

# **Programa de INGENIERÍA DE SOFTWARE**

## **1. MATERIA**

Ingeniería de software

## **2. CRÉDITOS**

12 créditos

## **3. OBJETIVOS DE LA ASIGNATURA**

La ingeniería de software es la disciplina relacionada a la aplicación de teoría, conocimiento y práctica para construir de forma efectiva y eficiente sistemas de software confiables que satisfagan los requisitos de los clientes y usuarios. Aplica a sistemas pequeños, medianos y grandes y acompaña todas las fases del ciclo de vida del software, desde su concepción hasta su operación por parte de los usuarios.

El objetivo de esta unidad curricular es brindar un panorama general de la disciplina a los estudiantes profundizando en los aspectos más relevantes. Busca introducir al estudiante en el uso de algunas técnicas y herramientas particulares para que pueda participar en distintos roles en un proyecto de ingeniería de software.

## **4. METODOLOGÍA DE ENSEÑANZA**

Esta unidad curricular tiene un fuerte componente de taller. La parte de taller será llevada adelante por grupos de estudiantes. En el taller se realizan prácticas y ejercicios que buscan que los estudiantes incorporen técnicas y habilidades básicas de la ingeniería de software.

El docente podrá dictar clases teóricas y/o distribuir material teórico que deberá ser estudiado de forma autónoma por los estudiantes. El material disponible será libro de curso, videos especialmente realizados para este curso, notas de teórico y artículos, entre otros.

Se dictarán clases de presentación del curso, presentación del taller, introducción a la ingeniería de software, y además cada grupo tendrá seguimiento de su trabajo de taller.

Cada semana los estudiantes deberán avanzar en los temas teóricos pautados para esa semana y realizarán trabajos prácticos y de taller en grupos.

Todo el curso, y en particular el taller, buscan que el estudiante incorpore elementos prácticos de las actividades más fundamentales de la disciplina (a modo de ejemplo, llevar a la práctica, con alguna dinámica, la forma de recolectar requisitos, aprender a escribir un caso de uso o historia de usuario, entender un diagrama de componentes de la arquitectura de software, realizar un plan de pruebas de un sistema).

Fuera de clase los estudiantes deben: estudiar el material teórico, realizar ejercicios en grupos que serán discutidos en las clases de taller, rendir las evaluaciones.

Carga horaria:

- Estudio teórico – 6 horas semanales.
- Trabajo en el taller, clases de taller, repaso del teórico, ejercicios prácticos, otros – 6 horas semanales.

## 5. TEMARIO

### 1. **Introducción a la ingeniería de software:**

Descripción del desarrollo profesional de software, descripción de la ingeniería de software, y algunos casos de estudio que permitan introducir los problemas particulares a tratar en el resto del curso.

### 2. **Procesos de desarrollo de software:** Un proceso de software es un conjunto coherente de actividades para la producción de software. En este tema se introducirá al estudiante a uno o más modelos de procesos de desarrollo de software y a aspectos relevantes, incluyendo, desarrollo incremental, y agilidad.

### 3. **Gestión de desarrollo de software:** Introducción a uno o más enfoques de gestión del desarrollo de software, esto incluye, por ejemplo, planificación de actividades, involucramiento de clientes, equipos de desarrollo y flujos de trabajo.

### 4. **Requisitos y Funciones del software:** Introducción consideración de las necesidades y comportamiento de clientes y usuarios como base para la definición de las funciones o servicios que el software proveerá y de las posibles restricciones.

### 5. **Diseño de software:** Descripción de la organización general del sistema y diseño de sus componentes e interacciones. Introducción al concepto de arquitectura de software y a consideraciones del diseño de componentes de software.

### 6. **Construcción de software:** Construcción de la solución de software. Introducción a consideraciones sobre la implementación de software, incluyendo, entre otros temas, consideraciones generales de la construcción de software, y estrategias de reutilización (por ejemplo, desarrollo basado en la nube).

### 7. **Pruebas de software:** El objetivo de las pruebas es mostrar que el software cumple con lo que se espera de su funcionamiento, encontrar comportamiento no esperado y apoyar en la validación de su adecuación al uso.

