

Programa del curso

2019-20

Nombre:	Arquitectura y diseño de software
Créditos	3
Zona	Curso básico profesional
Tipo de proyecto	Construcción de soluciones de TI
Pre-requisitos	ISIS 2603
Profesora	Kelly Garcés kj.garces971@uniandes.edu.co
Versión PDF	Click Aquí

Propósito

El propósito de este curso es desarrollar habilidades en las áreas de diseño y arquitectura de software, apoyándose en herramientas metodológicas como los estilos de arquitectura, los patrones (e.g. diseño, arquitectura). Igualmente, se pretende desarrollar la capacidad de usar y entender el impacto de la tecnología en la arquitectura del software.

Objetivos generales

El curso pretende desarrollar las competencias de definir, justificar, implementar y evaluar una arquitectura de software para un problema en el mundo empresarial. Al finalizar el curso el estudiante debe estar en capacidad de:

- Identificar los principales atributos de calidad del software (disponibilidad, desempeño, modificabilidad, seguridad) y su impacto en el diseño y desarrollo de software.
- Utilizar los requerimientos funcionales y requerimientos de calidad para seleccionar los estilos arquitecturales y patrones de diseño más apropiados para el diseño de una solución.
- Utilizar los conceptos básicos de arquitecturas de software para definir una estrategia de desarrollo. Esto significa, identificar los elementos del mundo de la solución □ los componentes de software, sus propiedades visibles externamente y las relaciones entre ellos □ para estructurar y organizar el proceso de desarrollo de software.
- Utilizar adecuadamente los patrones (e.g. diseño, arquitectura) y los estilos de arquitectura para explicar el diseño de un sistema. Explicar y proponer una

solución a los problemas de interacción entre los componentes. Justificar cómo una solución responde a los requerimientos y restricciones de un negocio.

- Comprender y desarrollar habilidades en el uso de tecnologías para el apoyo de arquitecturas de software.
- Realizar una aplicación mediana usando el modelo empresarial por componentes JEE.
- Diseñar e implementar experimentos para validar propiedades de calidad en el diseño de una solución.

Metodología

El curso gira alrededor de un proyecto de tamaño mediano y sobre el cual se introducen los elementos teóricos y metodológicos. El objetivo principal es diseñar la solución y experimentar si la aproximación seleccionada cumple o no con los objetivos propuestos a nivel de atributos de calidad. Las clases se centran en el análisis y la propuesta de diseño del proyecto. A lo largo del semestre, se desarrollarán tres experimentos. El objetivo de los experimentos es comprobar si las decisiones de diseño son apropiadas a la luz de los atributos de calidad del software. El estudiante debe asistir y participar de manera activa en las clases teóricas y desarrollar los laboratorios relacionados con el tema.

Categorías de habilidades y objetivos pedagógicos

Objetivo Pedagógico	Metas específicas	%
O4	Identificar y documentar escenarios de calidad	12%
O6	Identificar y documentar motivadores de negocio	3%
O8	Diseñar y documentar una arquitectura de software: Aplicar tácticas, estrategias y estilos	23%
O9	Aplicar patrones de diseño detallado a la solución	15%
O10	Diseñar e implementar experimentos para validar propiedades de calidad en el diseño de una solución.	6%
O11	Integrar la arquitectura propuesta en un proceso de desarrollo de software	3%
O13	Construcción de prototipos para experimentación	15%
O14	Documentar una arquitectura de solución	15%
O17	Presentar y negociar una arquitectura de software con los stakeholders (involucrados)	6%

Categorías de habilidades y objetivos pedagógicos

Objetivo Pedagógico	Metas específicas
OT1	Utilizar herramientas empresariales para la implementación de los prototipos
OT2	Experimentar propuestas de arquitectura
OT3	Entender los resultados de los experimentos
OT4	Diseñar e implementar una arquitectura en equipos
OT9	Aplicar conceptos a situaciones y dominios heterogéneos
OT12	Respetar los resultados de los experimentos

Manejo de los Aspectos Transversales

Aspecto	Descripción
Gerencia	El diseño de alto nivel implica generar habilidades de negociación y manejo de recursos tanto con los stakeholders como con el equipo de desarrollo
Calidad	<ul style="list-style-type: none"> Las fases de validación de la arquitectura permite mejorar la calidad del producto Entender las ventajas que ofrece una buena documentación arquitectural durante la construcción de una solución Sector externo Presentación de casos de estudio tomados de situaciones reales a nivel nacional e internacional para ser estudiados y analizados como parte del proyecto de curso
Innovación	Utilización de tecnologías de punta y proyectos con requerimientos no tradicionales
Seguridad	Se considera la seguridad como un atributo de calidad principal que guía y determina el resultado tanto del diseño arquitectural como del diseño detallado

Reglas de convivencia

La entrada a clase debe ser puntual, máximo hasta 10 minutos de iniciada esta. Posterior a este plazo no se permitirá el ingreso a clase. Todo trabajo debe ser entregado en las fechas estipuladas. Después de la fecha designada no se aceptan trabajos sin excusa justificada. Se restringe el uso de celulares, portátiles, reproductores de audio, y el consumo de alimentos en clase.

