



UNIVERSIDAD  
DE LA REPÚBLICA  
URUGUAY

## Programa de PROFESIÓN EN INGENIERÍA DE SOFTWARE

### 1. NOMBRE DE LA UNIDAD CURRICULAR

Profesión en Ingeniería de Software

### 2. CRÉDITOS

6 créditos

### 3. OBJETIVOS DE LA UNIDAD CURRICULAR

Brindar un panorama general sobre la profesión en ingeniería de software discutiendo diversos aspectos de la misma.

Se busca que a través de la realización de este curso el estudiante pueda:

- Conocer y utilizar el cuerpo de conocimiento de la ingeniería de software (SWEBOK) así como saber de sus limitaciones.
- Conocer sobre los aspectos que definen una profesión y en particular la profesión en la ingeniería de software.
- Conocer y diferenciar distintos elementos que conforman a una profesión en ingeniería, tales como: educación profesional inicial, acreditaciones, desarrollo de habilidades, certificaciones, licenciamiento, desarrollo profesional, sociedades profesionales y código de ética. Como ejemplo se discutirán artículos que presentan cómo está hoy en día la Ingeniería de Software con respecto a cada uno de estos elementos.

### 4. METODOLOGÍA DE ENSEÑANZA

Clases presenciales obligatorias una vez por semana de 2 horas. Son un total de 8 clases presenciales.

El curso se divide en dos partes. La primera es la parte teórica y la segunda es un trabajo en grupo.

### **Primera parte**

La primera parte del curso se basa en libros y artículos que los estudiantes deberán leer y comprender durante el curso. Los libros y artículos están escritos en su gran mayoría en inglés. Los estudiantes deberán preparar y presentar cada una de las clases a menos de algún tema puntual. Estas clases estarán basadas en el material de lectura. La preparación y presentación de las clases se realizará por parte de grupos de estudiantes conformados para tal fin.

Todas las clases, a excepción de la primera (que será una clase teórica introductoria), se dividirán en 4 partes:

1. Parciales de control de lectura de artículos al comienzo de cada clase.
2. Presentaciones de artículos por grupos de estudiantes.
3. Preguntas sobre los artículos preparadas por grupos de estudiantes.
4. Discusión de los temas presentados en los artículos.

Las clases comenzarán con una prueba parcial individual que buscará evaluar los conocimientos adquiridos por cada estudiante.

Para las clases se asignarán artículos para leer a todos los estudiantes y estos deberán ser presentados por parte de uno o más grupos de estudiantes. Se debe evitar hacer una presentación resumida del artículo ya que todos los estudiantes lo han leído antes de clase. Entonces, la presentación se deberá enfocar en presentar algún tipo de discusión o análisis sobre el/los artículos que les fueron asignados al grupo.

Además, los estudiantes tendrán que buscar un nuevo artículo, vinculado al tema en cuestión, para presentar al resto de los grupos. Este artículo, a diferencia del asignado por los docentes, al no ser leído por el resto de los estudiantes, debe ser presentado enfocándose en explicar lo que los autores proponen. Obviamente también se puede realizar un análisis crítico.

A su vez, uno o más grupos deberán preparar preguntas sobre los artículos que fueron enviados para leer para discutir en clase.

Las clases también tendrán un espacio de media hora de discusión guiado por los docentes de los temas presentados. Los estudiantes deberán participar en clase discutiendo cada uno de los temas.

### **Segunda parte**

Los estudiantes desarrollarán, con los mismos grupos definidos en la primera parte, un trabajo escrito vinculado a los temas vistos en el curso. El trabajo deberá presentarse en clase. El cuerpo docente propondrá varios temas de los cuales cada grupo elegirá uno.

La dedicación del estudiante se distribuye de la siguiente manera:

- 16 horas de clase presenciales (8 clases de 2 horas.)
- 31 horas de lectura de artículos y estudio individual.
- 17 horas de búsqueda de artículos, preparación de presentación de artículos y discusión de artículos.
- 26 horas en preparación del trabajo grupal.

