# Características generales

Calidad de Software

**Nombre:** Calidad de Software

**Sigla:** CI-0140

# Créditos: 4

**Horas lectivas**: 5 horas de teoría

**Requisitos:** CI-0126 Ingeniería de Software

**Correquisitos:** Ninguno

**Clasificación:** Curso propio

**Ciclo de carrera:** I ciclo, 4to año (énfasis de Ingeniería de Software)

**Docente(s):** María Isabel Murillo Quintana **Datos de contacto**: [maria.murilloquintana@ucr.ac.cr](mailto:maria.murilloquintana@ucr.ac.cr) **Grupos:** 01

**Semestre y año:** II ciclo 2024

**Modalidad** Bajo virtual

**Horario y lugar de clases:** {classDay} {classStartTime}-{classEndTime} / V 9-11:50 aula {classroom}

**Horario y lugar de consulta:** {consultationDay1} y K {consultationStartTime1}-{consultationEndTime1} ({consultationMode1} oficina {officeNumber1} IF), y {consultationDay2} - V {consultationStartTime2}-{consultationEndTime2}, ({consultationMode2} via Zoom o Teams {officeNumber2})

**Asistente:** Se comunicará posteriormente

# Descripción

Este curso abarca los fundamentos de aseguramiento de la calidad de software, incluyendo métricas, estándares, gestión e implementación de la calidad del software. Está diseñado para complementar los conocimientos y habilidades adquiridos por los estudiantes en el curso previo de Ingeniería de Software. Además, sirve de base para los cursos Práctica Dirigida y Pruebas de Software.

# Objetivos

**Objetivos general**

El *objetivo general* del curso es que los estudiantes adquieran los conocimientos y las habilidades para asegurar la calidad del software, mediante una combinación de estrategias teóricas y prácticas.

# Objetivos específicos

Durante el curso el estudiante desarrollará habilidades para:

* 1. Identificar los principios básicos de calidad de software mediante la discusión de lecturas para reconocer la importancia de la calidad en el software.
  2. Diferenciar entre las métricas para productos, proyectos y procesos de software, mediante la presentación de ejemplos y discusión de lecturas, con el fin de clasificar y seleccionar las más apropiadas en un contexto.
  3. Reconocer los estándares modernos de aseguramiento de calidad mediante su estudio y aplicación en casos prácticos, para identificar las mejores prácticas a nivel internacional.
  4. Explicar las diferentes actividades que permiten verificar y validar la calidad del software, mediante el estudio de casos prácticos donde se hayan aplicado, para elegir las más adecuadas en un contexto.
  5. Describir la estructura organizacional que permite asegurar la calidad del software, mediante la discusión de lecturas y casos de estudio, para determinar los roles de un equipo de expertos de calidad en una organización.
  6. Valorar la utilidad de un modelo de calidad para implementar prácticas de aseguramiento de la calidad del software en una organización, a través del estudio de casos prácticos, para evaluar la idoneidad de un modelo de calidad en un contexto.

# Contenidos

|  |  |  |
| --- | --- | --- |
| **Objetivo** | **Eje temático** | **Desglose** |
| 1 | Calidad del software | Definición y terminología: qué es calidad de software, factores de calidad del software, costo de la calidad, aseguramiento de la calidad del software. |
| 2 | Métricas del software | Definición de medición, medida y métrica. Importancia de las métricas. Métricas de productos, proyectos y procesos de software. Métricas de productividad y calidad. Proceso de medición.  Modelo *Goal-Question-Metric* (GQM). |

|  |  |  |
| --- | --- | --- |
| **Objetivo** | **Eje temático** | **Desglose** |
| 4 | Verificación y validación del software | Estándares internacionales de verificación y validación del software, técnicas y herramientas para verificar y validar la calidad. |
| 5 | Estructura organizacional de la calidad del software | Rol de la Gerencia en la gestión de la calidad. La unidad de aseguramiento de la calidad y su rol organizacional. |
| 6 | Implementación de la calidad | Estrategia para implementar la calidad del software en una organización a través del *Capacity Maturity Model Integrated* (CMMI). |

1. **Metodología**

{methodology}

# Evaluación

{evaluation}

**{@rawXml}**

# Cronograma

{chronogram}

**{@rawChronogram}**

Bibliografía

Libro de texto recomendado:

* 1. Pressman, R., Ingeniería de software: un enfoque práctico, Mc Graw Hill, 7ta. Edición, 2010.

Otra bibliografía de apoyo:

* 1. Braude, E.J., Ingeniería de software: Una perspectiva orientada a objetos, Alfaomega, 1ra. edición, 2003.
  2. Galin, D. Software Quality Assurance. From theory to implementation. Pearson Addison Wesley, 2004. Stephen, Kan, Metrics and Models in Software Quality Engineering. Adisson Wesley, Segunda Edición. 2003.
  3. CMMI Product Team, CMMI for Development Version 1.3, CMMI-DEV, V1.3, The Software Engineering Institute Carnegie Mellon University (www.sei.cmu.edu), CMU/SEI-2010-TR-33, 2010.
  4. IEEE Software Engineering Standards, IEEE Inc., 2008.

# Recursos Estudiantiles

Para información sobre recursos estudiantiles disponibles en la UCR, incluyendo el Sistema de bibliotecas y la normativa universitaria vigente, favor visitar la página: [https://www.ecci.ucr.ac.cr/vida-estudiantil/servicios-institucionales-para-estudiantes/guia-de-](https://www.ecci.ucr.ac.cr/vida-estudiantil/servicios-institucionales-para-estudiantes/guia-de-recursos-estudiantiles-de-la-ucr) [recursos-estudiantiles-de-la-ucr](https://www.ecci.ucr.ac.cr/vida-estudiantil/servicios-institucionales-para-estudiantes/guia-de-recursos-estudiantiles-de-la-ucr).