

PROGRAMA DE ASIGNATURA

1. INFORMACIÓN GENERAL DE UBICACIÓN DE LA ASIGNATURA

UNIVERSIDAD:	MAYOR, REAL Y PONTIFICIA DE SAN FRANCISCO XAVIER DE CHUQUISACA
ÁREA:	Ciencias Sociales y Humanísticas (....)
	Ciencias de la Salud (....)
	Ciencias Tecnológicas y Agrarias (x)
	Ciencias Económicas y Financieras (....)
FACULTAD:	Facultad de Tecnología
CARRERA:	Ingeniería de Sistemas
ASIGNATURA:	Ingeniería de Software II
SIGLA:	SIS325
CURSO:	Noveno
SISTEMA:	Semestralizado (x)
	Anualizado (....)
DOCENTE:	Angel Hilmar Baspineiro Valverde
FECHA:	1 de Marzo del 2012

2. DESCRIPCIÓN Y JUSTIFICACIÓN DE LA ASIGNATURA

La asignatura está orientada a mejorar la comprensión de la Gestión de Proyectos de Software abordando actividades de gestión que están presentes en todo el ciclo de vida de desarrollo. De esta manera se prepara al Ingeniero de Sistemas para que pueda liderar y gerenciar proyectos siguiendo estándares y buenas prácticas de la Ingeniería.

La asignatura cierra el ciclo de las asignaturas dedicadas a la Ingeniería de Software dentro de la formación del estudiante y complementa muy bien todos los saberes y capacidades que van logrando desde primeros semestres en el área.

La importancia de la gestión de proyectos radica en saber administrar adecuadamente los recursos disponibles que generalmente son limitados, tener presente las restricciones tanto técnicas como de otra índole y aún así poder lograr los objetivos o metas del proyecto. Mencionar también que la gestión de proyectos busca tanto productividad como calidad en el proceso, como también en el producto, por tanto, es importante para la asignatura que el estudiante pueda llegar a manejar variables de gestión de manera objetiva.

3. RELACIONES DE LA ASIGNATURA

La asignatura cierra el ciclo de las asignaturas dedicadas a la Ingeniería de Software dentro de la formación del estudiante y complementa muy bien todos los saberes y capacidades que van logrando desde primeros semestres en el área. Verticalmente se relaciona directamente con la asignatura de SIS324 Ingeniería de Software I, y el resto de asignaturas por debajo de esta. Horizontalmente se relaciona con asignaturas como Gestión de Proyectos y Proyecto de Grado I. Esta asignatura es muy importante en general para estudiantes que realizan proyectos de grado relacionados con desarrollar software como elemento principal de su proyecto, ya que les permite

Planificar, Estimar, gestionar riesgos de manera mas objetiva, abordando mejor su Perfil de grado como el desarrollo de su proyecto.

4. OBJETIVO (s) GENERAL (es) DE LA ASIGNATURA

La asignatura tiene como objetivo general desarrollar en el estudiante conocimientos y habilidades sobre gestión de proyectos de software, que le permitan liderar el desarrollo de proyectos aplicando estándares, buenas prácticas y herramientas para la gestión, mejorando el grado de certidumbre que puede tener un gestor de proyectos respecto a las variables de gestión del mismo.

OBJETIVOS ESPECÍFICOS

- Introducir al estudiante en los conceptos básicos de la gestión de proyectos y su importancia como campo de estudio, así también de la caracterización de un proyecto.
- Desarrollar conocimientos y habilidades para la identificación y formulación de proyectos de software siguiendo buenas prácticas y el uso de métricas de la ingeniería de software para un mejor manejo de la variables de gestión.
- Desarrollar los conocimientos y habilidades para comprender, valorar y aplicar análisis de factibilidad de proyectos de software siguiendo estándares y buenas prácticas.
- Desarrollar los conocimientos y habilidades específicos para la gestión de riesgos de un proyecto de software siguiendo estándares y buenas prácticas.
- Fortalecer los conocimientos sobre el manejo de ciclos de vida para mejorar la planificación de proyectos considerando tanto planificación clásica, como también planificación adaptativa y extrema.
- Conocer y diferenciar planes de desarrollo y planes de soporte, haciendo énfasis en la importancia del aseguramiento de la calidad y gestión de la configuración del software.
- Desarrollar habilidades para el uso de herramientas o técnicas orientadas al aseguramiento de la calidad y gestión de la configuración del software.
- Desarrollar los conocimientos y habilidades para hacer estimacion de proyectos de software orientados a mejorar la certidumbre en la planificación de proyectos tanto de manera predictiva como tambien ágil.
- Introducir al estudiante en el manejo de modelos y estándares de calidad tanto para el proceso como para el producto de tal forma que a futuro el pueda incorporar los mismos en el desarrollo de software.

