

ACTIVIDAD CURRICULAR DE FORMACIÓN

Facultad o Instituto	:	Ciencias de la Ingeniería
Carrera	:	Ingeniería Civil Informática

I. IDENTIFICACIÓN DE LA ACTIVIDAD CURRICULAR

Nombre	:	Taller de Desarrollo de Software					
Código	:	INF-524					
Semestre lectivo	:	X Semestre					
Horas	:	Presencial:	54	Autónomas:	96	TOTAL:	150
Créditos SCT	:	5					
Duración	:	Trimestral		Semestral:	x	Anual:	
Modalidad	:	Presencial:	x	Semi-presencial:		A Distancia:	
Área de Formación	:	Disciplinar:		General:		Profesional:	x
						Práctica:	
Pre-requisito (Si los hubiese)	:	Ingeniería de Software II					

II. DESCRIPCIÓN Y CARACTERIZACIÓN DE LA ACTIVIDAD CURRICULAR

La actividad curricular de Taller de desarrollo de software se desarrolla en el décimo semestre del Plan de estudios de la Carrera de Ingeniería Civil Informática. Tal actividad curricular, localizada en el ciclo avanzado de formación, tiene un carácter práctico-teórico- y pertenece al área curricular de Formación Profesional.

El énfasis de esta actividad curricular se centra en aplicar paradigmas de desarrollo de software en situación de desempeño real donde el conjunto de estudiantes debe investigar, organizar, estimar costos, seleccionar herramientas y personas, definir y ejecutar planes relacionados con todas las fases del paradigma seleccionado.

Se espera que el estudiante juzgue, seleccione y aplique los procesos de diseño de software en la construcción software que satisface los requisitos de los clientes.

La metodología de enseñanza y aprendizaje empleada será con aprendizaje colaborativo, aprendizaje orientado a proyectos.

La evaluación será por medio de productos, tales como artefactos de proyectos y presentaciones orales.

III. COMPETENCIAS DEL PERFIL DE EGRESO ASOCIADAS A LA ACTIVIDAD CURRICULAR.

III.1 COMPETENCIAS PROFESIONALES.

COMPETENCIA	SUBCOMPETENCIA
Gestionar proyectos de software, aplicando estándares y metodologías en el desarrollo, operación y mantenimiento de software de calidad que satisfacen necesidades organizacionales	Gestionar proyectos de software de calidad, que satisfacen necesidades organizacionales reales.
Diseñar soluciones tecnológicas que involucren la integración de software y hardware para la interconectividad entre sistemas informáticos.	Integrar sistemas complejos en arquitecturas procesamiento remoto.

III.2 COMPETENCIAS GENÉRICAS.

COMPETENCIA	SUBCOMPETENCIA
Demostrar coherencia ética entre sus postulados valóricos y sus acciones, respetando los derechos humanos y participando activamente en las organizaciones comunitarias, haciendo primar la responsabilidad social desde una perspectiva cristiana.	Juzgar sus actuaciones basándose en fuentes primarias del cristianismo y referentes espirituales.
Desarrollar la identidad regional, generando instancias de integración recíproca entre la Universidad y la comunidad.	Contribuir al desarrollo de programas en la comunidad regional y local.
Comunicar ideas, tanto en la lengua materna como en el idioma inglés, haciendo uso de las tecnologías de la información para desenvolverse en diversos escenarios, dando soluciones a diversas problemáticas de la especialidad.	Comunicarse de forma oral en inglés de acuerdo a lenguaje científico haciendo uso de las tecnologías de la información en contextos propios de su profesión.

IV. RESULTADOS DE APRENDIZAJE - APRENDIZAJE ESPERADO.

RESULTADOS DE APRENDIZAJES
1.- Formular el plan de proyecto a partir de la identificación de un problema de negocio a solucionar de un contexto local o regional de complejidad media, organizado de acuerdo a la selección y justificación de la metodología de desarrollo a considerar.
2.- Formalizar los requisitos, modelos de análisis y diseño arquitectónico de la solución de acuerdo los artefactos que define el proceso metodológico seleccionado considerando criterios de éxito del proyecto y sus restricciones.
3.- Utilizar diferentes framework y case/entorno en la construcción de software a partir de las especificaciones documentadas entregadas de la solución de acuerdo a estándares establecidos.
4.- Aplicar plan de pruebas al producto software desarrollado, generando la documentación asociada y estableciendo consideraciones para la implantación y mantenimiento futuro.

V. UNIDADES DE APRENDIZAJE Y EJES TEMÁTICOS

R. AP.	UNIDAD	EJE(S) TEMÁTICO(S)
1	Gestión de proyectos de Software	Plan de proyecto. Gestión de recursos, riesgos. Procesos de desarrollo de software basado en paradigmas (metodologías de desarrollo, métodos, herramientas, técnicas). Criterios de elección.
2	Proceso de desarrollo de software: Requisitos, análisis y diseño. Técnicas de documentación	Requisitos de software: elicitación, técnicas, documentación, trazabilidad, verificación, validación. Análisis y Diseño: buenas prácticas, modelos y artefactos, prototipaje. Técnicas de documentación: formalización y herramientas case de apoyo.
3	Proceso de desarrollo de software: Construcción	Buenas prácticas de construcción, gestión de la configuración, uso de framework y herramientas case y entornos.
4	Proceso de desarrollo de software: Pruebas, implantación, mantenimiento	Plan de pruebas, estrategias, técnicas y tipos de pruebas. Herramientas de automatización de pruebas. Implantación: estrategias de puesta en marcha, actividades previas a la puesta en marcha (capacitación usuaria, manuales de usuarios, configuración de equipamiento, instalación de configuraciones y software, etc). Mantenimiento: proceso de mantenimiento en régimen de operación de sistemas.

