



PROGRAMA DE ASIGNATURA¹

NOMBRE ASIGNATURA: **Arquitectura de Software**

Código: **INFO229**

Identificación general

Identificación general			
Docente responsable	Matthieu VERNIER mvernier@inf.uach.cl	Docentes colaboradores	
Correo electrónico		Correo electrónico	
Horario y sala de clases	Viernes 11.30 hrs – 13.00 hrs Viernes 15.50 hrs – 17.20 hrs Edificio 10.000 – Sala Cuiñanco		
Año y semestre	2019 – Segundo semestre		

Antecedentes de la asignatura, según proyecto curricular de la carrera									
Unidad Académica	Facultad de Ciencias de la Ingeniería		Carrera	Ingeniería Civil en Informática			Semestre en plan de estudios		Semestre VI
Asignaturas- requisito (con código)	INFO133 - Base de Datos						Créditos SCT-Chile		5
Horas cronológicas semestre	Teóricas presenciales	25,5	Prácticas presenciales	25,5	Trabajo Autónomo	76,5	Total		127,5
Ciclo formativo	Bachillerato		Licenciatura	X	Profesional				
Área de formación	Especialidad	X	General		Vinculante-profesional		Optativa		
Descripción de la asignatura	La asignatura Arquitectura de software tiene como principal propósito que los estudiantes utilicen procesos de construcción de software para la integración y el despliegue continuo, el monitoreo y pruebas de software, y el diseño con patrones arquitectónicos, describiendo cómo estos procesos permiten facilitar el ciclo de desarrollo de software en términos de tiempo de desarrollo, posibilidad de mantención y evolución del software.								

Aporte de la asignatura al Perfil de Egreso, según proyecto curricular de la carrera						
Competencias	Nivel de dominio que alcanza la competencia en la asignatura					
-Específicas:	Básico	Medio		Superior	Avanzado	
C8 - Concebir sistemas de software que aporten valor a la organización, con una visión sistémica e innovadora determinando su factibilidad y planificando su desarrollo, en el marco de un modelo de proceso de software apropiado. (*)			X			
C9 - Gestionar el proceso de software eligiendo para cada una de las etapas los estándares, técnicas, metodologías y formalismo que permitan la creación o mantención de un producto de software de calidad que responda a la especificación de requisitos y necesidades de la organización, desde un enfoque multidisciplinario, en un contexto globalizado y cambiante. (**)			X			
-Genéricas:	Básico	Medio		Superior	Avanzado	
C3 - Manifestar una actitud innovadora, emprendedora y de adaptación al cambio en contextos globales y locales del ejercicio de la carrera (***)			X			
-Sello:	Básico	Medio		Superior	Avanzado	
C2 - Demostrar compromiso con la calidad y la excelencia, en el contexto formativo del desarrollo personal y profesional del estudiante con sello UACH (****)			X			

(*) d2 - Propone una arquitectura de software de acuerdo a las necesidades detectadas, considerando componentes de hardware y software, y sus conexiones

(**) d2 - Diseñar producto de software que responde a la especificación de requisitos y necesidades de la organización. d3 - Implementa el producto de software, de acuerdo a los artefactos de diseño, verificando, validando y documentando este proceso

(***) d2 - Es capaz de formular propuestas de cambios, mejoras o ajustes, tanto en áreas generales como de especialidad, a partir del análisis de "situaciones problemáticas o casos" resueltos, los cuales posibilitan el empleo de estrategias de innovación y emprendimiento para lograr una optimización de la solución.

(****) d1 - Es capaz de analizar "casos" de situaciones problemáticas que afectan el compromiso con la calidad y la excelencia, considerando sus principales causas y efectos, tanto en contextos rutinarios como en contextos profesionales.

Programación por Unidades de Aprendizaje

Unidades de Aprendizaje	Resultados de aprendizaje Es capaz de...	Estrategias de enseñanza y aprendizaje	Estrategias de evaluación de los aprendizajes y ponderación	Horas presenciales	Horas de trabajo autónomo
Unidad 1: Procesos de diseño con patrones arquitectónicos <u>¿Cómo facilitar las posibilidades de modificaciones y evolución del software?</u>	Utilizar patrones arquitectónicos para el diseño de software Describir cómo patrones arquitectónicos faciliten las posibilidades de evolución del software	Tutoriales técnicos guiados Clases expositivas-activas: Lectura crítica de artículos, libros y videos	Prueba formativa (20%): participación activa en clase Prueba sumativa (80%): test individual Nota Final: U1	17 (5 semanas)	25,5
Unidad 2: Procesos de integración continua, pruebas y monitoreo de una arquitectura de software <u>¿Cómo acelerar el ciclo de desarrollo de arquitectura de software?</u>	Utilizar procesos y tecnologías de integración continua, pruebas y monitoreo Describir cómo la integración continua, las pruebas y el monitoreo permiten acelerar el ciclo de desarrollo	Tutoriales técnicos guiados Clases expositivas-activas: Lectura crítica de artículos, libros y videos	Prueba formativa (20%): participación activa en clase Prueba sumativa (80%): test individual Nota Final: U2	17 (5 semanas)	25,5
Unidad 3: Procesos de despliegue continuo de una arquitectura de software <u>¿Cómo asegurar que los usuarios tengan una disponibilidad rápida de las últimas modificaciones?</u>	Utilizar procesos y tecnologías de despliegue continuo Describir cómo el despliegue continuo permite que los usuarios tengan siempre acceso a las últimas modificaciones del software	Tutoriales técnicos guiados Clases expositivas-activas: Lectura crítica de artículos, libros y videos	Prueba formativa (20%): participación activa en clase Prueba sumativa (80%): test individual Nota Final: U3	17 (5 semanas)	25,5

Requisitos de aprobación

Es condición de aprobación de la asignatura asistir a al menos 70% de las clases. Los atrasos de más de 5 minutos se consideran como ausencia.

La nota mínima de aprobación es 4,0.

La nota final de la asignatura está constituida por el promedio de las 3 unidades:

NF: $(U1 + U2 + U3)/3$

Existirá 1 evaluación adicional de carácter recuperativo o sustitutivo que reemplazará una de las notas U1, U2 o U3. Esta evaluación se realizará al final del semestre.

Recursos de aprendizaje

Bibliografía obligatoria:

Software Engineering, 10th Edition. Ian Sommerville (2016)

Bibliografía recomendada:

Software architecture for developers. Simon Brown (2015)

Software architecture patterns. Mark Richards (2015)

Software architecture with Python. Anand Balachandran Pillai (2017)

Python microservices development. Tarek Ziade (2017)

Otros recursos

Manual DevOps INF-UACH: <http://devops.inf.uach.cl/>

Slack: canal #info229