntos de vista y perspectivas

Proceso de arquitectura

Catálogo de puntos de vista

Catálogo de perspectivas

METODOLOGÍA

Se impartirá clase magistral 2 veces por semana.

Así mismo se formarán grupos para el desarrollo de un proyecto grupal 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 patrones de diseño y conferencias recientes relacionadas con el curso.

EVALUACIÓN

EVALUACIONES PARCIALES

Se realizarán 2 evaluaciones parciales individuales (27 febrero y 17 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 (15 febrero), evaluar framework de desarrollo, herramientas de apoyo al proyecto, sistema a desarrollar y arquitectura a utilizar.
- Fase 2 (13 marzo), definición de arquitectura, desarrollo, profiling y pruebas de rendimiento.
- Fase 3 (01 mayo), 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 TEMAS

Cada grupo debe realizar 2 exposiciones en el curso:

- Un patrón de diseño
- Una conferencia de InfoQ

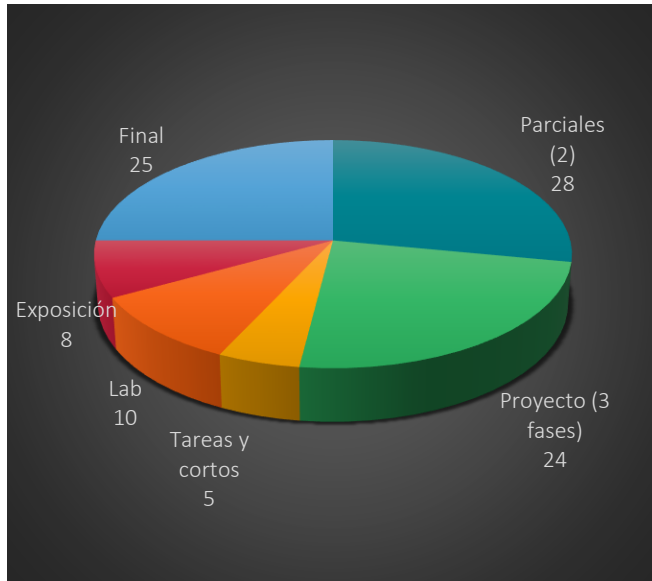
LABORATORIO

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.

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
6. The Docker Book: Containerization is the new virtualization, James Turnbull

CALENDARIO

FECHA	EN CLASE	TAREA / ENTREGAS
Martes Ene 23	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 25	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 29	Principios de diseño Single Responsibility Principle (SRP) Open-Closed Principle (OCP)	Exposición patrón decorator Exposición patrón Factory
Jueves Feb 01	Liskov Substitution Principle (LSP) Interface Segregation Principle (ISP)	Exposición patrón singleton Exposición patrón command
Martes Feb 06	Dependency Inversión Principle (DIP) Inversión del Control (IoC)	Exposición patrón adapter y facade Exposición patrón template method
Jueves Feb 08	Vistas, puntos de vista y perspectivas Vista Punto de vista	Exposición patrón iterator y composite Exposición patrón state
Martes Feb 13	Perspectiva	Exposición patrón proxy Exposición patrones compuestos
Jueves Feb 15	Proceso de arquitectura Definir contexto y alcance inicial Involucrar stakeholders Capturar primera versión de temas de interés Definir la arquitectura	Entrega Fase 1 proyecto Exposición infoq, grupo 1
Martes Feb 20	Calificación fase 1 proyecto	

Jueves Feb 22	Estilos de arquitectura Escenarios Estilos de arquitectura	Exposición infoq, grupo 2
Martes Feb 27	Primer examen parcial	
Jueves Mar 01	Estilos de arquitectura	Exposición infoq, grupo 3
Martes Mar 06	Catálogo de puntos de vista Punto de vista de contexto Punto de vista funcional Punto de vista de información	Exposición infoq, grupo 4
Jueves Mar 08	Punto de vista de concurrencia punto de vista de desarrollo	Exposición infoq, grupo 5
Martes Mar 13	Prácticas de desarrollo	Entrega Fase 2 proyecto
Jueves Mar 15	Calificación fase 2 proyecto	
Martes Abr 03	Punto de vista de deployment Punto de vista operacional	Exposición infoq, grupo 6
Jueves Abr 05	Catálogo de perspectivas Perspectiva de seguridad Perspectiva de rendimiento y escalabilidad	Exposición infoq, grupo 7
Martes Abr 10	Perspectiva de rendimiento y escalabilidad Perspectiva de disponibilidad y resistencia	Exposición infoq, grupo 8
Jueves Abr 12	Escalabilidad y rendimiento, un enfoque cuantitativo Introducción Stack de software	
Martes Abr 17	Segundo examen parcial	
Jueves Abr 19	Probando rendimiento y escalabilidad de software	

Martes Abr 24	Probando rendimiento y escalabilidad de software Teoría de colas
Jueves Abr 26	
Martes May 01	Feriado Entrega fase 3 proyecto
Jueves May 03	Calificación fase 3 proyecto

LISTADO CONFERENCIAS INFOQ

1. The Evolution of Reddit.com's Architecture, <https://www.infoq.com/presentations/reddit-architecture-evolution>
2. Architecting a Modern Financial Institution, <https://www.infoq.com/presentations/nubank-architecture>
3. Introduction to Blockchain Technology and Ethereum, <https://www.infoq.com/presentations/blockchain-ethereum>
4. Scaling Stack Overflow: Keeping it Vertical by Obsessing Over Performance, <https://www.infoq.com/presentations/stack-exchange>
5. Coding for High Frequency Trading, <https://www.infoq.com/presentations/financial-services-languages-environments>
6. Serverless Architectures, <https://www.infoq.com/presentations/serverless-introduction>
7. Scaling Event Sourcing for Netflix Downloads, <https://www.infoq.com/presentations/netflix-scale-event-sourcing>