



1. Datos Generales de la asignatura

Nombre de la asignatura:	Ingeniería de software
Clave de la asignatura:	DAD-2414
SATCA¹:	2-3-5
Carrera:	Ingeniería en Desarrollo de Aplicaciones.

2. Presentación

Caracterización de la asignatura

Esta asignatura aporta al perfil del ingeniero en desarrollo de aplicaciones las siguientes habilidades:

- Analiza, diseña y construye aplicaciones web, móvil o de cómputo en la nube que satisfacen requerimientos de los usuarios, mediante la aplicación de principios, herramientas, prácticas, metodologías y tecnologías emergentes.
- Diseña interfaces de manera responsable y con sentido ético, considerando las buenas prácticas y herramientas apropiadas, para facilitar la interacción adecuada con las aplicaciones.
- Formula, diseña, ejecuta y evalúa proyectos de desarrollo de aplicaciones orientados a la satisfacción de necesidades previamente detectadas en las organizaciones, el entorno o la sociedad misma.
- Desarrolla proyectos de inversión viables, empleando como una de las estrategias de solución el uso de nuevas tecnologías, para la satisfacción de necesidades de las organizaciones y la sociedad.

Esta asignatura es fundamental en el contexto actual, donde la presencia en línea es esencial para las organizaciones. La correcta aplicación de principios y técnicas de ella garantizan la creación de aplicaciones web funcionales, seguras y eficientes, lo que se traduce en una mejor experiencia para los usuarios y en el cumplimiento de los objetivos del negocio. A través de un enfoque práctico y teórico, los estudiantes explorarán desde sus fundamentos hasta aspectos avanzados como la seguridad, la calidad del software y el proceso de desarrollo de aplicaciones.

Para abordar de manera adecuada su contenido son necesarios los conocimientos de fundamentos de investigación, fundamentos de ingeniería de software, programación orientada a objetos, diseño web y administración gerencial. Se relaciona, posteriormente, con la asignatura denominada Aseguramiento de la Calidad del Software.

¹ Sistema de Asignación y Transferencia de Créditos Académicos



Intención didáctica

Ingeniería de Software, al igual que Fundamentos de Ingeniería de Software, sigue siendo una asignatura donde convergen los conocimientos y habilidades técnicas que se desarrollan para las tecnologías de la información con los conceptos y habilidades económico-administrativas que aportan las ciencias económicas. Por lo tanto, este programa ha sido desarrollado para que los estudiantes vayan viviendo la experiencia de cada una de las etapas del desarrollo de software desde un enfoque administrativo y vaya aplicando tecnologías competentes en cada una de ellas desde un enfoque tecnológico, por lo cual, el docente no debe dejarse llevar por su pasión sobre las tecnologías de la información si es que las tiene.

Dado lo anterior, se recomienda que el enfoque de la asignatura sea integral, considerando los aspectos técnicos, éticos y administrativos, tratando de desarrollar en los estudiantes una proclividad hacia la calidad en el desarrollo de aplicaciones web. Asimismo, se sugiere que se promueva el aprendizaje activo y colaborativo, fomentando la participación de los estudiantes en discusiones del proyecto que seguirán desarrollando dentro de esta asignatura y que traen desde Fundamentos de Ingeniería de Software.

Asimismo, se aconseja al maestro que los contenidos sean tratados con la extensión y profundidad necesaria para que los estudiantes adquieran una sólida base teórica, práctica, administrativa y técnica en el desarrollo de aplicaciones web, dándose énfasis en la comprensión de conceptos clave y en la aplicación adecuada de los procesos estudiados en cada etapa abarcada de la ingeniería de software, apoyándose en las herramientas tecnológica para dar vida real a los modelos que desarrolla en cada una de ellas.

Se asume que el docente desempeñará un papel de facilitador y guía, promoviendo el aprendizaje activo y colaborativo, brindará orientación y asesoramiento en la resolución de problemas y la aplicación de conceptos y herramientas en el desarrollo de aplicaciones web y el espíritu crítico y reflexivo en los estudiantes, estimulando el pensamiento creativo y la innovación en el campo de la ingeniería de software.

En otros asuntos, se expresa que el contenido temático de la asignatura se alinea con la bibliografía de (Braude & Bernstein, 2016), (PRESSMAN S. & Maxim, 2019), (Pressman & Maxim, 2019). No obstante, lo anterior, se han agregado algunas fuentes bibliográficas más para efectos de ejemplificación en los temas y para permitir al catedrático elegir los libros que más adecuados le parezcan o que estén conforme a su forma de pensar.

