

Nombre	Fundamentos de Ingeniería de Software para Producción Industrial
Créditos	8
Objetivos de la Asignatura	<p>Brindar un panorama de los aspectos técnicos y administrativos más relevantes de la Ingeniería de Software para estudiantes de ingeniería que no son del área informática.</p> <p>Brindar conocimientos de algunas técnicas y herramientas utilizadas en la producción industrial de software.</p> <p>Brindar conocimientos del proceso de desarrollo de software y de los subprocesos de las disciplinas centrales del desarrollo de software.</p> <p>Este curso está pensado para carreras que no son de informática pero que requieren que el estudiante tenga conocimiento sobre la producción industrial de software. Por ende, se trabajará a nivel de conocimiento en las áreas más técnicas del software, y a nivel de conocimiento y adquisición de habilidades en las áreas que tratan aspectos de gestión y procesos de la producción de software (como gestión de proyectos de software, gestión de la calidad de software, gestión de la configuración, proceso de desarrollo de software, proceso de ingeniería de requisitos, entre otras).</p> <p>Entonces, el objetivo central de esta asignatura es que el estudiante reconozca las diferencias entre la producción de software y otros tipos de producción en ingeniería y que sea capaz de entender las tareas de gestión y procesos de la producción de software. En conjunto con otras asignaturas que el estudiante pueda tomar y con la adquisición de experiencia profesional, podrá participar o co-participar en actividades de gestión de software en proyectos de desarrollo de software.</p>
Metodología de enseñanza	<p>2 clases por semana de 2 horas cada una durante 15 semanas.</p> <p>Cada clase tendrá contenido tanto teórico como práctico.</p> <p>Los estudiantes deberán ir a las clases con los temas ya leídos del libro para poder participar en la discusión de los distintos tópicos. También se presentarán artículos publicados en reconocidas revistas y/o conferencias que complementen el libro del curso.</p> <p>La dedicación del estudiante se distribuye de la siguiente manera:</p> <ul style="list-style-type: none"> - 60 horas de clases (30 clases de 2 horas distribuidas en 15 semanas) - 60 horas de estudio individual para el seguimiento de las clases, realización de los prácticos y preparación de parciales.
Temario	<p>1. Introducción</p> <p>1.1. Definición de la ingeniería de software como disciplina. Referencias: [Sommerville 10 – cap. 1] [Parnas 11]</p> <p>1.2. Ética en ingeniería de software Referencias: [SECEPP] [Sommerville 10 – cap. 1]</p> <p>1.3. La ingeniería de software como una profesión. Referencias: [McConnell 13] [Sommerville 10 – cap. 1]</p> <p>1.4. Habilidades del ingeniero de software Referencias: [Rivera-Ibarra 10]</p>

2. Procesos de desarrollo de software

Referencias: [Sommerville 10 – cap. 2]

3. Ingeniería de Requisitos

- 3.1. Requisitos funcionales y no funcionales
- 3.2. El documento de requisitos de software
- 3.3. Especificación de requisitos
- 3.4. Proceso de ingeniería de requisitos
- 3.5. Modelado con Casos de Uso

Referencias: [Sommerville 10 – cap. 4]

4. Arquitectura y diseño de software

- 4.1. Qué es la arquitectura de software y su importancia
- 4.2. Vistas de la arquitectura de software: de casos de uso, lógica, de procesos, de desarrollo y física (se explicará y mostrará un ejemplo de cada vista)
- 4.3. Arquitectura en capas y arquitectura cliente-servidor (se explicarán estos dos tipos de arquitectura en particular)
- 4.4. Diagrama de clases y de interacción para diseño de bajo nivel

Nota: Este punto se tratará a nivel de conocimientos ya que las habilidades necesarias para desarrollar arquitecturas de software o diseños de bajo nivel no son necesarios para este curso. El estudiante debe comprender la importancia de la arquitectura y el diseño de software así como los modelos más conocidos utilizados en el desarrollo. Para este tema no se utilizará el libro del curso.