5. CONTENIDOS MÍNIMOS – EJES DE APRENDIZAJE

Conocimientos, habilidades y valores

Tema 1: INTRODUCCIÓN A LA GESTIÓN DE PROYECTOS DE SOFTWARE		
Objetivo particular : Introducir al estudiante en los conceptos básicos de la gestión de proyectos y su importancia como campo de estudio		
Sistema de conocimientos	Sistema de habilidades	Sistema de valores
<p>➤ Conoce sobre la importancia de la gestión de proyectos aplicada al software, la variables de gestión que normalmente intervienen, el manejo de métricas y el alcance de las actividades de gestión.</p> <p>Contenidos mínimos:</p> <ul style="list-style-type: none"> ➤ Que es la ingeniería de software ➤ Importancia de la ingeniería de software ➤ Modelos de procesos ➤ Costos de la ingeniería de software. ➤ Paradigmas, procesos, métodos y herramientas ➤ Que es un proyecto ➤ Caracterización de un proyecto ➤ Actividades de gestión frente a actividades técnicas del desarrollo. ➤ Productividad y Calidad ➤ Métricas del software ➤ Variables de la gestión de proyectos de software 	<ul style="list-style-type: none"> ➤ Diferencia diferentes modelos de procesos de desarrollo ➤ Diferencia diferentes tipos de métricas ➤ Caracteriza a las variables de gestión. ➤ Describe la taxonomía de un proyecto a través de un ejemplo ➤ Describe y diferencia de manera inicial, diferentes tipos de actividades involucradas en la gestión de proyectos. 	<ul style="list-style-type: none"> ➤ Demuestra responsabilidad y puntualidad en la presentación de trabajos y asistencia a clases. ➤ Se concientiza sobre la importancia de aplicar ingeniería al software. ➤ Demuestra interés por aprender desde el punto de vista del perfil de su profesión.

Tema 2: FACTIBILIDAD DE PROYECTOS DE SOFTWARE		
Objetivo particular : Desarrollar los conocimientos y habilidades para comprender, valorar y aplicar análisis de factibilidad de proyectos de software siguiendo estándares y buenas prácticas.		
Sistema de conocimientos	Sistema de habilidades	Sistema de valores
<p>➤ Conoce sobre modelos de análisis de factibilidad, factibilidad tecnológica, operativa y económica, costo beneficio, costo efectividad, e indicadores de evaluación económica, haciendo énfasis en ROI</p> <p>Contenidos mínimos:</p> <ul style="list-style-type: none"> ➤ Que es factibilidad de proyectos ➤ Modelos de análisis de factibilidad ➤ Factibilidad Técnica o Tecnológica ➤ Factibilidad Operativa ➤ Factibilidad Económica ➤ Costo Beneficio vs Costo Efectividad ➤ Indicadores para la evaluación económica de un proyecto ➤ ROI aplicado a desarrollo de Software 	<ul style="list-style-type: none"> ➤ Caracteriza diferentes formas de evaluar la factibilidad ➤ Diferencia entre costo-beneficio y costo-efectividad ➤ Es capaz de hacer un análisis de costos de un proyecto de software a partir de un caso de estudio. ➤ Aplica formulas diferentes según el caso de evaluación económica ➤ Diferencia las ventajas y desventajas de cada técnica de evaluación económica aplicada al software. 	<ul style="list-style-type: none"> ➤ Se fomenta la capacidad de análisis y síntesis en diferentes prácticas ➤ Se fomenta el trabajo colaborativo ➤ Se fomenta las capacidades orales y de comprensión de un tema

