



Universidad de Ingeniería y Tecnología
Escuela Profesional de
Ciencia de la Computación
Silabo del curso
Periodo Académico 2019-I

1. **Código del curso y nombre:** CS3901. Ingeniería de Software III (Electivo)
2. **Créditos:** 4
3. **Horas de Teoría y Laboratorio:** 2 HT; 4 HL; (Semanal)
4. **Profesor(es) del curso, email y horario de atención**

Atención previa coordinación con el profesor

5. **Bibliografía básica**

- [PM14] Roger S. Pressman and Bruce Maxim. *Software Engineering: A Practitioner's Approach*. 8th. McGraw-Hill, Jan. 2014.
- [Som10] Ian Sommerville. *Software Engineering*. 9th. Addison-Wesley, Mar. 2010.

6. **Información del curso**

- (a) **Breve descripción del curso** El desarrollo de software requiere del uso de mejores prácticas de desarrollo, gestión de proyectos de TI, manejo de equipos y uso eficiente y racional de frameworks de aseguramiento de la calidad, estos elemento son pieza clave y transversal durante todo el proceso productivo.

La construcción de software contempla la implementación y uso de procesos, métodos, modelos y herramientas que permitan lograr la realización de los atributos de calidad de un producto.

- (b) **Prerrequisitos:** CS2902. Ingeniería de Software II. (7^{mo} Sem)
- (c) **Tipo de Curso:** Electivo
- (d) **Modalidad:** Presencial

7. **Objetivos del curso.**

Competencias

- c) Diseñar, implementar y evaluar un sistema, proceso, componente o programa computacional para alcanzar las necesidades deseadas. (**Usar**)
- d) Trabajar efectivamente en equipos para cumplir con un objetivo común. (**Usar**)
- i) Utilizar técnicas y herramientas actuales necesarias para la práctica de la computación. (**Usar**)
- j) Aplicar la base matemática, principios de algoritmos y la teoría de la Ciencia de la Computación en el modelamiento y diseño de sistemas computacionales de tal manera que demuestre comprensión de los puntos de equilibrio involucrados en la opción escogida. (**Evaluar**)
- m) Transformar sus conocimientos del área de Ciencia de la Computación en emprendimientos tecnológicos. (**Evaluar**)
- o) Mejorar las condiciones de la sociedad poniendo la tecnología al servicio del ser humano. (**Usar**)



Objetivos de Aprendizaje

- Comprender y poner en práctica los conceptos fundamentales sobre la gestión de proyectos y manejo de equipos de software.
- Comprender los fundamentos de la gestión de proyectos, incluyendo su definición, alcance, y la necesidad de gestión de proyectos en la organización moderna.

- Los alumnos deben comprender los conceptos fundamentales de CMMI, PSP, TSP para que sean adoptados en los proyectos de software.
- Describir y comprender los modelos de aseguramiento de la calidad como marco clave para el éxito de los proyectos de TI.

8. Tópicos del curso

1. Evolución de Software
2. Gestión de Proyectos de Software
3. Gestión de Proyectos de Software
4. Procesos de Software
5. Estándares ISO/IEC

9. Metodología y sistema de evaluación

Metodología:

Sesiones Teóricas:

Las sesiones de teoría se llevan a cabo en clases magistrales donde se realizarán actividades que propicien un aprendizaje activo, con dinámicas que permitan a los estudiantes interiorizar los conceptos.

Sesiones de Laboratorio:

Para verificar que los alumnos hayan alcanzado el logro planteado para cada una de las unidades de aprendizaje, realizarán actividades que les permita aplicar los conocimientos adquiridos durante las sesiones de teoría y se les propondrá retos que permitan evaluar el desempeño de los alumnos.

Exposiciones individuales o grupales:

Se fomenta la participación individual y en equipo para exponer sus ideas, motivándolos con puntos adicionales en las diferentes etapas de la evaluación del curso.

Lecturas:

A lo largo del curso se proporcionan diferentes lecturas, las cuales son evaluadas. El promedio de las notas de las lecturas es considerado como la nota de una práctica calificada. El uso del campus virtual UTEC Online permite a cada estudiante acceder a la información del curso, e interactuar fuera de aula con el profesor y con los otros estudiantes.

