




UNIVERSIDAD DE GUAYAQUIL  
VICERRECTORADO DE FORMACIÓN ACADÉMICA Y PROFESIONAL

				UNIVERSIDAD DE GUAYAQUIL SYLLABUS									
A: DATOS INFORMATIVOS													
Facultad:		CIENCIAS MATEMÁTICAS Y FÍSICAS					Dominio:		CIENCIAS BÁSICAS, BIOCONOCIMIENTO Y DESARROLLO INDUSTRIAL				
Carrera:		SOFTWARE											
Asignatura:		INTRODUCCIÓN A LA INGENIERÍA DE SOFTWARE		Código:	004		UOC:	UNIDAD BÁSICA		Campo Formación:		PRAXIS PROFESIONAL	
Semestre:		PRIMERO		Paralelo:	SOF-S-MA-1-1			Horario:	Lunes: 08:00 - 10:00 Jueves: 09:00 - 11:00				
					SOF-S-MA-1-2				Martes: 07:00 - 09:00 Jueves: 07:00 - 09:00				
					SOF-S-MA-1-4				Jueves: 12:00 - 14:00 Viernes: 12:00 - 14:00				
					SOF-S-MA-1-7				Martes: 12:00 - 14:00 Miércoles: 10:00 - 12:00				
					SOF-S-MA-1-8				Lunes: 12:00 – 14:00 Miércoles: 12:00 – 14:00				
					SOF-S-NO-1-3				Martes: 18:00 - 20:00 Jueves: 20:00 - 22:00				
Plan de estudios:		N° Créditos:	3	Horas componente docencia:	48		Horas componente de práctica y experimentación:		16		Horas componente trabajo autónomas:	56	
Prerrequisitos:		N/A											
Período Académico:		2019								Ciclo:		II	
Docente:		BOTTO TOBAR MIGUEL ANGEL				Título de posgrado:							
		AVILES MONROY JORGE ISAAC					MASTER UNIVERSITARIO EN INGENIERIA DE SOFTWARE						
		SORNOZA MOREIRA JIMMY IGNACIO					MAGISTER EN SISTEMAS DE INFORMACIÓN GERENCIAL						

## B: JUSTIFICACIÓN DEL CONOCIMIENTO DEL SYLLABUS EN EL CAMPO DE FORMACIÓN

### Breve justificación de los contenidos del Syllabus:

La cátedra de Introducción de Ingeniería de Software tiene como base que el estudiantado maneje los conceptos generales de las etapas de un sistema y las diferentes alternativas de desarrollo del mismo, integrando aspectos éticos, sociales, legales y económicos.

### Objetivo General:

Desarrollar una visión general de la aplicación de principios de ingeniería de software robustos, haciendo énfasis en las etapas de especificación y de diseño para la construcción de soluciones orientada a obtener software de calidad.

Aportes teóricos	Aportes metodológicos	Aporte a la comprensión de los problemas del campo profesional	Contextos de aplicación
<ol style="list-style-type: none"> <li>1. Conocer las actividades primordiales de la Ingeniería de Software.</li> <li>2. Lograr la capacidad para valorar las necesidades del cliente y especificar los requisitos de software para satisfacer las necesidades, armonizando los objetivos mediante la búsqueda de compromisos aceptables de costo, tiempo, existencia de sistemas ya desarrollados y de las propias organizaciones</li> <li>3. Lograr la capacidad para desarrollar, mantener y evaluar servicios y sistemas de software que satisfagan todos los requisitos del usuario y se comporten de forma fiable, eficiente y cumplan normas de calidad, aplicando las teorías, principios, métodos y prácticas de la Ingeniería del Software.</li> <li>4. Lograr la capacidad para diseñar soluciones apropiadas en uno o más dominios de aplicación utilizando métodos de la ingeniería del software que integren aspectos éticos, sociales, legales y económicos.</li> </ol>	<p>El aporte metodológico en el que se ha enmarcado la Carrera de Ingeniería de Software dentro de los lineamientos del constructivismo, donde el estudiante construye su propio conocimiento a través de la interrelación con los materiales, el entorno y los medios tecnológicos que le permitan relacionar la teoría con la práctica</p>	<p>La Ingeniería en Software se rige por modelos o patrones de desarrollo en los cuales se apoya para la realización de software, desde un enfoque sistemático, disciplinado y cuantificable al desarrollo, operación y mantenimiento de software, la misma que integra matemáticas, ciencias de la computación y prácticas cuyos orígenes se encuentran en la ingeniería.</p>	<p>La Ingeniería de software ofrece un contexto de aplicación muy amplio en donde el campo de acción el mismo que puede ser en:</p> <ul style="list-style-type: none"> <li>✚ Análisis de Requerimientos</li> <li>✚ Diseño de Proyectos</li> <li>✚ Desarrollar la del Sistema</li> <li>✚ Supervisión y Prácticas del Sistema.</li> <li>✚ Elaboración de la Documentación.</li> </ul>