5. Pruebas (*testing*) de software

- 5.1. Definiciones básicas de pruebas de software
- 5.2. Pruebas durante el desarrollo de software: unitarias, de integración y de sistema
- 5.3. Pruebas de liberación (*release*)

Referencias: [Sommerville 10 – cap. 8]

6. Evolución del software

- 6.1. Proceso de evolución
- 6.2. Mantenimiento de software

Referencias: Referencias: [Sommerville 10 – cap. 9]

7. Gestión de proyectos de software

- 7.1. Nociones básicas y diferencias con la gestión proyectos de otras ramas de la ingeniería
- 7.1. Gestión del riesgo
- 7.2. Gestión de recursos humanos

7.3. Trabajo en equipos

Referencias: Referencias: [Sommerville 10 – cap. 22]

8. Planificación de proyectos de software

- 8.1. Precio del software
- 8.2. Desarrollo guiado por el plan de proyecto
- 8.3. Calendario del proyecto
- 8.4. Técnicas de estimación de software

Referencias: Referencias: [Sommerville 10 – cap. 23 (sin sección 23.4)]

9. Gestión de la calidad de software

- 9.1. Calidad de software
- 9.2. Estándares de software
- 9.3. Revisiones e inspección
- 9.4. Medición y medidas de software

Referencias: Referencias: [Sommerville 10 – cap. 24]

10. Gestión de la configuración del software

- 10.1. Gestión del cambio
- 10.2. Control y gestión del versionado

10.3. Construcción del sistema

10.4. Gestión de la liberación (*release*)

Referencias: Referencias: [Sommerville 10 – cap. 25]

Bibliografía

[McConell 13] Steven McConnell and Leonard L. Tripp; Software Engineering Professional Practices, en Software Engineering Essentials, Volume II: The Supporting Processes, Richard Hall Thayer and Merlin Dorfman, Eds., 2013, cap. 11

[Parnas 11] David Parnas; Software Engineering - Missing in Action: A Personal Perspective, IEEE Computer, vol. 44, no. 10, pp. 54-58, 2011.

[Rivera-Ibarra 10] Rivera-Ibarra, J.G.; Rodriguez-Jacobo, J.; Serrano-Vargas, M.A., Competency Framework for Software Engineers, Software Engineering Education and Training (CSEE&T), 2010 23rd IEEE Conference on , vol., no., pp.33,40, 9-12 March 2010

[SECEPP] Joint Task Force on Software Engineering Ethics and Professional Practices, IEEE Computer Society and Association for Computing Machinery; Software Engineering Code of Ethics and Professional Practice;1999

[Sommerville 10] Ian Sommerville; Software Engineering, 9th Edición, Addison-Wesley, 2010, ISBN 978-0137035151

Conocimientos previos recomendados

- Conocimientos de programación
- Conocimientos básicos de producción

Anexo

Carrera: Ingeniería de Producción

1) Cronograma tentativo

El curso tiene una duración de 15 semanas y se dicta durante el primer semestre.

Semana	Tema
1	1. Introducción
2	2. Procesos de desarrollo de software
3	2. Procesos de desarrollo de software
4	3. Ingeniería de Requisitos / Prueba parcial
5	3. Ingeniería de Requisitos / Obligatorio I
6	4. Arquitectura y diseño de software
7	5. Pruebas (<i>testing</i>) de software
8	5. Pruebas (<i>testing</i>) de software / Prueba parcial
9	6. Evolución del software
10	7. Gestión de proyectos de software
11	7. Gestión de proyectos de software
12	8. Planificación de proyectos de software / Prueba parcial
13	8. Planificación de proyectos de software / Obligatorio II
14	9. Gestión de la calidad de software
15	10. Gestión de la configuración del software / Prueba parcial

2) Modalidad del curso y procedimiento de evaluación

Modalidad de curso: Clases teórico-prácticas obligatorias. Lecturas de libro y artículos recomendados. En el curso se utilizará también material multimedia como videos de exposiciones teóricas.

Procedimiento de evaluación:

- Asistencia a clase obligatoria. Los estudiantes deben asistir al 80% de las clases para poder exonerar el curso. Es importante la asistencia a clase para fomentar la discusión de los temas planteados.
- Se tomarán pruebas parciales y continuas durante el curso. Para exonerar el estudiante debe obtener al menos un 40% en cada prueba individual y un 60% del puntaje obtenido en el total de las pruebas.
- Se realizarán dos trabajos obligatorios grupales que deberán ser presentados y defendidos en clase. El resultado de cada obligatorio es aprobado o reprobado. Para exonerar el curso ambos obligatorios deben ser aprobados.
- El resultado que se puede obtener es exoneración del curso o reprobación del mismo. En este último caso el estudiante tendrá la posibilidad de volver a cursarlo en una próxima edición del mismo.

3) **Materia**

Ingeniería de computación

4) **Previaturas** (de curso)

Carrera: 22-2

- El estudiante debe haber aprobado al menos 270 créditos en la carrera.

5) **Cupo**

Sin cupo.

6) **Esta asignatura no adhiere a resolución del consejo sobre condición de libre**

APROB. RES. CONSEJO DE FAC. ING.

DECISION 18.6.15 Exp. 060120-000365-15

3
fre