VI. ESTRATEGIAS DE ENSEÑANZA-APRENDIZAJE

De acuerdo al Modelo Educativo de la Universidad Católica del Maule, la metodología de trabajo propuesta para el desarrollo de la actividad curricular se basa en un enfoque activo-participativo. Esto implica que el docente fomenta una participación activa del estudiante y provee orientaciones y lineamientos que permitan construir su propio aprendizaje. Para lograr este objetivo, las distintas clases consideran una serie de estrategias metodológicas seleccionadas por el docente, tales como:

- Aprendizaje orientado a proyectos.
- Aprendizaje Colaborativo.
- Uso de entorno, framework y software

VII. PROCEDIMIENTOS DE EVALUACION DE APRENDIZAJES.

RESULTADO DE APRENDIZAJES	INDICADORES	INSTRUMENTO Y/O TÉCNICA EVALUATIVA	PONDERACIÓN (%)
1	<p>Crea Plan de proyecto.</p> <p>Establece criterios de elección del proceso software a seguir</p> <p>Selecciona un proceso de software.</p>	<p>Informe avance proyecto/ Rúbrica</p> <p>Presentación avance proyecto oral /Rúbrica</p>	20%
2	<p>Aplica estrategias para levantamiento de requisitos.</p> <p>Documenta los requisitos</p> <p>Construye modelos a partir del análisis de requisitos.</p> <p>Crea diseño de la solución a partir del análisis o refinamiento de modelos de análisis.</p> <p>Utiliza herramientas case para modelado.</p> <p>Utiliza prototipado.</p> <p>Utiliza técnicas de documentación y herramientas de apoyo.</p>	<p>Informe avance proyecto/ Rúbrica</p> <p>Presentación avance proyecto oral /Rúbrica</p>	40%

3	<p>Establece procedimientos de buenas prácticas para el desarrollo de software.</p> <p>Interpreta especificaciones de desarrollo de software.</p> <p>Aplica gestión de la configuración.</p> <p>Construye software a partir del uso de framework, lenguajes de programación, case, entornos.</p>	<p>Presentación avance proyecto oral /Rúbrica</p> <p>Informe documentación proyecto/Rúbrica</p>	20%
4	<p>Establece plan de pruebas</p> <p>Utiliza herramienta para hacer pruebas de software.</p> <p>Realiza depuración de software.</p> <p>Establece actividades y tareas para la puesta en marcha.</p> <p>Crea los artefactos y/ o productos para la puesta en marcha.</p> <p>Establece las consideraciones y proceso para el mantenimiento futuro del sistema.</p>	<p>Informe Final de proyecto/ Rúbrica</p> <p>Presentación y defensa Proyecto/ Rúbrica</p>	20%

VIII. RECURSOS DE INFRAESTRUCTURA

Laboratorio de computación, telón, Pizarra acrílica, proyector, servidor de aplicaciones, LMS-UCM

IX. RECURSOS BIBLIOGRÁFICOS

	Autor, Título, Editorial, Año de Edición	Biblioteca donde se encuentra	N° Libros Disponibles
BÁSICA OBLIGATORIA	- Suzanne Robertson, James Robertson, "Mastering the requirements process: getting requirements right", Addison-Wesley, 2013.	- - -	- - -

	<ul style="list-style-type: none"> - Jakob Freund, Bernd Rücker, Bernhard Hitpass, “BPMN 2.0 : manual de referencia y guía práctica”, Cuarta Edición, BMP Center, 2014, Chile. - Laurent Debrauwer, Fien Van Der Heyde, “UML 2 : iniciación, ejemplos y ejercicios corregidos”, Editorial Eni Ediciones, 2009 --- Ian Sommerville. Software engineering. Addison-Wesley, 2011. - Pierre Bourque, Richard E. Fairley. Guide to the Software Engineering Body of Knowledge (SWEBOK(R)): Version 3.0. IEEE Computer Society (Author), - Scrum: The Art of Doing Twice the Work in Half the Time Sep 30, 2014 by Jeff Sutherland and JJ Sutherland - Jez Humble and David Farley. Continuous delivery. 		
COMPLEMENTARIA	<ul style="list-style-type: none"> - Project Management Institute. 2013. A Guide to the Project Management Body of Knowledge (PMBOK Guide) Newtown Square, Pa: Project Management Institute. 2013. - 	- -	- -

X. OTROS RECURSOS

Nombre Recurso	Tipo de Recurso
<i>Servidor Web para hospedar software de la asignatura y proyectos de los estudiantes</i>	<i>Hardware</i>