Sistema de Evaluación:

10. Contenido



Unidad 1: Evolución de Software (12)	
Competences esperadas: C7, C11, C12, CS6	
Objetivos de Aprendizaje	Tópicos
<ul style="list-style-type: none"> • Identificar los problemas principales asociados con la evolución del software y explicar su impacto en el ciclo de vida del software [Familiarizarse] • Estimar el impacto del cambio de requerimientos en productos existentes de tamaño medio [Usar] • Usar refactorización en el proceso de modificación de un componente de software [Usar] • Estudiar los desafíos de mejorar sistemas en un entorno cambiante [Familiarizarse] • Perfilar los procesos de pruebas de regresión y su rol en el manejo de versiones [Familiarizarse] • Estudiar las ventajas y desventajas de diferentes tipos de niveles de confiabilidad [Familiarizarse] 	<ul style="list-style-type: none"> • Desarrollo de Software en el contexto de código grande pre existente <ul style="list-style-type: none"> – Cambios de software – Preocupaciones y ubicación de preocupaciones – <i>Refactoring</i> • Evolución de Software. • Características de Software mantenible. • Sistemas de Reingeniería. • Reuso de Software. <ul style="list-style-type: none"> – Segmentos de código – Bibliotecas y <i>frameworks</i> – Componentes – Líneas de Producto
Lecturas : [PM14], [Som10]	



Unidad 2: Gestión de Proyectos de Software (10)**Competences esperadas: C18, C19, CS7, CS9****Objetivos de Aprendizaje**

- Discutir los comportamientos comunes que contribuyen al buen funcionamiento de un equipo [Familiarizarse]
- Crear y seguir un programa para una reunión del equipo [Usar]
- Identificar y justificar las funciones necesarias en un equipo de desarrollo de software [Usar]
- Entender las fuentes, obstáculos y beneficios potenciales de un conflicto de equipo [Usar]
- Aplicar una estrategia de resolución de conflictos en un ambiente de equipo [Usar]
- Utilizar un método ad hoc para estimar el esfuerzo de desarrollo del software (ejemplo, tiempo) y comparar con el esfuerzo actual requerido [Usar]
- Listar varios ejemplos de los riesgos del software [Familiarizarse]
- Describir el impacto del riesgo en el ciclo de vida de desarrollo de software [Familiarizarse]
- Describir las diferentes categorías de riesgo en los sistemas de software [Familiarizarse]
- Demostrar a través de la colaboración de proyectos de equipo los elementos centrales de la construcción de equipos y gestión de equipos [Usar]
- Describir como la elección de modelos de procesos afectan la estructura organizacional de equipos y procesos de toma de decisiones [Familiarizarse]
- Crear un equipo mediante la identificación de los roles apropiados y la asignación de funciones a los miembros del equipo [Usar]
- Evaluar y retroalimentar a los equipos e individuos sobre su desempeño en un ambiente de equipo [Usar]
- Usando un software particular procesar, describir los aspectos de un proyecto que necesita ser planeado y monitoreado, (ejemplo, estimar el tamaño y esfuerzo, un horario, reasignación de recursos, control de configuración, gestión de cambios, identificación de riesgos en un proyecto y gestión) [Familiarizarse]

Tópicos

- La participación del equipo:
 - Procesos elemento del equipo, incluyendo responsabilidades de tarea, la estructura de reuniones y horario de trabajo
 - Roles y responsabilidades en un equipo de software
 - Equipo de resolución de conflictos
 - Los riesgos asociados con los equipos virtuales (comunicación, la percepción, la estructura)
- Estimación de esfuerzo (a nivel personal)
- Riesgo.
 - El papel del riesgo en el ciclo de vida
 - Categorías elemento de riesgo, incluyendo la seguridad, la seguridad, mercado, finanzas, tecnología, las personas, la calidad, la estructura y el proceso de
- Gestión de equipos:
 - Organización de equipo y la toma de decisiones
 - Roles de identificación y asignación
 - Individual y el desempeño del equipo de evaluación
- Gestión de proyectos:
 - Programación y seguimiento de elementos
 - Herramientas de gestión de proyectos
 - Análisis de Costo/Beneficio

Asignaturas : [PM14], [Som10]

Unidad 3: Gestión de Proyectos de Software (8)	
Competences esperadas: C18, C19, CS7, CS9	
Objetivos de Aprendizaje	Tópicos
<ul style="list-style-type: none"> • Realizar el seguimiento del progreso de alguna etapa de un proyecto que utiliza métricas de proyectos apropiados [Usar] • Comparar las técnicas simples de tamaño de software y estimación de costos [Usar] • Usar una herramienta de gestión de proyectos para ayudar en la asignación y rastreo de tareas en un proyecto de desarrollo de software [Usar] • Describir el impacto de la tolerancia de riesgos en el proceso de desarrollo de software [Evaluar] • Identificar riesgos y describir enfoques para manejar riesgos (evitar, aceptar, transferir, mitigar) y caracterizar fortalezas y defectos para cada uno [Familiarizarse] • Explicar cómo el riesgo afecta las decisiones en el proceso de desarrollo de software [Usar] • Identificar los riesgos de seguridad para un sistema de software [Usar] • Demostrar un enfoque sistemático para la tarea de identificar los peligros y riesgos en una situación particular [Usar] • Aplicar los principios básicos del manejo de riesgos en una variedad de escenarios simples incluyendo una situación de seguridad [Usar] • Dirigir un análisis de costo/beneficio para el enfoque de mitigación de riesgos [Usar] • Identificar y analizar alguno de los riesgos para un sistema entero que surgen de aspectos distintos del software [Usar] 	<ul style="list-style-type: none"> • Software de medición y técnicas de estimación. • Aseguramiento de la calidad del software y el rol de las mediciones. • Riesgo. <ul style="list-style-type: none"> – Identificación de riesgos y gestión. – Análisis riesgo y evaluación. – La tolerancia al riesgo (por ejemplo, riesgo adverso, riesgo neutral, la búsqueda de riesgo) – Planificación de Riesgo • En todo el sistema de aproximación al riesgo, incluyendo riesgos asociados con herramientas.
Lecturas : [PM14], [Som10]	



Unidad 4: Procesos de Software (12)**Competencias esperadas: C7, C13, C19, CS6, CS7**

Objetivos de Aprendizaje	Tópicos
<ul style="list-style-type: none">• Describa cómo el software puede interactuar y participar en varios sistemas, incluyendo la gestión de información, integración, control de procesos y sistemas de comunicaciones [Usar]• Describir las ventajas y desventajas relativas entre varios modelos importantes de procesos (por ejemplo, la cascada, iterativo y ágil) [Usar]• Describir las diferentes prácticas que son componentes clave de los diversos modelos de procesos [Usar]• Diferenciar entre las fases de desarrollo de software [Usar]• Describir cómo la programación en grandes equipos difiere de esfuerzos individuales con respecto a la comprensión de una gran base de código, lectura de código, comprensión de las construcciones, y comprensión de contexto de cambios [Usar]• Explicar el concepto de ciclo de vida del software y proporcionar un ejemplo que ilustra sus fases incluyendo los entregables que se producen [Usar]• Comparar varios modelos comunes de procesos con respecto a su valor para el desarrollo de las clases particulares de sistemas de software, teniendo en cuenta diferentes aspectos tales como, estabilidad de los requisitos, tamaño y características no funcionales [Usar]• Definir la calidad del software y describir el papel de las actividades de aseguramiento de la calidad en el proceso de software [Usar]• Describir el objetivo y similitudes fundamentales entre los enfoques de mejora de procesos [Usar]• Comparar varios modelos de mejora de procesos, tales como CMM, CMMI, CQI, <i>Plan-Do-Check-Act</i>, o ISO9000 [Usar]• Evaluar un esfuerzo de desarrollo y recomendar cambios potenciales al participar en la mejora de procesos (usando un modelo como PSP) o involucración en una retrospectiva de un proyecto [Usar]• Explicar el papel de los modelos de madurez de procesos en la mejora de procesos [Usar]• Describir varias métricas de procesos para la evaluación y el control de un proyecto [Usar]• Usar las medidas en proyecto para describir el estado actual de un proyecto [Usar]	<ul style="list-style-type: none">• Consideraciones a nivel de sistemas, ejem., la interacción del software con su entorno.• Introducción a modelos del proceso de software (e.g., cascada, incremental, ágil):<ul style="list-style-type: none">– Actividades con ciclos de vida de software.• Programación a gran escala versus programación individual.• Evaluación de modelos de proceso de software.• Conceptos de calidad de software.• Mejoramiento de procesos.• Modelos de madurez de procesos de software.• Mediciones del proceso de software.
Lecturas : [PM14], [Som10]	



Unidad 5: Estándares ISO/IEC (6)	
Competences esperadas: C7, C13, C19, CS6, CS7	
Objetivos de Aprendizaje	Tópicos
<ul style="list-style-type: none"> • Aprender y aplicar correctamente normas y estándares internacionales. [Usar] 	<ul style="list-style-type: none"> • ISO 9001:2001. • ISO 9000-3. • ISO/IEC 9126. • ISO/IEC 12207. • ISO/IEC 15939. • ISO/IEC 14598. • ISO/IEC 15504-SPICE. • IT Mark. • SCRUM. • SQuaRE. • CISQ.
Lecturas : [Som10], [PM14]	





Universidad de Ingeniería y Tecnología
Escuela Profesional de
Ciencia de la Computación
Silabo del curso
Periodo Académico 2019-I

1. **Código del curso y nombre:** BI0021. Bioinformática y Bioestadística (Obligatorio)
2. **Créditos:** 4
3. **Horas de Teoría y Laboratorio:** 2 HT; 4 HL; (Semanal)
4. **Profesor(es) del curso, email y horario de atención**

Atención previa coordinación con el profesor

5. **Bibliografía básica**

- [Alu06] Srinivas Aluru, ed. *Handbook of Computational Molecular Biology*. Computer and Information Science Series. Boca Raton, FL: Chapman & Hall, CRC, 2006.
- [CB00] P. Clote and R. Backofen. *Computational Molecular Biology: An Introduction*. 279 pages. John Wiley & Sons Ltd., 2000.
- [Dur+98] R. Durbin et al. *Biological Sequence Analysis: Probabilistic Models of Proteins and Nucleic Acids*. Cambridge University Press, 1998, p. 357. ISBN: 9780521629713.
- [Kro+94] Anders Krogh et al. "Hidden Markov Models in Computational Biology, Applications to Protein Modeling". In: *J Molecular Biology* 235 (1994), pp. 1501–1531.
- [Pev00] Pavel A. Pevzner. *Computational Molecular Biology: an Algorithmic Approach*. Cambridge, Massachusetts: The MIT Press, 2000.
- [SM97] João Carlos Setubal and João Meidanis. *Introduction to computational molecular biology*. Boston: PWS Publishing Company, 1997, pp. I–XIII, 1–296. ISBN: 978-0-534-95262-4.

6. **Información del curso**

- (a) **Breve descripción del curso** El uso de métodos computacionales en las ciencias biológicas se ha convertido en una de las herramientas claves para el campo de la biología molecular, siendo parte fundamental en las investigaciones de esta área.
En Biología Molecular, existen diversas aplicaciones que involucran tanto al ADN, al análisis de proteínas o al secuenciamiento del genoma humano, que dependen de métodos computacionales. Muchos de estos problemas son realmente complejos y tratan con grandes conjuntos de datos.
Este curso puede ser aprovechado para ver casos de uso concretos de varias áreas de conocimiento de Ciencia de la Computación como: Lenguajes de Programación (PL), Algoritmos y Complejidad (AL), Probabilidades y Estadística, Manejo de Información (IM), Sistemas Inteligentes (IS).
- (b) **Prerrequisitos:** CS2102. Análisis y Diseño de Algoritmos. (5^{to} Sem)
- (c) **Tipo de Curso:** Obligatorio
- (d) **Modalidad:** Presencial

7. **Objetivos del curso.**

Competencias

- a) Aplicar conocimientos de computación y de matemáticas apropiadas para la disciplina. (**Usar**)
- b) Analizar problemas e identificar y definir los requerimientos computacionales apropiados para su solución. (**Evaluar**)
- 1) Desarrollar principios investigación en el área de computación con niveles de competitividad internacional. (**Evaluar**)