Dado que se pide trabajar en equipo dentro de la asignatura y que se desea que los estudiantes vivan el proceso de desarrollo de software desde el enfoque de la ingeniería de software, el catedrático podría pedirles que asuman puestos similares a los temas vistos en ella, tal como diseñador, desarrollador, inspector de seguridad e inspector de calidad. Conforme a esos puestos, se les puede solicitar que estudien algo más para poder ser más productivos en la etapa o actividad que les corresponde, debiendo tener como responsabilidad los conocimientos y habilidades impartidos en clase para los temas que no les toquen.



Con ello, se fomenta la división del trabajo y la especialización, permitiéndoles vivir el proceso estudiado en la asignatura y tener fortalezas en las unidades que les toca como responsabilidad.

Para efectos de valoración de los resultados de cada unidad estudiada, se recomienda que el catedrático haga una rúbrica o un instrumento de evaluación donde los equipos auto califiquen los resultados que obtienen, reciban retroalimentación de dos equipos y de él mismo. La idea es acostumbrar a que los estudiantes sean autocríticos y que vayan adquiriendo la cultura de la calidad en los trabajos que realizan.

En lo relativo a las unidades que la integran, está integrada por cuatro temas que avanzan, progresivamente, hacia la obtención de una aplicación web, las cuales son diseño, implementación, introducción a la seguridad de software y pruebas. Dados que las actividades a seguir ya se especificaron y ya se brindaron las sugerencias requeridas, no se emite ninguna recomendación más salvo el hecho de que en la implementación el catedrático deberá elegir las herramientas a emplear, considerando el tiempo disponible para obtener resultados y para aprenderlas. Esto quiere decir que, si considera que un framework o cualquier otra combinación de herramientas permitirá a los estudiantes llegar con mayor rapidez a los resultados, puede emplearlo sin ningún problema.

Para terminar, se expresa que las competencias genéricas que más se desarrollan dentro de la asignatura son trabajo en equipo, capacidad de análisis y síntesis, así como capacidad de aprender.

3. Participantes en el diseño y seguimiento curricular del programa

Lugar y fecha de elaboración o revisión	Participantes	Observaciones
Tecnológico Nacional de México 23 de octubre del 2023	Representantes del Instituto Tecnológico de Chetumal.	Presentación de la propuesta de la carrera de Ingeniería en Desarrollo de Aplicaciones.
Instituto Tecnológico de Querétaro Campus Norte del 19 al 22 de marzo 2024.	Representantes de los Institutos Tecnológicos de: Celaya, Chetumal, Ensenada, La Zona Olmeca, Querétaro, Villahermosa. Tecnológicos Superiores de: Huetamo, Mario Molina Pasquel y Henríquez (unidad Mascota), Purhépecha. Representante de Ciencias Básicas de los Institutos de: Celaya, Morelia y CIIDET.	Diseño y/o desarrollo curricular de la carrera de Ingeniería en Desarrollo de Aplicaciones.



Tecnológico Nacional de México del 22 al 24 de abril del 2024.	Representantes de los Institutos Tecnológicos de Chetumal y Villahermosa.	Contraste y ajuste de las asignaturas Ing. en Desarrollo de Aplicaciones con respecto a las de Ing. en Inteligencia Artificial, Ing. en Ciberseguridad e Ingeniería en Ciencia de Datos.
Tecnológico Nacional de México del 27 al 31 de mayo de 2024	Representantes de los Institutos Tecnológicos de: Celaya, Chetumal, Ensenada, La Zona Olmeca, Villahermosa.	Consolidación curricular de la carrera de Ingeniería en Desarrollo de Aplicaciones.

4. Competencia(s) a desarrollar

Competencia(s) específica(s) de la asignatura
Produce aplicaciones web de calidad, empleando conocimientos, buenas prácticas y principios de la ingeniería de software, para satisfacer necesidades de las organizaciones y la sociedad.

5. Competencias previas

- Elabora el modelo de requisitos, satisfaciendo políticas de calidad y aplicando los conceptos y procesos de la ingeniería de software, para el diseño e implementación adecuados de una aplicación.
- Aplica la programación orientada a objetos para resolver problemas reales y de ingeniería.
- Diseña sitios web, aplicando lenguajes de marcado y de presentación, así como elementos de diseño de web, para la satisfacción apropiada de necesidades de la sociedad y las organizaciones.
- Analiza los requerimientos, manipula bases de datos para brindar soluciones al tratamiento de información basándose en modelos y estándares; de igual modo, implementa bases de datos para apoyar la toma de decisiones considerando las reglas de negocio.
- Diseña interfaces de interacción entre el usuario y el software, considerando las tendencias en este ámbito, aplicando herramientas para su construcción y empleando evaluaciones competentes, para el logro de altos niveles de aceptabilidad, usabilidad y de buena experiencia en el uso de ellas.