Evaluación

La evaluación está compuesta de la siguiente manera, hay actividades individuales y

grupales:

Evaluación Individual

- Parcial 1 (15%)
- Parcial 2 (15%)
- Laboratorios individuales (15%)
- Tareas individuales (5%)

Evaluación Grupal

- Experimentos (30%)
- Refinamiento de requerimientos del proyecto (diseños que se producen a lo largo del semestre) (5%)
- SAD 1 (5%)
- Sustentación Final SAD (10%)

Cada entrega grupal tendrá una nota global, pero cada miembro tendrá una nota individual de acuerdo al esfuerzo invertido en la entrega. El grupo determina el esfuerzo porcentual en una reunión recordando las tareas que cada individuo hizo y plasman el resultado en un formato que debe ser enviado junto con la entrega. El formato es definido por la profesora. Por ejemplo, un grupo de 4 miembros saca 4 en la nota global, la siguiente tabla muestra el porcentaje de esfuerzo de cada miembro y la nota individual asignada:

Integrante	Esfuerzo (%)	Nota individual
1	30	4.8
2	20	3.2
3	25	4
4	25	4

La nota individual no excede 5. Para aprobar el curso la nota calculada debe ser al menos de 3.00. Las definitivas se darán en dos decimales. Las notas finales tendrán dos decimales. Las reglas del redondeo que se aplican al segundo decimal son: Si el tercer dígito es menor que 5, el anterior no se modifica. Si es mayor o igual que 5, el anterior se incrementa en una unidad. La asistencia a las clases teóricas y laboratorios es obligatoria. El estudiante que no asista a por lo menos el 80% de las sesiones no podrá aprobar el curso. Para cada laboratorio, hay entrega parcial (al final de la sesión de laboratorio) y una definitiva (a definir por sección). La entrega definitiva se acepta si se hizo una entrega parcial. La entrega de los experimentos es los domingos a medianoche. Para los laboratorios se dispone de máquinas virtuales que poseen software básico (e.g., Netbeans, Eclipse, Glassfish). Para algunos laboratorios es responsabilidad del estudiante instalar el software que falte antes de cada sesión de laboratorio. Para esto cada laboratorio indica una lista de requerimientos. Para algunos entregables se propodrán bonos, un grupo obtiene bono si satisface requerimientos no funcionales adicionales a los básicos. En todo caso, la nota del entregable más la nota del bono no excede 5.0.

Bibliografía

- [1] Rozanski, N., Woods, E., [Software Systems Architecture], Addison Wesley. 2005
- [2] Bass, L. Clements, P., Kazman, R., [Software Architecture in Practice], Addison-Wesley, Second Edition, 2006.
- [3] Paul Clements et al, [Documenting Software Architectures: Views and Beyond], Addison Wesley, 2002.
- [4] Paul Clements et al, [Evaluating Software Architectures], Addison Wesley, 2002.
- [5] Erl, T., [SOA Principles of Service Design], Prentice Hall, 2008
- [6] Geary, D., Horstmann, C., [Core JavaServer Faces] Second Edition, Prentice Hall, 2007
- [7] Richard Taylor, Nenad Medvidovic, Eric Dashofy. [Software Architecture Foundations, Theory and Practice, 2009.
- [8] Frank Buschmann, Kevin Henney, Douglas Schmidt. Pattern-Oriented Software Architecture. Volume 4.
- [9] John Reekie, Rohan McAdam. A Software Architecture Primer. Angophora Press. 2006
- [10] Jan Bosh. Design & Use of Software Architecture. Addison-Wesley. 2000
- [11] Kai Qian et Al. Software Architecture and Design Illuminated. 2010.
- [12] Paul Clements, et al. Documenting Software Architectures: Views and Beyond (2nd Edition), Addison-Wesley Professional; 2 edition (October 15, 2010)
- [13] Rozanski Nick, Eoin woods. Software Systems Architecture: Working With Stakeholders Using Viewpoints and Perspectives (2nd Edition). Addison-Wesley Professional; 2 edition (November 4, 2011).
- [14] Peter Eeles, Peter Crips. The Process of Software Architecting. Addison-Wesley Professional; 1 edition (July 24, 2009).
- [15] Dave Hendricksen, 12 Essential Skills for Software Architects. Addison-Wesley Professional; 1 edition (October 5, 2011)
- [16] Anthony Lattanze. Architecting Software Intensive Systems: A Practitioners Guide. Auerbach Publications (November 18, 2008)
- [17] Michael Sikora. EJB 3 Developer Guide : A Practical Guide for Developers and Architects to the Enterprise Java Beans Standard. Ebook.