Tema 3: ANALISIS DE RIESGOS DE UN PROYECTO DE SOFTWARE		
Objetivo particular : Desarrollar los conocimientos y habilidades específicos para la gestión de riesgos de un proyecto de software siguiendo estándares y buenas prácticas.		
Sistema de conocimientos	Sistema de habilidades	Sistema de valores
<p>Conoce un proceso para la gestión de riesgos siguiendo estándares y buenas prácticas para identificar, evaluar, priorizar, controlar riesgos de un proyecto de software.</p> <p>Contenidos mínimos:</p> <ul style="list-style-type: none"> ➤ Definiciones ➤ Por que existen los riesgos? ➤ Un proceso de análisis de riesgos ➤ Clasificaciones de Riesgos 	<ul style="list-style-type: none"> ➤ Comprende y describe un proceso de gestión de riesgos ➤ Precisa cada etapa de la gestión de riesgos. ➤ Identifica en base a plantillas riesgos asociados a un caso de estudio. ➤ Realiza evaluación , priorización y control de riesgos utilizando buenas prácticas. ➤ Documenta adecuadamente un 	<ul style="list-style-type: none"> ➤ Toma conciencia de la importancia de gestionar riesgos en proyectos reales. ➤ Se fomenta la capacidad de análisis y síntesis en diferentes prácticas ➤ Se fomenta el trabajo colaborativo ➤ Se fomenta las participación individual y opinión propia ➤ Se fomenta la discusión

<ul style="list-style-type: none"> ➤ Identificación de riesgos ➤ Evaluación del riesgo ➤ Matriz de riesgos ➤ Priorización de riesgos ➤ Niveles de Referencia para la evaluación de riesgos ➤ Gestionar, controlar, mitigar el riesgo ➤ Gerencia de Riesgos IEEE 1540-2001 	proceso de gestión de riesgos.	constructiva
--	--------------------------------	--------------

Tema 4: PLANIFICACIÓN DE PROYECTOS DE SOFTWARE

Objetivo particular : Fortalecer los conocimientos sobre el manejo de ciclos de vida para mejorar la planificación del desarrollo de proyectos considerando tanto planificación clásica, como también planificación adaptativa y extrema.

Sistema de conocimientos	Sistema de habilidades	Sistema de valores
<p>Conoce y comprende sobre diferentes ciclos de vida para el desarrollo o procesos de desarrollo, tipos de planificación, procesos ágiles para la planificación, y estructuras orgánicas de grupos de desarrollo de software</p> <p>Contenidos mínimos:</p> <ul style="list-style-type: none"> ➤ Plan del proyecto ➤ Objetivos de la Planificación del desarrollo ➤ Planes genéricos vs. Planes Detallados ➤ Planes de desarrollo vs Ciclos de Vida ➤ Planificación Clásica frente a Planificación Adaptativa y Extrema ➤ SCRUM como proceso y marco de trabajo ágil. ➤ RUP como proceso y marco de trabajo no tan ágil. ➤ Otros procesos ágiles. 	<ul style="list-style-type: none"> ➤ Diferencia ciclos de vida y puede elegir el que mejor se adecua a un caso de estudio ➤ Diferencia los planes detallados de los planes genéricos ➤ Caracteriza los principales elementos de los procesos ágiles tomando como ejemplo a SCRUM ➤ Define un plan basado en un proceso SCRUM otro tipo de procesos ➤ Reconoce y diferencia roles en diferentes marcos de trabajo ➤ Define tareas de un plan en base al proceso elegido y marco de trabajo. ➤ Utiliza de manera básica herramientas automatizadas para la planificación de proyectos. 	<ul style="list-style-type: none"> ➤ Se fomenta la capacidad de análisis a través de casos de estudio ➤ Toma interés por resolver problemas propios del trabajo en equipo ➤ Se refuerza el sentido del compromiso con un proyecto o equipo. ➤ Responsabilidad en la presentación de tareas.

Tema 5: PLANES DE SOPORTE

Objetivo particular : Conocer y diferenciar planes de desarrollo y planes de soporte, haciendo énfasis en la importancia del aseguramiento de la calidad y gestión de la configuración del software desarrollando habilidades para el uso de herramientas o técnicas orientadas a estas actividades.