6. Temario

No.	Temas	Subtemas
1	Diseño	<ul style="list-style-type: none">1.1. Diseño<ul style="list-style-type: none">1.1.1. Conceptos de diseño.1.1.2. Diseño de la arquitectura.1.1.3. Diseño a nivel componente.1.1.4. Patrones de diseño.1.1.5. Diseño para la movilidad.1.2. Métricas y calidad en el diseño.<ul style="list-style-type: none">1.2.1. Grado de entendimiento, cohesión y acoplamiento.1.2.2. Grado de suficiencia como meta de diseño1.2.3. Grado de flexibilidad como meta de calidad en el diseño1.2.4. Grado de reusabilidad como meta de calidad en el diseño1.2.5. Grado de eficiencia de tiempo como una medida de calidad en el diseño.
2	Implementación.	<ul style="list-style-type: none">2.1. Principios de implementación.<ul style="list-style-type: none">2.1.1. Enfoque para la implementación.2.1.2. Elección de la herramienta de programación.2.1.3. Consejos y trucos para la programación.2.1.4. Identificación de las clases.2.1.5. Definición de métodos.2.1.6. Prácticas de implementación.2.1.7. Programación defensiva.2.1.8. Estándares de codificación.2.1.9. Comentarios.2.1.10. Herramientas y ambientes para programación.2.2. Métricas y calidad en la implementación.<ul style="list-style-type: none">2.2.1. Calidad de la implementación.2.2.2. Inspecciones de código y procedimientos relacionados con la calidad.



3	Introducción a la seguridad de software.	<ul style="list-style-type: none">3.1. Ingeniería en seguridad de software.<ul style="list-style-type: none">3.1.1. Importancia de la seguridad de la información.3.1.2. Modelos de ciclos de vida de la seguridad.3.1.3. Ingeniería de requisitos de la seguridad.3.1.4. Casos de mal uso y abuso, así como de patrones de ataque.3.1.5. Análisis de riesgos de seguridad.3.1.6. Ataque superficial.3.1.7. Código seguro.3.1.8. Medición.3.1.9. Mejora del proceso de seguridad y modelo de madurez.3.2. Seguridad en los datos.<ul style="list-style-type: none">3.2.1. Introducción a la encriptación de datos.3.2.2. Modelos de encriptación de datos AES, MD5, HASHING.3.3. Protocolo de comunicación web seguros (HTTPS, SFTP, SSL/TLS, etc.).3.4. Certificados web.<ul style="list-style-type: none">3.4.1. Funcionamiento y tipos de certificados Web.3.4.2. Proceso de obtención e instalación de certificados Web.3.4.3. Importancia de los certificados web en la seguridad de las comunicaciones.
4	Pruebas de software.	<ul style="list-style-type: none">4.1. Introducción.4.2. Pruebas a nivel componente.4.3. Pruebas a nivel integración.4.4. Pruebas para la movilidad.



7. Actividades de aprendizaje de los temas

1. Diseño	
Competencias	Actividades de aprendizaje
<p><i>Específica(s):</i> Diseña, con calidad, una aplicación web, aplicando conceptos, filosofías, estrategias y metodologías del diseño de software, para su posterior construcción.</p> <p><i>Genérica(s):</i></p> <ul style="list-style-type: none">• Trabajo en equipo• Capacidad de análisis y síntesis• Capacidad de aprender	<p>En equipo de trabajo, considerando la elección e indicaciones del catedrático, realizar la siguiente:</p> <ul style="list-style-type: none">• Autoevaluarse y solicitar a dos equipos más para que valoren la calidad del modelo de requisitos previamente elaborado en Fundamentos de Ingeniería de Software, haciendo uso de una rúbrica o instrumento de evaluación diseñado por el profesor.• Corregir el modelo de requisitos conforme a los lineamientos especificados por el catedrático.• Elaborar un plan de trabajo, conforme a los días y horas proyectados a emplear en el semestre, aplicando una herramienta tal como Microsoft Project o Gantt Project.• Acordar con el maestro los tipos de modelo de diseño a aplicar para el caso de la aplicación que van a construir y proceder a su realización.• Dar seguimiento al programa de trabajo semanalmente, teniendo reuniones para tomar medidas correctivas con relación a la desviación.• Proceder al diseño de la aplicación, conforme a lo estudiado en clases y la bibliografía señalada por el maestro, evaluando objetivamente los resultados mediante la aplicación de una rúbrica o instrumento de evaluación proporcionada por el profesor. Si los resultados no cumplen con los requisitos mínimos o la puntuación establecida previamente, entonces, realizar este proceso hasta que sea aceptable.• Solicitar la evaluación del catedrático para hacer las correcciones correspondientes• Terminar el diseño de la aplicación, procediendo a elaborar el documento de diseño conforme a las características establecidas por el profesor.