## 5. TEMARIO

1. Introducción
  - 1.1. Definición de la ingeniería de software como disciplina.
  - 1.2. Diferencias entre la ciencia de la computación y la ingeniería de software.
  - 1.3. Qué son las organizaciones internacionales ACM e IEEE-CS.
  - 1.4. Introducción básica a los temas del curso.
2. Cuerpo de conocimiento en ingeniería de software (SWEBOK).
  - 2.1. El cuerpo de conocimiento y su importancia.
  - 2.2. Áreas que conforman el cuerpo de conocimiento.
  - 2.3. Relación entre el cuerpo de conocimiento, la Educación y la Profesión en ingeniería de software.
  - 2.4. Historia del cuerpo de conocimiento.
3. Profesión en ingeniería de software.
  - 3.1. ¿Qué define a una profesión?
  - 3.2. La ingeniería de software como una profesión.
  - 3.3. Ética en ingeniería de software.
  - 3.4. Licenciamiento en ingeniería de software y otras "credenciales"
  - 3.5. Habilidades del ingeniero de software

## 6. BIBLIOGRAFÍA

A continuación se presenta la bibliografía a utilizarse para el curso.

Todos los libros y/o artículos son accesibles a través de Timbó o se pueden descargar libremente desde la web. A su vez, dichos recursos están disponibles en el EVA para que los estudiantes lo descarguen.

Tema	Básica	Complementaria
Introducción	(1-4)	(17)
Cuerpo de conocimiento en ingeniería de software (SWEBOK).	(5-6)	
Profesión en ingeniería de software.	(7-16)	(18)

### 6.1 Básica

1. Parnas, David (2011). Software Engineering - Missing in Action: A Personal Perspective, IEEE Computer, vol. 44, no. 10, pp. 54-58.
2. Joint Task Force on Computing Curricula IEEE Computer Society Association for Computing Machinery (2015). Software Engineering 2014 - Curriculum Guidelines for Undergraduate Degree Programs in Software Engineering, pp. 10-17.
3. The Association for Computing Machinery(ACM);The Association for Information Systems (AIS);The Computer Society (IEEE-CS), (2005). Computing Curricula 2005 - The Overview Report covering undergraduate degree programs in Computer Engineering, Computer Science, Information Systems, Information Technology, Software Engineering, cap. 2 pp 13-15.
4. Brooks, Frederick (1987). No Silver Bullet Essence and Accidents of Software Engineering. IEEE Computer Society Press, vol. 20, no. 4, pp. 1-5.
5. P. Bourque and R. E. Fairley (2014). Guide to the Software Engineering Body of Knowledge (SWEBOK), IEEE Computer Society Press.
6. Camilloni, Lucia; Vallespir, Diego (2014). Sobre profesión y educación en ingeniería de software, Reporte Técnico InCo/PEDECIBA (versión borrador).
7. McConnell, Steve;Tripp Leonard (2013); Software Engineering Professional Practices, en Software Engineering Essentials, Volume II: The Supporting Processes, Richard Hall Thayer and Merlin Dorfman, Eds., 2013, cap. 11.
8. Joint Task Force on Software Engineering Ethics and Professional Practices, IEEE Computer Society and Association for Computing Machinery (1999).Software Engineering Code of Ethics and Professional Practice.
9. Rogerson, Simon (2002). The Software Engineering Code of Ethics and Professional Practice:a case for being proactive, in 26 th Annual International Computer Software and Applications Conference.
10. Bagert, Donald J. (1999). Licensing Software Engineers: The Computer Society Must Continue to Lead, in COMPSAC.
11. Laplante, Phillip A. (2013). An International Perspective on U.S. Licensure of Software Engineers, Technology and Society, vol. 32, no. 1, pp. 28-30.
12. Kruchten, Philippe (2008). Licensing Software Engineers?; IEEE Software 25(6): pp. 35-37.

13. Ellis, Heidi J.C. , Steven A (2009). Demurjian and J. Fernando Naveda; Software Engineering: Effective Teaching and Learning Approaches and Practices, IGI Global, cap. 1-17-18.
14. Rivera-Ibarra, J.G.; Rodriguez-Jacobo, J.; Serrano-Vargas, M.A. (2010), 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.
15. IEEE CS (2014). The software engineering competency model (SWECOM), draft version.
16. Sedelmaier, Y.; Landes, D. (2012). A research agenda for identifying and developing required competencies in software engineering, Interactive Collaborative Learning (ICL), 2012 15th International Conference on , vol., no., pp.1,5, 26-28 Sept. 2012.