Sistema de conocimientos	Sistema de habilidades	Sistema de valores
<p>Contenidos mínimos:</p> <ul style="list-style-type: none"> ➤ Aseguramiento de la calidad : definiciones ➤ Principales componentes de un programa de calidad ➤ Actividades de aseguramiento de la calidad ➤ Costos de desarrollo vs defectos ➤ Revisiones formales e informales ➤ Pruebas del software ➤ Verificación y Validación del Software ➤ Estándares para aseguramiento de la calidad ➤ Herramientas para la AQS. ➤ Administración de la configuración del Software: Definiciones ➤ Actividades fundamentales ➤ Control de revisiones ➤ Administración de cambios ➤ Control de versiones ➤ Estándares para Administración de la Configuración del Software ➤ Herramientas para la ACS. 	<ul style="list-style-type: none"> ➤ Comprende y aplica las actividades de un plan de AQS ➤ Caracteriza cada tipo de revisión ➤ Utiliza herramientas para hacer pruebas del software ➤ Describe un plan de pruebas y diferencia diferentes tipos de pruebas ➤ Utiliza herramientas para hacer control de versiones ➤ Aplica actividades de AQS a un caso de estudio ➤ Diferencia entre V&V, Pruebas y AQS ➤ Aplica AQS y ACS en procesos ágiles. ➤ Integra actividades de AQS y ACS en planes de desarrollo 	<ul style="list-style-type: none"> ➤ Se fomenta el trabajo colaborativo ➤ Se fomenta la capacidad de análisis a través de casos de estudio ➤ Se refuerza el sentido del compromiso con un proyecto o equipo. ➤ Muestra interés por manejar herramientas para el trabajo colaborativo. ➤ Responsabilidad en la presentación de tareas.

Tema 6: ESTIMACIÓN DE PROYECTOS DE SOFTWARE

Objetivo particular : Desarrollar los conocimientos y habilidades para hacer estimación de proyectos de software orientados a mejorar la certidumbre en la planificación de proyectos tanto de manera predictiva como también ágil.

Sistema de conocimientos	Sistema de habilidades	Sistema de valores
Contenidos mínimos: <ul style="list-style-type: none"> ⇒ Introducción a la estimación ⇒ Tipos de modelos de estimación ⇒ Tamaño del software vs. Esfuerzo ⇒ Métricas del tamaño ⇒ Estimación Predictiva vs Estimación Ágil ⇒ COCOMO II ⇒ Estimación Paramétrica ⇒ Velocidad de desarrollo ⇒ Estimación del Coste. ⇒ Herramientas automatizadas para la estimación ⇒ Tiempo, esfuerzo y Staffing. 	<ul style="list-style-type: none"> ⇒ Reconoce y diferencia diferentes tipos de modelos de estimación ⇒ Utiliza estimación paramétrica basada en parámetros de velocidad para procesos ágiles ⇒ Utiliza estimación predictiva basada en COCOMO II ⇒ Describe las principales características de las estaciones del modelo COCOMO II ⇒ Aplica técnicas para la estimación del tamaño de proyectos de sw ⇒ Aplica cajas de tiempo para la planificación de iteraciones en procesos ágiles ⇒ Utiliza herramientas automatizadas para la estimación del esfuerzo, tiempo y costo. 	<ul style="list-style-type: none"> ⇒ Se fomenta el trabajo colaborativo ⇒ Se fomenta la capacidad de análisis a través de casos de estudio ⇒ Se refuerza el sentido del compromiso con un proyecto o equipo. ⇒ Muestra interés por manejar herramientas para el trabajo colaborativo. ⇒ Responsabilidad en la presentación de tareas.

Tema 7: CALIDAD DEL SOFTWARE

Objetivo particular : Introducir al estudiante en el manejo de modelos y estándares de calidad tanto para el proceso como para el producto de tal forma que el pueda incorporar los mismos en el desarrollo de software..

