

ESCUELA COLOMBIANA DE INGENIERÍA

ARQUITECTURAS DE SOFTWARE (ARSW)

PROGRAMA DEL CURSO 2020-2

Objetivos

- Al finalizar el curso el estudiante estará en capacidad de proponer, construir y evaluar arquitecturas y diseños de software para sistemas de información empresariales. Específicamente adquirirá conocimientos y desarrollará habilidades para:
- Conocer los estilos arquitectónicos y patrones de diseño útiles en la profesión.
- Definir y justificar una arquitectura para un problema específico considerando los estilos arquitectónicos existentes, y sus tecnologías relacionadas.
- Tomar decisiones de carácter arquitectónico en un proyecto de software, considerando el impacto que tienen dichas decisiones sobre los diferentes atributos de calidad.
- Implementar adecuadamente los estilos arquitectónicos planteados en el curso.
- Documentar adecuadamente la arquitectura de un software.

Contenido del curso

- Para cada uno de los estilos acá indicados, durante el desarrollo del curso tendrán asociado un conjunto de 'vistas', atributos de calidad y tecnologías concretas relacionadas.
- Concurrencia y paralelismo
- Ley de Moore vs Ley de Amdahl.
- Hilos vs Procesos, CPU vs GPU: Problemas embarazosamente paralelos.
- Concurrencia: interleaving, condiciones de carrera, regiones críticas.
- Mecanismos de sincronización, deadlocks.
- Estilos arquitectónicos modulares
- Estilo multicapa.
- Principio de inversión de dependencias.
- Contenedores livianos e inyección de dependencias.
- Estilos arquitectónicos tipo componente-conector
- Estilo cliente/servidor.
- Estilo punto a punto
- Arquitecturas orientadas a servicios (SOA): Conceptos básicos, principios, protocolos y estilos relacionados (SOAP/REST).
- Clientes Web
- Modelo SPA (manipulación del DOM), programación asíncrona, callbacks (callback hell), promesas, async/await.
- Server-side PUSH, brokers y protocolos de mensajería (STOMP sobre WebSockets).
- Atributos de calidad: concepto, ejemplos, y el problema de la compensación (tradeoff). Escenarios de atributos de calidad.
- Estrategias de escalabilidad.
- Bases de datos NoSQL: documentales, llave valor, de grafos, etc.
- Estrategias de Caché.
- Balanceo de carga, reverse proxies
- Infraestructura como Servicio (IAAS).
- Documentación de arquitecturas de software.

Metodología

Sesiones teóricas y prácticas

Semanalmente el curso tendrá una sesión teórica y otra práctica. En la sesión teórica se realizarán exposiciones de cada uno de los temas planteados en el temario, mientras que en la sesión práctica se buscará que los estudiantes apropien más los temas junto con algunas herramientas que les serán de utilidad tanto para el desarrollo del proyecto del curso como para el ejercicio profesional, a través de ejercicios puntuales. En el primer tercio del semestre los ejercicios tendrán dos secciones: una para realizarla presencialmente (y entregar al final de la clase), y otra para finalizarla como tarea. Estos ejercicios se evaluarán como si se tratara de un proyecto, y será requisito haber entregado lo indicado en la sesión presencial para evaluar la totalidad del ejercicio. Diez minutos después de iniciada la clase (teórica o práctica) eventualmente se realizará un control de lectura sobre las lecturas definidas previamente para cada semana. Dichas lecturas se irán agregando en el enlace "Contenido semanal + Lecturas a preparar" disponible en la página del curso.

Proyecto de curso

En el primer corte del curso se deben conformar grupos de tres personas. Cada uno de estos deberá plantear un proyecto, el cual se irá afinando con la ayuda del profesor. Las condiciones y restricciones para dicha propuesta se darán en las primeras semanas del curso. Se espera que como propuesta de proyecto se entregue un documento bien redactado, bien presentado, y suficientemente preciso. En caso de que al final del tercio no se llegase a una propuesta aceptable, se tendrá una evaluación deficiente en este ítem, y el proyecto será asignado por el profesor. Para el segundo tercio los grupos iniciarán con la implementación del proyecto usando como marco de gestión SCRUM, una metodología orientada por pruebas, y usando como medio de administración de la configuración un repositorio GIT. Sin embargo, es posible que se exija la separación de un grupo si, al haber trabajado conjuntamente en los laboratorios, se encuentran diferencias marcadas en la evaluación del primer tercio. El requisito para iniciar el proyecto en el tercer tercio será que lo entregado en el segundo funcione correctamente, así que, en caso de tener un resultado deficiente en el segundo tercio, será responsabilidad del estudiante hacer los ajustes que requiera su proyecto antes de continuar con el trabajo. En estos dos tercios (segundo y tercero) se continuará con el trabajo en las sesiones prácticas, pero ahora teniendo como propósito brindar las herramientas necesarias para culminar el proyecto. El resultado en las sesiones prácticas también tendrá un porcentaje (menor) en las notas definitivas, y para motivar la entrega total de dichos ejercicios, pueda que se evalúen sólo algunos al azar.

Reglas de juego

Con la metodología del curso se busca que los estudiantes adquieran habilidades y conocimientos en temas de arquitecturas y tecnologías de construcción de software. Sin embargo, esto tiene como requisito compromiso, seriedad y honestidad de su parte. Por esta razón, se debe tener presente que:

- Si durante la revisión de las entregas preliminares de los proyectos se identifica, así sean pequeños segmentos de código duplicado, esto se considerará plagio, y tendrá como efecto el reprobar la evaluación, tanto para el que proveyó su proyecto como para el que lo copió. Esto también aplica para los ejercicios de laboratorio.
- La nota obtenida en el proyecto podría ser diferente a la evaluación del mismo. Es decir, si uno de los integrantes muestra desconocimiento del proyecto, o hace evidente que no trabajó en el mismo, su nota será una fracción de la evaluación del proyecto. Para las sustentaciones, eventualmente se delegará a solo uno de los integrantes.

Evaluación

Primer tercio

- Parcial 50%
- Laboratorios 30%
- Propuesta de proyecto 10%
- Quices y talleres 10%

Segundo y tercer tercio

- Parcial 40%
- Proyecto 40% (50% nota grupo, 50% nota individual)
- Laboratorios 15%
- Quices 5%

Bibliografía

Aunque semanalmente se darán referencias a material de lectura disponible en línea, las siguientes son referencias básicas a ser usadas a lo largo de todo el semestre:

- [1]. Clements, P., Garlan, D., Bass, L., Stafford, J., Nord, R., Ivers, J., & Little, R. (2002). Documenting software architectures: views and beyond. Pearson Education. (Disponible en la biblioteca).
- [2]. Brian Goetz, Joshua Bloch, Joseph Bowbeer, Doug Lea, David Holmes, Tim Peierls. (2006), Java Concurrency in Practice. (Disponible en la biblioteca).
- [3]. Barbacci, M., Klein, M. H., Longstaff, T. A., & Weinstock, C. B. (1995). Quality Attributes (No. CMU/SEI-95-TR-021). CARNEGIE-MELLON UNIV PITTSBURGH PA SOFTWARE ENGINEERING INST. <http://www.sei.cmu.edu/reports/95tr021.pdf>