## 6.2 Complementaria

17. Shaw, Mary (2009). Continuing prospects for an engineering discipline of software, IEEE Software, vol. 26, no. 6, pp. 64-67.
18. McConnell, Steve;Tripp Leonard (1999). Professional Software Engineering-Factor or Fiction?, IEEE Software, vol. 16, no. 6, pp. 13-18.

## 7. CONOCIMIENTOS PREVIOS EXIGIDOS Y RECOMENDADOS

**7.1 Conocimientos Previos Exigidos:** Conocimientos de ingeniería de software. Avance en la carrera suficiente como para entender temas de la profesión de ingeniero de software.

**7.2 Conocimientos Previos Recomendados:**

## ANEXO A

### Para todas las Carreras

Esta primera parte del anexo incluye aspectos complementarios que son generales de la unidad curricular.

#### A1) INSTITUTO

Instituto de Computación (InCo)

#### A2) CRONOGRAMA TENTATIVO

A continuación se presenta el cronograma de las horas dedicadas en clase para cada uno de los temas. Las restantes horas (todas no presenciales) se distribuyen de forma uniforme a lo largo de la duración del curso.

Semana 1	Tema 1. Introducción (2 horas de clase)
Semana 2	Tema 1. Introducción (1 hora de clase). Tema 2. Cuerpo de conocimiento en ingeniería de software (SWEBOK), (1 hora de clase).
Semana 3	Tema 2. Cuerpo de conocimiento en ingeniería de software (SWEBOK), (1 hora de clase). Tema 3. Profesión en ingeniería de software. (1 hora de clase)
Semana 4	Tema 3. Profesión en ingeniería de software (2 horas de clase).
Semana 5	Tema 3. Profesión en ingeniería de software (2 horas de clase).
Semana 6	Tema 3. Profesión en ingeniería de software (2 horas de clase).
Semana 7	Presentación de trabajo por parte de estudiantes (2 horas de clase).
Semana 8	Presentación de trabajo por parte de estudiantes (2 horas de clase).

#### A3) MODALIDAD DEL CURSO Y PROCEDIMIENTO DE EVALUACIÓN

La evaluación consta de cuatro partes:

1. Pruebas parciales en cada clase. (60% de los puntos)
2. Presentaciones de los grupos de estudiantes de temas del curso. Incluye preguntas elaboradas por los estudiantes. (20% de los puntos)
3. Trabajo final y presentación del mismo (20% de los puntos)
4. Asistencia a clase obligatoria. El estudiante debe asistir al 85% de las clases para poder exonerar el curso (dado que son ocho clases presenciales se puede entonces tener una única falta). Es importante la asistencia a clase para fomentar la discusión de los temas planteados.

El curso se aprueba con el 60% de los puntos.

El resultado que se puede obtener es la exoneración del curso o la reprobación del mismo.

**A4) CALIDAD DE LIBRE**

La unidad curricular no permite el acceso de estudiantes en la Calidad de Libre.

**A5) CUPOS DE LA UNIDAD CURRICULAR**

Cupos mínimos: 10

Cupos máximos: 36

**ANEXO B para las carreras Ingeniería en Computación (plan 97) y Licenciatura en Computación**

**B1) ÁREA DE FORMACIÓN**

Ingeniería de Software

**B2) UNIDADES CURRICULARES PREVIAS**

Curso: Para realizar este curso se debe contar con:

- el examen aprobado de Introducción a la ingeniería de software
- al menos 250 créditos aprobados

Examen: No aplica.

**ANEXO C para la carrera Ingeniería en computación (código de carrera 70)**

**C1) ÁREA DE FORMACIÓN**

Este plan no se divide en áreas.

**C2) UNIDADES CURRICULARES PREVIAS**

Curso: Para realizar este curso se debe contar con:

- el examen aprobado de Bases de Datos
- el examen aprobado de Taller III
- el examen aprobado de Taller IV
- el examen aprobado de Programación III

Examen: No aplica.

**ANEXO B para la(s) carrera(s) Ingeniería en Producción**

**B1) ÁREA DE FORMACIÓN**

Ingeniería en Computación

**B2) UNIDADES CURRICULARES PREVIAS**

Para el Curso: tener al menos 270 créditos aprobados

Para el Examen: no aplica.

11  
APROB. RES. CONSEJO DE FAC. ING.  
Fecha 15/8/18 Exp. 060120-02146-17