


| | | |
|--|---|---------------------------|
| <p style="text-align: center;">UNIVERSIDAD AUTÓNOMA DE CHIHUAHUA</p>  <p style="text-align: center;">UNIDAD ACADÉMICA: FACULTAD DE INGENIERÍA</p> <p style="text-align: center;">PROGRAMA ANALÍTICO DE LA UNIDAD DE APRENDIZAJE: <u>INGENIERÍA DE SOFTWARE</u></p> | DES: | INGENIERÍA |
| | Programa académico | Ingeniería en Computación |
| | Tipo de materia (Obli/Opta): | Obligatoria |
| | Clave de la materia: | CO302 |
| | Semestre: | Tercero |
| | Área en plan de estudios: | Específica |
| | Total de horas por semana: | 4 |
| | <i>Teoría: Presencial o Virtual</i> | 4 |
| | <i>Laboratorio o Taller:</i> | 0 |
| | <i>Prácticas:</i> | 0 |
| | <i>Trabajo extra-clase:</i> | 0 |
| | Créditos Totales: | 4 |
| | Total de horas semestre (x sem): | 64 |
| | Fecha de actualización: | Octubre 2024 |
| <i>Prerrequisito (s):</i> | N/A | |

DESCRIPCIÓN:

El curso promueve que el estudiante conozca los procesos, técnicas, métodos y mejores prácticas de ingeniería de software, identificando la importancia del desarrollo de software de calidad, abarcando desde la especificación de requerimientos, diseño, construcción y pruebas de software, y su aplicación en diferentes entornos.

COMPETENCIAS PARA DESARROLLAR:

B4. Transformación Digital

Transforma la cultura digital en la sociedad, en las organizaciones e instituciones educativas para aprovechar al máximo el potencial de las tecnologías y herramientas digitales; propiciar su uso responsable y ético que estimule la creatividad, innovación, la comunicación efectiva y el trabajo colaborativo e interdisciplinar en la solución de problemas de la sociedad digital; promoviendo la privacidad y la seguridad, así como el respeto a los derechos de autor y la propiedad intelectual.

E1. DISEÑO Y DESARROLLO DE SOFTWARE

Utilizar en el diseño y desarrollo de software, integrando algoritmos avanzados y estructuras de datos para crear soluciones de software robustas y de calidad. Implica una comprensión profunda de los principios de programación, un enfoque metódico para la solución de problemas y la capacidad de adaptar y mejorar continuamente las prácticas de desarrollo para satisfacer las cambiantes necesidades tecnológicas y las demandas de los diversos sectores.

| DOMINIOS (Se toman de las competencias) | OBJETOS DE ESTUDIO (Contenidos necesarios para desarrollar cada uno de los dominios) | RESULTADOS DE APRENDIZAJE (Se plantean de los dominios y contenidos) | METODOLOGÍA (Estrategias, secuencias, recursos didácticos) | EVIDENCIAS (Productos tangibles que permiten valorar los resultados de aprendizaje) |
|---|---|--|--|---|
| <p>B4.2 Utiliza de forma responsable las tecnologías de la información, comunicación, conocimiento y aprendizaje (TICCA), en el proceso de construcción de saberes y el desarrollo de proyectos sociales innovadores en el ámbito digital.</p> <p>ESPECÍFICAS E1.DISEÑO Y DESARROLLO DE SOFTWARE</p> <p>* Proponer soluciones innovadoras en el diseño y desarrollo de software, en diferentes plataformas y dispositivos, aplicando procesos, métodos y mejores prácticas de ingeniería de software, para desarrollar proyectos medibles, repetibles y de</p> | <p>UNIDAD 1. CONCEPTOS FUNDAMENTALES DE INGENIERÍA DE SOFTWARE.</p> <p>1.1 El desarrollo profesional del software 1.2 Historia de la Ingeniería de Software 1.3 Necesidad de la ingeniería de software</p> | <p>Distingue la diferencia entre los principales conceptos de ingeniería de software.</p> <p>Reconoce la importancia del desarrollo de software profesional, considerando la calidad, disciplina, administración y procesos.</p> <p>Analiza los aspectos que dieron origen a la ingeniería de software e identifica la necesidad de la ingeniería de software.</p> | <p>Exposiciones del profesor</p> <p>Búsqueda y análisis de información</p> <p>Elaboración de ensayos</p> | <p>Mapa conceptual</p> <p>Línea del tiempo</p> <p>Ensayo</p> <p>Examen teórico</p> <p>Cada actividad debe incluir rubricas y/o listas de cotejo</p> |

calidad.

| | | | |
|--|--|--|--|
| <p>UNIDAD 2. INGENIERÍA DE REQUERIMIENTOS</p> <p>3.1 Concepto de ingeniería de requerimientos</p> <p>3.1.1 Definición</p> <p>3.1.2 Proceso</p> <p>3.1.3 Fuentes de requerimientos</p> <p>3.1.4 Tipos de requerimientos.</p> <p>3.2 Técnicas para obtención, análisis y validación de requerimientos.</p> <p>3.2.1 Entrevistas</p> | <p>Identifica los diferentes tipos de requerimientos de un producto de software.</p> <p>Conoce el proceso de obtención, análisis, especificación y validación de requerimientos de software.</p> | <p>Búsqueda y análisis de información</p> <p>Exposiciones del profesor</p> | <p>Documento de especificación de requerimientos</p> <p>Examen teórico</p> |
|--|--|--|--|

