

# Programa de Unidad Curricular

**Nombre de la unidad curricular** Ingeniería de Software Basada en Evidencias y Revisiones Sistemáticas

**Créditos** 7 Créditos

**Objetivo de la unidad curricular** La ingeniería de software basada en evidencias (EBSE por sus siglas en inglés) tiene como propósito mejorar la toma de decisiones relacionada al desarrollo y mantenimiento de software integrando la mejor evidencia actual de la investigación con experiencias prácticas. Para lograr esto, alienta un fuerte énfasis en el rigor metodológico involucrando los siguientes cinco pasos:

1. Convertir un problema relevante o una necesidad de información en una pregunta que pueda ser respondida.
2. Buscar en la literatura la mejor evidencia para responder a esa pregunta.
3. Evaluar de forma crítica la validez, el impacto y la aplicación de la evidencia.
4. Integrar la evidencia evaluada con la experiencia práctica y los valores y circunstancias de los interesados.
5. Evaluar el desempeño y buscar maneras de mejorarlo.

La EBSE surge como una analogía a la práctica médica en donde el paradigma ha cambiado drásticamente la investigación. En esa rama de las ciencias se utilizan revisiones para resumir, por ejemplo, los resultados de una nueva droga presente en distintos medicamentos probados en diferentes lugares del mundo. La rigurosidad metodológica hace que los resultados sean más confiables ya que es posible estudiar el procedimiento llevado a cabo para su obtención así como también reproducirlo. El mecanismo normal para la identificación y la agregación de evidencia de investigación es la revisión sistemática de literatura.

El objetivo de esta asignatura es brindar una visión general y práctica de ingeniería de software basada en evidencias, describiendo sus fundamentos, técnicas, y herramientas para su aplicación práctica.

Al finalizar el curso el estudiante será capaz de:

- Comprender los conceptos básicos de la ingeniería de software basada en evidencias.
- Identificar problemas de la actividad profesional que pueden ser respondidos mediante la búsqueda de evidencia en la literatura.
- Evaluar revisiones sistemáticas de literatura sobre ingeniería de software ya publicadas.
- Participar en la planificación y ejecución de revisiones sistemáticas de literatura.

<b>Metodología de enseñanza</b>	<p>La enseñanza estará realizada fundamentalmente en modalidad de clases presenciales. Se estima un total de aproximadamente de 100 horas de trabajo del estudiante, desglosadas según el siguiente detalle:</p> <ul style="list-style-type: none"> <li>• 36 horas de asistencia a clases.</li> <li>• 24 horas de estudio de material teórico previo a cada clase.</li> <li>• 24 horas de trabajo práctico posterior a cada clase.</li> <li>• 10 horas de lectura y preparación de presentación de un artículo científico.</li> <li>• 10 horas de finalización del trabajo grupal correspondiente al laboratorio.</li> </ul>
<b>Temario</b>	<p>Durante las clases se realizarán diferentes actividades que estimulen la investigación, la lectura de artículos y la discusión y reflexión sobre las problemáticas presentadas. Se buscará también involucrar a los participantes en discusiones activas.</p> <ol style="list-style-type: none"> <li>1. Introducción al paradigma basado en evidencias.</li> <li>2. Ingeniería de software basada en evidencias</li> <li>3. Revisiones sistemáticas en ingeniería de software</li> <li>4. Planificación de una revisión sistemática</li> <li>5. Búsqueda de estudios primarios</li> <li>6. Selección de estudios</li> <li>7. Evaluación de la calidad de los estudios</li> <li>8. Extracción de datos</li> <li>9. Síntesis de datos       <ol style="list-style-type: none"> <li>a. Análisis de estudios de mapeo</li> <li>b. Síntesis cualitativas</li> <li>c. Síntesis cuantitativas</li> </ol> </li> <li>10. Informe de una revisión sistemática</li> <li>11. Traducción del conocimiento y difusión</li> </ol>
<b>Bibliografía y referencias</b>	<p>Kitchenham, B. A., Budgen, D., &amp; Brereton, P. (2015). <u>Evidence-Based Software Engineering and Systematic Reviews</u>. CRC Press. Disponible en biblioteca del Instituto de Computación.</p> <p>Genero Bocco, M., Cruz Lemus, J. A., &amp; Piattini Velthuis, M. G. (2014). <u>Métodos de investigación en ingeniería del software</u>. Paracuellos de Jarama, Madrid: Ra-Ma. Disponible en biblioteca del Instituto de Computación.</p> <p>Pizard, S., Acerenza, F., Casella, V., Moreno, S., &amp; Vallespir, D. (2015). <u>Conceptos de Ingeniería de Software basado en evidencias</u>. Montevideo: Instituto de Computación - Facultad de Ingeniería, UdelaR.  <a href="https://www.fing.edu.uy/inco/pedeciba/bibliote/reptec/TR1508.pdf">https://www.fing.edu.uy/inco/pedeciba/bibliote/reptec/TR1508.pdf</a></p>
<b>Conocimientos previos exigidos y recomendados</b>	<p>Conocimientos básicos de ingeniería de software y modelado de sistemas.</p>