C: PROPÓSITOS Y APORTES AL PERFIL DE EGRESO				
Propósitos del aprendizaje del syllabus relacionado con el campo de estudio y objetivos de la carrera:	Aportes al perfil de egreso: Capacidades integrales y/o competencias, logros o resultados de aprendizaje			
	Genéricas de la UG.	Específicas de la carrera.	Logros de aprendizaje.	Ámbito.
Permitir el uso de diversas herramientas en el Análisis, Diseño y Modelado de los sistemas, así como su funcionalidad, además de incorporar la documentación respectiva para el conocimiento y uso del sistema.	Gestionar eficientemente las tecnologías de la información y comunicación para integrar procesos de conectividad en la práctica profesional.	Trabajar en equipo en el logro de objetivos y metas determinadas en el área de su profesión o en escenarios multidisciplinares Organizar e integrar el conocimiento y asumir con un pensamiento sistémico las transformaciones actuales, adoptando enfoques multidisciplinares para la comprensión de los problemas que presenta la sociedad.	Propone sistemas informáticos mediante el análisis, diseño y desarrollo de procesos de información, aplicando diferentes métodos y métricas para el desarrollo de software Demuestra dominio de habilidades y destrezas interdisciplinares a través de la elaboración, redacción y difusión de proyectos que involucren soluciones informáticas.	Conocimientos.
Aprender a diseñar metas realizables a corto, mediano y largo plazo. A Conocimientos de informática relativos al ámbito de estudio	Evaluar la factibilidad de alternativas de solución y retroalimentación de problemas en el ámbito de la profesión, logrando participar activamente y con responsabilidad en diferentes contextos sociales, en el logro del bien común.	Evaluar la factibilidad de alternativas de solución y retroalimentación de problemas en el ámbito de la profesión, logrando participar activamente y con responsabilidad en diferentes contextos sociales, en el logro del bien común que le permitan aplicar indicadores de gestión para alcanzar resultados de excelencia.	Identificar variables, actores y escenarios que definan los requerimientos de las organizaciones públicas o privadas para la consecución de proyectos de software Analizar la incidencia y participación de la información obtenida en las etapas de levantamiento de información con la finalidad de traducir en un lenguaje sencillo y directo que aporte con la creación de un diseño de software	Habilidades.
Desarrollar métodos que permitan levantamiento de información de manera eficaz, compartiendo responsabilidades con sus compañeros, con enfoques de mejora continua en procesos de software.	Organizar e integrar el conocimiento y asumir con un pensamiento sistémico las transformaciones actuales, adoptando enfoques multidisciplinares para la comprensión de los problemas que presenta la sociedad.	Generar levantamientos de información a procesos manuales, manera metódica. Aplicar de manera eficaz técnicas de simulación de sistemas	Compañerismo: los ingenieros del software serán justos y apoyarán a sus compañeros. Competencia Personal: los ingenieros del software participarán en el aprendizaje continuo referente a la práctica de su profesión y promoverán un enfoque ético en la práctica de la profesión.	Valores y actitudes.

D: UNIDADES TEMÁTICAS O DE ANÁLISIS:							
Unidad #: <u>  1  </u>		Descripción: INTRODUCCIÓN DE LA INGENIERÍA DE SOFTWARE					
OBJETIVO: Caracterizar la fundamentación teórica del software mediante el análisis de la evolución y el proceso de ingeniería del mismo para identificar el ámbito general de la ingeniería de software.							
Contenidos: conocimientos a desarrollar.	Métodos, técnicas e instrumentos en función de las actividades de organización del aprendizaje.				Tiempo de aprendizaje.	Escenarios en función de los ambientes de aprendizaje.	Recursos didácticos.
	Componente de docencia.		Componente de prácticas de aplicación y experimentación de los aprendizajes.	Componente de aprendizaje autónomo.			
	Actividades de aprendizaje asistido por el profesor.	Actividades de aprendizaje colaborativo.					
<b>1.1 Fundamentos a la Ingeniería de Software</b> <ul style="list-style-type: none"><li>Definiciones y Objetivos de la Ingeniería de software.</li><li>La evolución del software</li><li>La crisis del software</li><li>El software en la actualidad</li></ul> <b>1.2 Proceso de Ingeniería de software</b> <ul style="list-style-type: none"><li>Factores de calidad del software.</li><li>Problemas en el desarrollo de software.</li><li>La ingeniería del software.</li><li>Visión general del proceso de ingeniería del software.</li><li>Responsabilidad ética y profesional en ingeniería del software.</li></ul> <b>1.3 Sistemas de información</b> <ul style="list-style-type: none"><li>Concepto</li><li>Información y datos</li><li>Tipos de Sistemas de información</li></ul>	Clase teórico práctica. Exposición dialogada.	Talleres. Resolución de caso de estudio. Exposición Grupal	Desarrollo de caso de estudio.	Deberes Consultas. Desarrollo de caso de estudio.	8 horas	Aulas/ Laboratorio de Computación	Laptop, Proyector, Pizarra, Marcador, Plataformas Educativas Gratuitas.