Sistema de conocimientos	Sistema de habilidades	Sistema de valores
Contenidos mínimos: <ul style="list-style-type: none"> ⇒ Ideas Introductorias. ⇒ Calidad en el proceso vs calidad en el producto ⇒ Modelos o Estándares para la calidad en el producto ⇒ Modelos o Estándares para la calidad en el producto 	<ul style="list-style-type: none"> ⇒ Diferencia entre calidad del producto y calidad en el proceso ⇒ Define diferentes atributos de calidad de alto nivel ⇒ Caracteriza diferentes modelos de calidad. 	<ul style="list-style-type: none"> ⇒ Se concientiza sobre la importancia de la calidad en el software como elemento clave para la competitividad ⇒ Se fomenta las capacidades orales y de comprensión de un tema ⇒ Responsabilidad en la presentación de tareas.

6. DISTRIBUCIÓN DEL FONDO DE TIEMPO

Nº	Tema	Horas Teóricas	Horas Prácticas			Horas laboratorio	Horas virtuales	Horas evaluación	Total horas (por tema)
			Taller	Investigación	Extensión				
1	INTRODUCCIÓN A LA GESTIÓN DE PROYECTOS DE SOFTWARE	4	4	2			4	1	15
2	FACTIBILIDAD DE PROYECTOS DE SOFTWARE	4	4	2			2	2	14
3	ANÁLISIS DE RIESGOS DE UN PROYECTO DE SOFTWARE	4	4	2			4	4	18
4	PLANIFICACIÓN DE PROYECTOS DE SOFTWARE	6	6	2			6	4	24
5	PLANES DE SOPORTE	4	6	2			6	4	22
6	ESTIMACIÓN DE PROYECTOS DE SOFTWARE	6	6	4			6	4	26
7	CALIDAD DEL SOFTWARE	4	2	4			2	2	14
TOTAL HORAS SEMESTRE									133
TOTAL HORAS ANUALES									266

7. CRONOGRAMA**7.1 Cronograma de plan temático y actividades**

Nº	Tema	Semanas																			
		1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20
1	INTRODUCCIÓN A LA GESTIÓN DE PROYECTOS DE SOFTWARE																				
2	FACTIBILIDAD DE PROYECTOS DE SOFTWARE																				
3	ANÁLISIS DE RIESGOS DE UN PROYECTO DE SOFTWARE																				
4	PLANIFICACIÓN DE PROYECTOS DE SOFTWARE																				
5	PLANES DE SOPORTE																				
6	ESTIMACIÓN DE PROYECTOS DE SOFTWARE																				
7	CALIDAD DEL SOFTWARE																			EF	EF

8.2 Cronograma de evaluaciones (parciales y finales)

Nº	Evaluación	Fechas	
		Sem 1	Sem2
	Primer parcial (temas 1 a 3)	9/abr/2012	29/sep/2012
	Segundo parcial (temas 4 a 6)	11/jun/2012	11/nov/2012
	Evaluación final	25/jun/2012	1/dic/2012
	Examen de segunda instancia	9/jul/2012	8/dic/2012

8. INDICACIONES METODOLÓGICAS Y DE ORGANIZACIÓN

El enfoque pedagógico adoptado en la asignatura es constructivista, en el sentido de que permite aprender compartiendo conocimientos y significancias, estimulando el trabajo cooperativo en grupo.

Se proponen las siguientes estrategias didácticas en el proceso de enseñanza aprendizaje, los cuáles serán elegibles en función a las circunstancias que se presentan semestre a semestre.

Estrategias para asimilar conocimientos:

- ➡ Conferencia Magistral
- ➡ Método de preguntas
- ➡ Lecturas y mapas conceptuales
- ➡ Talleres sobre herramientas
- ➡ Investigación y Disertaciones

Actualmente ya se está haciendo uso de plataformas virtuales para el PEA en muchas Universidades del país, sin embargo la Universidad de San Francisco Xavier aún esta inmersa en un modelo netamente presencial, por lo que, las actividades didácticas tienen como escenario principal el aula. Sin embargo la asignatura utilizará el e-campus que se esta desarrollando en la Universidad para Distribución de materiales previas a las clases, actividades de coordinación, informe, y también actividades de autoevaluación por parte del estudiante.

Dentro de las estrategias para desarrollar habilidades, las cuales están orientadas al trabajo individual mas que todo grupal colaborativo, se proponen las siguientes:

- ➡ Aprendizaje basado en problemas
- ➡ Método de proyectos a nivel grupal
- ➡ Prácticas en laboratorio

Estás ultimas actividades pueden ser desarrolladas dentro del aula, como así también fuera del aula como actividades no presenciales.