### 8. **Entrega y Soporte de software:** Descripción de los cambios que sufre el software luego de su liberación. Herramientas que se utilizan y su contribución a la ingeniería de software.

## 6. BIBLIOGRAFÍA

El curso puede dictarse en base a dos libros de referencia (ver sección 6.1). Aunque tienen enfoques diferentes ambos libros cubren la temática del curso (se recomienda utilizar solamente uno de ambos libros para todo el curso). A continuación se especifican los capítulos de cada libro para los diferentes temas del curso.

| <b>Tema</b>                              | <b>Libro 1</b>                                          | <b>Libro 2</b>                                                                                            |
|------------------------------------------|---------------------------------------------------------|-----------------------------------------------------------------------------------------------------------|
| Introducción a la Ingeniería de Software | 1. Introduction                                         | 1. Software Products                                                                                      |
| Proceso de Desarrollo de Software        | 2. Software Processes<br>3. Agile software development  | 2. Agile Software Engineering                                                                             |
| Gestión del Desarrollo de Software       | 22. Project management<br>23. Project planning          | 2. Agile Software Engineering                                                                             |
| Requisitos o Funciones del Software      | 4. Requirements engineering<br>5. System modeling       | 3. Features, Scenarios, and Stories                                                                       |
| Diseño de Software                       | 6. Architectural design<br>7. Design and implementation | 4. Software Architecture<br>5. Cloud-Based Software                                                       |
| Construcción de Software                 | 7. Design and implementation                            | 6. Microservices Architecture (opcional)<br>7. Security and Privacy (opcional)<br>8. Reliable Programming |
| Pruebas de Software                      | 8. Software Testing<br>24. Quality management           | 9. Testing                                                                                                |
| Entrega y Soporte de Software            | 9. Software evolution<br>25 Configuration management    | 10. DevOps and Code Management                                                                            |

## 6.1 Básica

1. Sommerville, Ian (2015). Software Engineering (10th Edition). USA: Pearson.
2. Sommerville, Ian (2021). Products: An Introduction to Modern Software Engineering. USA: Pearson.

## 6.2 Complementaria

- Design Patterns by Erich Gamma, Richard Helm, Ralph Johnson, and John Vlissides (Gang of Four - 1994). Addison-Wesley
- Refactoring by Martin Fowler, with Kent Beck (2018). Addison-Wesley.
- Mythical Man-Month, The: Essays on Software Engineering (1995). Frederik Brooks Jr. - Anniversary Edition, Addison-Wesley

## 7. PREVIATURAS

Estructuras de Datos y Algoritmos,  
Bases de Datos 1,  
curso de Programación Avanzada y  
curso de Bases de Datos 2

### Anexo 1: Formas de evaluación

De forma de lograr un aprendizaje continuo, acompañado de una evaluación continua de los conocimientos y habilidades adquiridas, el curso se basa en actividades de taller periódicas, en instancias de evaluación individuales (por lo menos dos) distribuidas coherentemente a lo largo del curso (la última de estas instancias es al final del curso y es una evaluación integradora de conocimientos). Las instancias de evaluación individual buscan evaluar conceptos teóricos y prácticos mediante preguntas teóricas y pequeños ejercicios prácticos.

Distribución de puntajes según instancias de evaluación:

|                           |             |
|---------------------------|-------------|
| Evaluaciones individuales | : 50 puntos |
| Taller                    | : 50 puntos |

El taller se evalúa mediante entregas de ejercicios prácticos y presentaciones de los estudiantes. Para aprobarlo, se deberá tener un mínimo del 60% del puntaje del taller.

### Escala de evaluación:

**Exoneración:** Para exonerar el curso se deben cumplir las siguientes condiciones:

Obtener al menos 60 puntos entre todas las evaluaciones.

Obtener al menos 18 puntos en la suma de las evaluaciones individuales.

Aprobar el Taller, es decir, obtener al menos 30 puntos en el Taller.

**Reprobación:** Si no se cumple alguna de las condiciones anteriores se reprueba el curso.

**Este curso no tiene examen.**