| | | | | |
|--|---|--|---|--|
| | <p>3.2.2 Encuestas 3.2.3 Documentos o sistemas existentes 3.2.4 Mapas de historias de usuario 3.2.5 Tablas o árboles de decisión</p> <p>3.3 Especificación de requerimientos de software</p> | | | |
| | <p>UNIDAD 3. DISEÑO DE SOFTWARE</p> <p>4.1 Diseño arquitectónico 4.1.1 Estilos y patrones arquitectónicos</p> <p>4.2 Diseño de procesos</p> <p>4.3 Diseño de interfaces de usuario 4.3.1 Proceso de diseño de interfaces</p> | <p>Comprende los distintos tipos de patrones de arquitectura de software y su aplicación para determinadas soluciones de problemas en la industria de software.</p> <p>Representa procesos del dominio por medio de diagramas utilizando herramientas de software.</p> <p>Diseña interfaces de usuario siguiendo un proceso y considerando los requerimientos de usuario, entendiendo criterios de usabilidad.</p> | <p>Búsqueda y análisis de información</p> <p>Exposiciones del profesor</p> <p>Aprendizaje orientado en proyectos</p> <p>Diseño y modelado de diagramas e interfaces</p> | <p>Diagrama de procesos</p> <p>Diseño de interfaces de usuario</p> <p>Examen teórico</p> |
| | <p>UNIDAD 4. CONSTRUCCIÓN DEL SOFTWARE</p> <p>5.1 Modelos de proceso de software</p> <p>5.2 Metodologías ágiles 5.2.1 SCRUM</p> | <p>Conoce los distintos procesos de desarrollo y su utilización en la industria.</p> <p>Entiende las prácticas de desarrollo ágil, reconociendo la importancia de su</p> | <p>Búsqueda y análisis de información</p> <p>Exposiciones del profesor</p> | <p>Examen teórico</p> <p>Exposiciones del alumno</p> |

| | | | | |
|--|--|---|--|---|
| | | aplicación en la actualidad para poder cumplir con la creciente demanda de software. | | |
| | <p>UNIDAD 5. PRUEBAS DE SOFTWARE.</p> <p>6.1 Fundamentos de pruebas</p> <p>6.1.1 Pruebas</p> <p>6.1.2 Casos de pruebas</p> <p>6.1.3 Error, defecto y fallo</p> <p>6.1.4 Testware</p> <p>6.1.5 Ingeniero de pruebas</p> <p>6.1.6 Reportes de pruebas</p> <p>6.1.7 Verificación y Validación</p> <p>6.2 Niveles de pruebas</p> <p>6.2.1 Pruebas unitarias</p> <p>6.2.2 Pruebas integrales</p> <p>6.2.3 Pruebas de sistema</p> <p>6.2.4 Pruebas de aceptación</p> <p>6.3 Tipos de pruebas</p> <p>6.3.1 Funcionales</p> <p>6.3.2 No funcionales</p> <p>6.3.3 Estructurales</p> <p>6.3.4 Asociadas al cambio</p> <p>6.4 Técnicas de pruebas</p> <p>6.4.1 Estáticas</p> <p>6.4.2 Dinámicas</p> <p>6.4.3 Caja negra</p> <p>6.4.4 Caja blanca</p> | <p>Comprende los conceptos básicos de pruebas.</p> <p>Entiende la necesidad de probar el software.</p> <p>Entiende los distintos niveles, tipos y técnicas de pruebas como instrumentos de aseguramiento de la calidad.</p> | <p>Búsqueda y análisis de información</p> <p>Exposiciones del profesor</p> | <p>Mapa conceptual</p> <p>Reportes de pruebas</p> <p>Examen teórico</p> |

| FUENTES DE INFORMACIÓN (Bibliografía, direcciones electrónicas) | EVALUACIÓN DE LOS APRENDIZAJES (Criterios, ponderación e instrumentos) |
|--|--|
| <ol style="list-style-type: none"> Gomez Fuentes M. (2019). Fundamentos de la Ingeniería de Software. Universidad Autónoma Metropolitana. ISBN 978-607-28-1659-6. Pantaleo, G., Rinaudo, L. (2015). <i>Ingeniería de Software</i>. Alpha Editorial. ISBN 6076222379, 9786076222379. Gómez Palomo, S. R., & Moraleda Gil, E. (2020). <i>Aproximación a la ingeniería del software</i>. Editorial Centro de Estudios Ramón Areces SA. ISBN 8499613292, 9788499613291. | <p>Para cada uno de los parciales se considera:</p> <p>Ejercicios y tareas: 40%</p> <p>Examen teórico: 60%</p> <p>La calificación mínima es 7.0</p> <p>Las actividades asignadas, así como tareas deben presentar rúbricas y/o listas de cotejo.</p> |

CRONOGRAMA

| Objetos de estudio | Semanas | | | | | | | | | | | | | | | |
|---|---------|---|---|---|---|---|---|---|---|----|----|----|----|----|----|----|
| | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | 14 | 15 | 16 |
| UNIDAD 1. CONCEPTOS FUNDAMENTALES DE INGENIERÍA DE SOFTWARE. | | | | | | | | | | | | | | | | |
| UNIDAD 2. INGENIERÍA DE REQUERIMIENTOS | | | | | | | | | | | | | | | | |
| UNIDAD 3. DISEÑO DE SOFTWARE | | | | | | | | | | | | | | | | |
| UNIDAD 4. CONSTRUCCIÓN DEL SOFTWARE | | | | | | | | | | | | | | | | |
| UNIDAD 5. PRUEBAS DE SOFTWARE. | | | | | | | | | | | | | | | | |