D: UNIDADES TEMÁTICAS O DE ANÁLISIS:							
Unidad #: <u>  2  </u>		Descripción: <b>MODELOS DE PROCESO DEL SOFTWARE</b>					
<b>Objetivo:</b> Identificar el ciclo de vida del software a través de los estándares IEEE para diferenciar los modelos de desarrollo de software							
Contenidos: conocimientos a desarrollar.	Métodos, técnicas e instrumentos en función de las actividades de organización del aprendizaje.				Tiempo de aprendizaje.	Escenarios en función de los ambientes de aprendizaje.	Recursos didácticos.
	Componente de docencia.		Componente de prácticas de aplicación y experimentación de los aprendizajes.	Componente de aprendizaje autónomo.			
	Actividades de aprendizaje asistido por el profesor.	Actividades de aprendizaje colaborativo.					
<b>2.1 Proceso del software</b> <ul style="list-style-type: none"><li>Definición de proceso de software.</li><li>Estándares relacionados con el proceso software: IEEE/EIA (ISO/IEC) 12207</li></ul> <b>2.2 El ciclo de vida del Software</b> <ul style="list-style-type: none"><li>Definición y etapas</li><li>Ámbito general de: Requerimientos, análisis, diseño y pruebas de software</li></ul> <b>2.3 Modelos de desarrollo: tradicionales vs alternativos</b> <ul style="list-style-type: none"><li>Concepto de modelo de Desarrollo de software</li><li>Modelos tradicionales: Cascada, prototipo y espiral.</li><li>Modelos ágiles: Scrum, Xp, Kanban, TDD(Test-Driven-Design)</li></ul>	Clase teórico práctica. Exposición dialogada.	Talleres. Resolución de caso de estudio. Exposición Grupal	Desarrollo de caso de estudio.	Deberes Consultas. Desarrollo de caso de estudio.	16 horas	Aulas de trabajo / Laboratorio de Computación	Laptop, Proyector, Pizarra, Marcador, Plataformas Educativas Gratuitas.



<b>3.3 Prueba del software.</b> <ul style="list-style-type: none"> <li>Objetivos de la prueba.</li> <li>Importancia de la prueba.</li> <li>Principios de la prueba.</li> <li>El proceso de prueba.</li> <li>Métodos de diseño de casos de prueba.</li> <li>Enfoque estructural.</li> <li>Enfoque funcional.</li> <li>Prueba de interfaces gráficas de usuario.</li> <li>Estrategias de prueba del software.</li> </ul>							
--	--	--	--	--	--	--	--

D: UNIDADES TEMÁTICAS O DE ANÁLISIS:							
Unidad #: <u>  4  </u>		Descripción: PROYECTO DE SOFTWARE					
Objetivo: Evaluar la viabilidad y riesgo de un proyecto de software a través de métricas y estimaciones para asegurar una adecuada planificación de proyectos de software.							
Contenidos: conocimientos a desarrollar.	Métodos, técnicas e instrumentos en función de las actividades de organización del aprendizaje.				Tiempo de aprendizaje.	Escenarios en función de los ambientes de aprendizaje.	Recursos didácticos
	Componente de docencia.		Componente de prácticas de aplicación y experimentación de los aprendizajes.	Componente de aprendizaje autónomo.			
	Actividades de aprendizaje asistido por el profesor.	Actividades de aprendizaje colaborativo.					
4.1 Introducción a proyectos de software. <ul style="list-style-type: none"><li>• Concepto de proyectos de software</li><li>• Características de los proyectos de software</li></ul>	Clase teórico práctica. Exposición dialogada.	Talleres. Resolución de caso de estudio. Exposición Grupal	Desarrollo de caso de estudio.	Deberes Consultas. Desarrollo de caso de estudio.	12 horas	Aulas de clase / laboratorio de Computación	Laptop, Proyector, Pizarra, Marcador, Plataformas Educativa Gratuitas.