## **Anexo (Ingeniería de Software Basada en Evidencias y Revisiones Sistemáticas)**

### **Cronograma tentativo.**

- ▣ Clases presenciales:
  - Clase 1: Introducción al paradigma basado en evidencias. Ingeniería de software basada en evidencias. Revisiones sistemáticas en ingeniería de software.
  - Clase 2: Planificación de una revisión sistemática.
  - Clase 3: Búsqueda de estudios primarios.
  - Clase 4: Selección de estudios.
  - Clase 5: Evaluación de la calidad de los estudios.
  - Clase 6: Extracción de datos. Asignación de artículos para evaluación.
  - Clase 7: Síntesis de datos. Análisis de estudios de mapeo.
  - Clase 8: Síntesis cualitativas. Síntesis cuantitativas.
  - Clase 9: Informe de una revisión sistemática. Traducción del conocimiento y difusión.
  - Clase 10: Trabajo práctico sobre revisiones sistemáticas.
  - Clase 11: Presentación de evaluación de artículos.
  - Clase 12: Presentación de evaluación de artículos. Repaso y cierre del Trabajo práctico.

### **Modalidad del curso y procedimiento de evaluación.**

El curso tiene una modalidad de dictado presencial y estudio por parte de los estudiantes antes y luego de las clases. Habrá un encuentro semanal de 3 horas que combinará exposiciones teóricas con actividades grupales. Además, se estimulará la investigación, la lectura de artículos provistos, y la discusión y reflexión sobre las problemáticas presentadas. Se buscará involucrar a los participantes en discusiones activas, por ejemplo, a través de la entrega y de la discusión de ejercicios.

El curso se evaluará mediante una prueba individual escrita y un trabajo grupal bajo la forma de laboratorio o taller. La prueba individual corresponderá al 70% de la evaluación.

El trabajo grupal constará de trabajos prácticos en cada clase y, además, la evaluación de un artículo científico. Cada grupo deberá entregar un informe final del trabajo realizado antes de finalizar curso y eventualmente realizar una presentación oral. El trabajo grupal corresponderá al 30% restante de la evaluación.

Para la aprobación final del curso se requiere un mínimo de 60% de los puntos en cada parte y un mínimo de 60% en el total.

### **Materia**

Ingeniería de Software – para las carreras 72 y 73.

### **Previaturas**

Carrera 73: LICENCIATURA EN COMPUTACIÓN (Plan 2012)

Para cursar esta asignatura es necesario tener aprobado:

- examen de Programación 4
- curso de Taller de Programación
- examen de Introducción a la Ingeniería de Software

Carrera 72: INGENIERÍA EN COMPUTACIÓN (Plan 97):

Para cursar esta asignatura es necesario tener aprobado:

- examen de Programación 4

- curso de Taller de Programación
  - examen de Introducción a la Ingeniería de Software
- Carrera 70 y 71: INGENIERÍA EN COMPUTACIÓN (Plan 87):
- Para cursar esta asignatura es necesario tener aprobados los cursos de:
- Taller III
  - Taller IV

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

10200  
4.7.17  
060120-001052-17