2. Implementación.	
Competencias	Actividades de aprendizaje
<p>Específica(s): Elabora una aplicación, a partir del diseño previamente elaborado y aplicando buenas prácticas de implementación, para la solución de un problema de una organización o de la sociedad.</p> <p>Genérica(s):</p> <ul style="list-style-type: none">• Trabajo en equipo.• Capacidad de análisis y síntesis.• Capacidad de aprender.	<p>En equipo de trabajo, considerando la elección e indicaciones del catedrático, realizar la siguiente:</p> <ul style="list-style-type: none">• Concluir, junto con el maestro, las mejores herramientas de desarrollo que podrían ser empleadas para la construcción de la aplicación.• Si es necesario, capacitarse en aquellas herramientas o conceptos que son requeridos para lograr la construcción de la aplicación.• Elaborar un programa de inspecciones de los resultados parciales o terminados para ir valorando al apego a las buenas prácticas aprendidas en clase y la obtención de los resultados requeridos.• Proceder a la construcción de la aplicación deseada, aplicando las buenas prácticas estudiadas en la unidad para el tipo de software que se desarrolla. Debe construirse cada uno de los tipos de modelos que se elaboraron en el diseño, vinculándolo con los otros tipos de modelos competentes.• En la medida de lo posible, ir probando el comportamiento y los resultados obtenidos por las unidades de código generadas.• Aplicar una evaluación de la aplicación obtenida, usando una rúbrica o instrumento de evaluación diseñado por el catedrático. Si es necesario, corregir los errores encontrados en cada uno de los tipos de modelos implementados.• Solicitar al catedrático la evaluación de la aplicación obtenida, debiendo aplicar la rúbrica o instrumento de evaluación elaborado por él. Si es necesario, realizar las correcciones pertinentes.
3. Introducción a la seguridad de software	
Competencias	Actividades de aprendizaje
<p>Específica(s): Planea las actividades requeridas, aplicando los conceptos y buenas prácticas de ingeniería de software, para la evaluación de la seguridad de una aplicación.</p>	<p>En equipo de trabajo, considerando la elección e indicaciones del catedrático, realizar la siguiente:</p> <ul style="list-style-type: none">• Elaborar un resumen de los capítulos, elegidos por el maestro, del libro de (Ortega Candel, 2020).

<p>Genérica(s):</p> <ul style="list-style-type: none"> • Trabajo en equipo • Capacidad de análisis y síntesis • Capacidad de aprender 	<ul style="list-style-type: none"> • Elaborar un resumen de los capítulos, seleccionados por el profesor, del libro (Reddy , 2023) • Establecer un plan de mejora de la aplicación desarrollada, realizando las actividades que puedan ejecutarse dentro del tiempo disponible, previa consulta con el catedrático. • Dar seguimiento al avance de las actividades programadas para la construcción de la aplicación.
4. Pruebas de software	
Competencias	Actividades de aprendizaje
<p>Específica(s): Planea las pruebas requeridas, aplicando conocimientos y buenas prácticas de la Ingeniería de Software, para la garantía de un nivel adecuado de calidad de una aplicación.</p> <p>Genérica(s):</p> <ul style="list-style-type: none"> • Trabajo en equipo • Capacidad de análisis y síntesis • Capacidad de aprender 	<p>En equipo de trabajo, considerando la elección e indicaciones del catedrático, realizar la siguiente:</p> <ul style="list-style-type: none"> • Elaborar un resumen de los capítulos, escogidos por profesor, de libro de (Mann. 2022). • Elaborar un resumen de los capítulos, seleccionados por el maestro, del libro de (Nexus AI, 2024). • Elaborar un pequeño plan de pruebas para verificar la calidad de la aplicación desarrollada, considerando las sugerencias del catedrático. • Considerando las sugerencias del maestro y el tiempo disponible, aplicar pruebas automatizadas, empleando la herramienta estudiada, procediendo a mejorarla si se encuentran errores. • Documentar la aplicación y hacer una presentación de los resultados de todo el proceso seguido para el catedrático.