<b>4.2 Planificación de proyectos de software.</b> <ul style="list-style-type: none"><li>• Recursos: Humano, Hardware y software.</li><li>• Métricas del software: Técnicas, calidad, Orientada a las personas, Orientada al tamaño, Orientada a la función.</li><li>• Estimación de proyectos: Modelo COCOMO.</li><li>• Identificación y análisis de riesgo.</li></ul>							
---	--	--	--	--	--	--	--



### E: EVALUACIÓN DE LOS APRENDIZAJES.

Sistema de evaluación de los aprendizajes en función de:		Actividades.	
Gestión formativa.	60 %	a) Trabajo participativo en clase, <input checked="" type="checkbox"/> b) Reportes de talleres y equipos colaborativos, <input checked="" type="checkbox"/> c) Controles de lectura, <input checked="" type="checkbox"/> d) Otros: (Detallar) _____ <input type="checkbox"/>	
Gestión práctica y autónoma.		a) Exposiciones individuales y grupales, <input checked="" type="checkbox"/> b) Demostración de uso directo de los acervos bibliotecarios o en red, <input type="checkbox"/> c) Trabajo de laboratorio, talleres, seminarios, <input checked="" type="checkbox"/> d) Ejercicios orales y escritos de técnica jurídica, <input type="checkbox"/> e) Prácticas diversas, incluyendo la de los laboratorios, <input checked="" type="checkbox"/> f) Trabajos de campo, <input type="checkbox"/> g) Trabajos individuales de lectura, análisis y aplicación, <input checked="" type="checkbox"/> h) Uso creativo y orientado de nuevas TICs y la multimedia, <input checked="" type="checkbox"/> i) Lectura crítica y análisis comparado de casos, <input type="checkbox"/> j) Asistencia y reporte de eventos académicos. <input checked="" type="checkbox"/> k) Otros: (Detallar) _____ <input type="checkbox"/>	
Acreditación y validación.	40 %	a) Exámenes orales y escritos teóricos, <input type="checkbox"/> b) Exámenes orales y escritos prácticos, <input type="checkbox"/> c) Sustentación de proyectos de investigación y casos prácticos. <input checked="" type="checkbox"/> d) Otros: (Detallar) <u>Examen Escrito</u> <input checked="" type="checkbox"/>	

F: BIBLIOGRAFÍA				
BÁSICA	No	Título de la obra.	Existencia en biblioteca.	Número de ejemplares.
	1	Un Enfoque Hacia la Ingeniería de Sistemas, Autor: Dougglas Hurtado Carmona, Editorial: Prentice Hall, Edición Cuarta	Sí	3
	2	Ingeniería de Software, Autor: Ian Sommerville, Editorial: Prentice Hall, Edición: Séptima y Novena	Sí	1
	3	Ingeniería de Software, Autor: Guillermo Pantaleo		
	4			
COMPLEMENTARIA	1	Ingeniería de Software de Roger Pressman, 6ta Edición	Sí	3
	2	Análisis y Diseño de Sistemas de Kendall & Kendall Edición: Séptima	Sí	2
	3			
	4			
SITIOS WEB	No	Dirección electrónica / URL		
	1	<a href="http://www.javiergarzas.com/">http://www.javiergarzas.com/</a>		
	2	<a href="http://garciagregorio.webcindario.com/ms/metricas_estimacion_planeacion.pdf">http://garciagregorio.webcindario.com/ms/metricas_estimacion_planeacion.pdf</a>		
	3	<a href="http://yaqui.mxl.uabc.mx/~molguin/as/IngSoft%201-4.pdf">http://yaqui.mxl.uabc.mx/~molguin/as/IngSoft%201-4.pdf</a>		
	4			

**G: FIRMAS DE RESPONSABILIDAD**

<b>Responsabilidad.</b>	<b>Nombre del responsable.</b>	<b>Firma.</b>	<b>Fecha entrega.</b>
Elaborado por:	Ing. Botto Tobar Miguel Angel, Msc.		
	Ing. Aviles Monroy Jorge Isaac, Msc.		
	Ing. Sornoza Moreira Jimmy Ignacio, Mgs		
Revisado por:	Ing. Jorge Luis Charco Aguirre, MSc.  Ing. Verónica Mendoza Morán, MSc.		
Aprobado por:	Ing. Gary Reyes, Mgs		
Secretario de la carrera:	Ab. Juan Chávez Atocha.		