También se presta atención a los aspectos que el estudiante debe desarrollar como persona, mas que todo en el sentido de habilidades sociales en el trabajo en grupo: compromiso con el grupo, interelación, desarrollo personal, creatividad, trabajo bajo presión, competitividad.

9. RECURSOS DIDÁCTICOS

Dentro de los medios a utilizar están principalmente :

- ➡ Uso de diapositivas y medios audiovisuales
- ➡ El pizarrón
- ➡ Documentos de casos de estudio
- ➡ Página de la materia en el eCampus Virtual
- ➡ Internet
- ➡ Libros tanto digitalizados como los no digitales.
- ➡ Fotografías y grabaciones, como también el uso de videos.
- ➡ Agendas grupales (Prácticas desarrolladas paso a paso)

10. ACTIVIDADES DE INVESTIGACIÓN Y/O INTERACCIÓN

Investigación descriptiva y exploratoria mas que todo en temas como planificación, planes de soporte, estimación y calidad del software

11. SISTEMA DE EVALUACIÓN DEL APRENDIZAJE

Pese a nuevos enfoques que podrían ser aplicados en la evaluación como los de la Formación Basada en Competencias que utiliza el término de “valoración” por ejemplo, mas que de evaluación, dando énfasis a la valoración continua del desempeño en el logro de las competencias deseadas, es inevitable tener que adecuarse a ciertas reglamentaciones Universitarias de San Francisco Xavier como es por ejemplo las formas de evaluación establecidas, de donde se eligió el Código SEA “E” que establece:

EVALUACIÓN SOBRE PRÁCTICA INDIVIDUAL Y GRUPAL	35%
EVALUACIONES PARCIALES	35%
EVALUACIÓN FINAL	30%

Se quiere destacar que se pretende dar mas valor al proceso de adquisición de conocimientos y mas que todo habilidades, tomando muy en cuenta las actividades prácticas tanto presenciales como no presenciales. Por tanto se da mas valor al proceso que a la evaluación final.

La naturaleza de las evaluaciones parciales será mas práctica que teórica, mientras que la evaluación final hará mas énfasis en los conocimientos.

Las evaluaciones están uniformemente espaciadas en las 18 semanas y consideran:

1era. Evaluación Parcial	Temas 1 a 3
2da.. Evaluación Parcial	Temas 4 a 6
Evaluación final*	Temas 1 a 7

**La Universidad SFX reglamenta que las pruebas oficiales deben ser escritas, sin embargo es posible que en esta asignatura las pruebas finales puedan ser Orales. Es una opción también la entrega de un trabajo final cuando es posible utilizar el Método de Proyectos con Instituciones del medio, o dentro de la misma Universidad.*

12. BIBLIOGRAFÍA

- ➔ "Ingeniería de Software", Iam Sommerville, 7ª Edición, Ed. Addison –Wesley, 2005.
- ➔ "Software Project Management", Bob Hughes & Mike Cotterel, 2ª Edición, Ed. McGrawHill, 1999.
- ➔ "Ingeniería de Software: Una Perspectiva Orientada a Objetos", E Braude, Ed. Alfa Omega, 2003.
- ➔ "Ingeniería de Software": Teoría y Práctica ", Shari L Pfleeger, Ed. Prentice Hall, 2003.
- ➔ "Ingeniería de Software", R.Pressman, 6ta Edición, Ed. MacGrawHill, 2006
- ➔ "Scrum Manager: gestión de proyectos", Rev 1.4.0, ScrumManager.Net, 2011.
- ➔ "Scrum y XP desde las trincheras", Henrik Kniberg, ed. INFOQ, 2007
- ➔ " Scrum", Ken Schwaber y Jeff Sutherland, Scrum.org, 2010
- ➔ "Cocoma II Model Definition Manual", B.Boehm,2002

Datos complementarios:

Programa elaborado por primera vez

Modificado X

Actualización de bibliografía

Fecha de entrega: 10 Agosto 2012

Firma del Docente:

Firma del (a) Director (a) de Carrera: