



## Sociedad Peruana de Computación (SPC)

Programa Profesional de  
Ciencia de la Computación  
Sílabo 2022-I

### 1. CURSO

CS391. Ingeniería de Software III (Obligatorio)

### 2. INFORMACIÓN GENERAL

2.1 Créditos	:	3
2.2 Horas de teoría	:	2 (Semanal)
2.3 Horas de práctica	:	-
2.4 Duración del periodo	:	16 semanas
2.5 Condición	:	Obligatorio
2.6 Modalidad	:	Presencial
2.7 Prerrequisitos	:	CS292. Ingeniería de Software II. (6 <sup>to</sup> Sem)

### 3. PROFESORES

Atención previa coordinación con el profesor

### 4. INTRODUCCIÓN AL 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 elementos 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.

### 5. OBJETIVOS

- 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.

### 6. COMPETENCIAS

- c) Diseñar, implementar y evaluar un sistema, proceso, componente o programa computacional para alcanzar las necesidades deseadas. (**Usar**)
- h) Incorporarse a un proceso de aprendizaje profesional continuo. (**Usar**)
- k) Aplicar los principios de desarrollo y diseño en la construcción de sistemas de software de complejidad variable. (**Usar**)
- l) Desarrollar principios de investigación en el área de computación con niveles de competitividad internacional. (**Usar**)

### 7. COMPETENCIAS ESPECÍFICAS

- c3) Utilizar distintas herramientas y lenguajes de programación en los componentes de software (*Full stack*).

- c4) Diseñar e implementar arquitecturas de software escalables debilmente acopladas, flexibles, seguras y de facil mantenimiento en distintas plataformas.
- c11) Diseñar e implementar un software integrado.
- h1) Desarrollar proyectos de investigación con niveles de complejidad apropiados para pregrado.
- h2) Demostrar que tiene capacidad de aprender a aprender de forma autónoma.
- k3) Aplicar metodologías de desarrollo de software.
- k5) Utilizar técnicas de algoritmos y estructuras de datos para construir software escalable.
- k6) Utilizar los principios de arquitectura de software para construir productos de software confiables.
- k7) Medir atributos de calidad de componentes de software.
- k9) Planear y gestionar proyectos de desarrollo de software.
- 13) Resolver problemas de nuestro entorno en base a nuevas propuestas de soluciones basadas en desarrollo de software.

## 8. TEMAS

Unidad 1: Evolución de Software (12)	
Competencias esperadas: c,d,i,j,m,o	
Temas	Objetivos de Aprendizaje
<ul style="list-style-type: none"> <li>• Desarrollo de Software en el contexto de código grande pre existente               <ul style="list-style-type: none"> <li>– Cambios de software</li> <li>– Preocupaciones y ubicación de preocupaciones</li> <li>– <i>Refactoring</i></li> </ul> </li> <li>• Evolución de Software.</li> <li>• Características de Software mantenible.</li> <li>• Sistemas de Reingeniería.</li> <li>• Reuso de Software.               <ul style="list-style-type: none"> <li>– Segmentos de código</li> <li>– Bibliotecas y <i>frameworks</i></li> <li>– Componentes</li> <li>– Líneas de Producto</li> </ul> </li> </ul>	<ul style="list-style-type: none"> <li>• Identificar los problemas principales asociados con la evolución del software y explicar su impacto en el ciclo de vida del software [Familiarizarse]</li> <li>• Estimar el impacto del cambio de requerimientos en productos existentes de tamaño medio [Usar]</li> <li>• Usar refactorización en el proceso de modificación de un componente de software [Usar]</li> <li>• Estudiar los desafíos de mejorar sistemas en un entorno cambiante [Familiarizarse]</li> <li>• Perfilar los procesos de pruebas de regresión y su rol en el manejo de versiones [Familiarizarse]</li> <li>• Estudiar las ventajas y desventajas de diferentes tipos de niveles de confiabilidad [Familiarizarse]</li> </ul>
Lecturas : [PM15], [Som17]	

Unidad 2: Gestión de Proyectos de Software (10)	
Competencias esperadas: c,d,i,j,m,o	
Temas	Objetivos de Aprendizaje
<ul style="list-style-type: none"> <li>• La participación del equipo: <ul style="list-style-type: none"> <li>– Procesos elemento del equipo, incluyendo responsabilidades de tarea, la estructura de reuniones y horario de trabajo</li> <li>– Roles y responsabilidades en un equipo de software</li> <li>– Equipo de resolución de conflictos</li> <li>– Los riesgos asociados con los equipos virtuales (comunicación, la percepción, la estructura)</li> </ul> </li> <li>• Estimación de esfuerzo (a nivel personal)</li> <li>• Riesgo. <ul style="list-style-type: none"> <li>– El papel del riesgo en el ciclo de vida</li> <li>– Categorías elemento de riesgo, incluyendo la seguridad, la seguridad, mercado, finanzas, tecnología, las personas, la calidad, la estructura y el proceso de</li> </ul> </li> <li>• Gestión de equipos: <ul style="list-style-type: none"> <li>– Organización de equipo y la toma de decisiones</li> <li>– Roles de identificación y asignación</li> <li>– Individual y el desempeño del equipo de evaluación</li> </ul> </li> <li>• Gestión de proyectos: <ul style="list-style-type: none"> <li>– Programación y seguimiento de elementos</li> <li>– Herramientas de gestión de proyectos</li> <li>– Análisis de Costo/Beneficio</li> </ul> </li> </ul>	<ul style="list-style-type: none"> <li>• Discutir los comportamientos comunes que contribuyen al buen funcionamiento de un equipo [Familiarizarse]</li> <li>• Crear y seguir un programa para una reunión del equipo [Usar]</li> <li>• Identificar y justificar las funciones necesarias en un equipo de desarrollo de software [Usar]</li> <li>• Entender las fuentes, obstáculos y beneficios potenciales de un conflicto de equipo [Usar]</li> <li>• Aplicar una estrategia de resolución de conflictos en un ambiente de equipo [Usar]</li> <li>• Utilizar un método ad hoc para estimar el esfuerzo de desarrollo del software (ejemplo, tiempo) y comparar con el esfuerzo actual requerido [Usar]</li> <li>• Listar varios ejemplos de los riesgos del software [Familiarizarse]</li> <li>• Describir el impacto del riesgo en el ciclo de vida de desarrollo de software [Familiarizarse]</li> <li>• Describir las diferentes categorías de riesgo en los sistemas de software [Familiarizarse]</li> <li>• Demostrar a través de la colaboración de proyectos de equipo los elementos centrales de la construcción de equipos y gestión de equipos [Usar]</li> <li>• Describir como la elección de modelos de procesos afectan la estructura organizacional de equipos y procesos de toma de decisiones [Familiarizarse]</li> <li>• Crear un equipo mediante la identificación de los roles apropiados y la asignación de funciones a los miembros del equipo [Usar]</li> <li>• Evaluar y retroalimentar a los equipos e individuos sobre su desempeño en un ambiente de equipo [Usar]</li> <li>• 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]</li> </ul>
<b>Lecturas :</b> [PM15], [Som17]	

Unidad 3: Gestión de Proyectos de Software (8)	
Competencias esperadas: c,d,i,j,m,o	
Temas	Objetivos de Aprendizaje
<ul style="list-style-type: none"> <li>• Software de medición y técnicas de estimación.</li> <li>• Aseguramiento de la calidad del software y el rol de las mediciones.</li> <li>• Riesgo. <ul style="list-style-type: none"> <li>– Identificación de riesgos y gestión.</li> <li>– Análisis riesgo y evaluación.</li> <li>– La tolerancia al riesgo (por ejemplo, riesgo adverso, riesgo neutral, la búsqueda de riesgo)</li> <li>– Planificación de Riesgo</li> </ul> </li> <li>• En todo el sistema de aproximación al riesgo, incluyendo riesgos asociados con herramientas.</li> </ul>	<ul style="list-style-type: none"> <li>• Realizar el seguimiento del progreso de alguna etapa de un proyecto que utiliza métricas de proyectos apropiados [Usar]</li> <li>• Comparar las técnicas simples de tamaño de software y estimación de costos [Usar]</li> <li>• Usar una herramienta de gestión de proyectos para ayudar en la asignación y rastreo de tareas en un proyecto de desarrollo de software [Usar]</li> <li>• Describir el impacto de la tolerancia de riesgos en el proceso de desarrollo de software [Evaluar]</li> <li>• Identificar riesgos y describir enfoques para manejar riesgos (evitar, aceptar, transferir, mitigar) y caracterizar fortalezas y defectos para cada uno [Familiarizarse]</li> <li>• Explicar cómo el riesgo afecta las decisiones en el proceso de desarrollo de software [Usar]</li> <li>• Identificar los riesgos de seguridad para un sistema de software [Usar]</li> <li>• Demostrar un enfoque sistemático para la tarea de identificar los peligros y riesgos en una situación particular [Usar]</li> <li>• Aplicar los principios básicos del manejo de riesgos en una variedad de escenarios simples incluyendo una situación de seguridad [Usar]</li> <li>• Dirigir un análisis de costo/beneficio para el enfoque de mitigación de riesgos [Usar]</li> <li>• Identificar y analizar alguno de los riesgos para un sistema entero que surgen de aspectos distintos del software [Usar]</li> </ul>
Lecturas : [PM15], [Som17]	

Unidad 4: Procesos de Software (12)	
Competencias esperadas: c,d,i,j,m,o	
Temas	Objetivos de Aprendizaje
<ul style="list-style-type: none"> <li>• Consideraciones a nivel de sistemas, ejem., la interacción del software con su entorno.</li> <li>• Introducción a modelos del proceso de software (e.g., cascada, incremental, ágil): <ul style="list-style-type: none"> <li>– Actividades con ciclos de vida de software.</li> </ul> </li> <li>• Programación a gran escala versus programación individual.</li> <li>• Evaluación de modelos de proceso de software.</li> <li>• Conceptos de calidad de software.</li> <li>• Mejoramiento de procesos.</li> <li>• Modelos de madurez de procesos de software.</li> <li>• Mediciones del proceso de software.</li> </ul>	<ul style="list-style-type: none"> <li>• 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]</li> <li>• Describir las ventajas y desventajas relativas entre varios modelos importantes de procesos (por ejemplo, la cascada, iterativo y ágil) [Usar]</li> <li>• Describir las diferentes prácticas que son componentes clave de los diversos modelos de procesos [Usar]</li> <li>• Diferenciar entre las fases de desarrollo de software [Usar]</li> <li>• 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]</li> <li>• Explicar el concepto de ciclo de vida del software y proporcionar un ejemplo que ilustra sus fases incluyendo los entregables que se producen [Usar]</li> <li>• 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]</li> <li>• Definir la calidad del software y describir el papel de las actividades de aseguramiento de la calidad en el proceso de software [Usar]</li> <li>• Describir el objetivo y similitudes fundamentales entre los enfoques de mejora de procesos [Usar]</li> <li>• Comparar varios modelos de mejora de procesos, tales como CMM, CMMI, CQI, <i>Plan-Do-Check-Act</i>, o ISO9000 [Usar]</li> <li>• 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]</li> <li>• Explicar el papel de los modelos de madurez de procesos en la mejora de procesos [Usar]</li> <li>• Describir varias métricas de procesos para la evaluación y el control de un proyecto [Usar]</li> <li>• Usar las medidas en proyecto para describir el estado actual de un proyecto [Usar]</li> </ul>
<b>Lecturas :</b> [PM15], [Som17]	

Unidad 5: Estándares ISO/IEC (6)	
Competencias esperadas: c,d,i,j,m,o	
Temas	Objetivos de Aprendizaje
<ul style="list-style-type: none"> <li>• ISO 9001:2001.</li> <li>• ISO 9000-3.</li> <li>• ISO/IEC 9126.</li> <li>• ISO/IEC 12207.</li> <li>• ISO/IEC 15939.</li> <li>• ISO/IEC 14598.</li> <li>• ISO/IEC 15504-SPICE.</li> <li>• IT Mark.</li> <li>• SCRUM.</li> <li>• SQuaRE.</li> <li>• CISQ.</li> </ul>	<ul style="list-style-type: none"> <li>• Aprender y aplicar correctamente normas y estándares internacionales. [Usar]</li> </ul>
Lecturas : [Som17], [PM15]	

## 9. PLAN DE TRABAJO

### 9.1 Metodología

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.

### 9.2 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.

### 9.3 Sesiones Prácticas

Las sesiones prácticas se llevan en clase donde se desarrollan una serie de ejercicios y/o conceptos prácticos mediante planteamiento de problemas, la resolución de problemas, ejercicios puntuales y/o en contextos aplicativos.

## 10. SISTEMA DE EVALUACIÓN

\*\*\*\*\* EVALUATION MISSING \*\*\*\*\*

## 11. BIBLIOGRAFÍA BÁSICA

[PM15] Roger S. Pressman and Bruce Maxim. *Software Engineering: A Practitioner's Approach*. 8th. McGraw-Hill, Jan. 2015.

[Som17] Ian Sommerville. *Software Engineering*. 10th. Pearson, Mar. 2017.