8. Práctica(s)

Dado el sentido práctico de la asignatura, se recomienda que los estudiantes, organizados en equipos de trabajo, se dediquen a la elaboración del análisis de requisitos desde la segunda unidad para que vayan presentando sus avances al catedrático y tengan la oportunidad de mejorar sus resultados. Sin embargo, si el catedrático percibe que el grupo requiere que realice prácticas de puede elegir algunas de las que se encuentran en el libro de (García Chi et al., 2020).

9. Proyecto de asignatura

El objetivo del proyecto que planteé el docente que imparta esta asignatura, es demostrar el desarrollo y alcance de la(s) competencia(s) de la asignatura, considerando las siguientes fases:

Fundamentación: marco referencial (teórico, conceptual, contextual, legal) en el cual se fundamenta el proyecto de acuerdo con un diagnóstico realizado, mismo que permite a los estudiantes

lograr la comprensión de la realidad o situación objeto de estudio para definir un proceso de intervención o hacer el diseño de un modelo.

Planeación: con base en el diagnóstico en esta fase se realiza el diseño del proyecto por parte de los estudiantes con asesoría del docente; implica planificar un proceso: de intervención empresarial, social o comunitario, el diseño de un modelo, entre otros, según el tipo de proyecto, las actividades a realizar los recursos requeridos y el cronograma de trabajo.

Ejecución: consiste en el desarrollo de la planeación del proyecto realizada por parte de los estudiantes con asesoría del docente, es decir en la intervención (social, empresarial), o construcción del modelo propuesto según el tipo de proyecto, es la fase de mayor duración que implica el desempeño de las competencias genéricas y específicas a desarrollar.

Evaluación: es la fase final que aplica un juicio de valor en el contexto laboral-profesión, social e investigativo, ésta se debe realizar a través del reconocimiento de logros y aspectos a mejorar se estará promoviendo el concepto de “evaluación para la mejora continua”, la metacognición, el desarrollo del pensamiento crítico y reflexivo en los estudiantes.

10. Evaluación por competencias

Las técnicas, herramientas y/o instrumentos sugeridos que permiten obtener el producto del desarrollo las actividades de aprendizaje:

- Códigos.
- Exposición.
- Mapas mentales o conceptuales.
- Resúmenes.
- Reportes de prácticas.
- Cuadros comparativos.
- Informes.

Las técnicas, herramientas y/o instrumentos recomendados, que, como docente, permite constatar el logro o desempeño de las competencias del estudiante:

- Rúbricas.
- Listas de cotejo.
- Listas de verificación.
- Guías de observación.



11. Fuentes de información

1. Braude, E.J., & Bernstein, M. E. (2016). Software Engineering. Modern Approaches. Second Edition. Waveland Press, Inc.
2. Braude, E. J., & Bernstein, M. E. (2016). Software Engineering: Modern Approaches, Second Edition. Waveland Press, Inc.
3. Dennis, A., Wixom, B., & Tegarden, D. (2020). Systems Analysis and Design: An Object-Oriented Approach with UML. 6th Edición. Wiley.
4. Dosaj, C. R. (2023). El probador de software autodidacta Una guía paso a paso para aprender a probar software usando proyectos de la vida real (Spanish Edition). Independently published.
5. Foster, E., & Towle Jr., B. (2021). Software Engineering: A Methodical Approach, 2nd Edition. Auerbach Publications.
6. García Chi, R. I., Morales Vázquez, M. G., & Hernández, M. A. (2020). Estrategias didácticas de Ingeniería de Software: ¡A practicar!: Tecnológico Nacional de Méxicocampus Cd.Valles. Editorial Académica Española.
7. Mall, R. (2018). FUNDamentals of Software Engineering- Fifth Edition. Asoke K. Ghosh.
8. Mann, R. S. (2022). Pruebas de software y garantía de calidad (Spanish Edition). Ediciones Nuestro Conocimiento.
9. Montilva C., J. (2021). Ingeniería del Software: Un enfoque basado en procesos (Spanish Edition). Universidad de Los Andes.
10. Nexus AI. (2024). Selenium Mastery: A Comprehensive Guide to Web Browser Automation and Testing. Independently published.
11. Ortega Candell, J. M. (2020). Desarrollo Seguro En Ingeniería Del Software. Marcombo.
12. Pinzón Amaya, S. (2023). La ciberseguridad en las pruebas de software: Generalidades (Spanish Edition). Independently published.
13. PRESSMAN S., R., & Maxim, B. R. (2019). Ingeniería de Software. Un Enfoque Práctico. McGraw-Hill Interamericana de España S.L.
14. Pressman, R. S., & Maxim, B. R. (2019). Software Engineering: A Practitioner's Approach. McGraw Hill.
15. Reddy, P. (2023). Seguridad en la Web: Aprendizaje automático para la detección de vulnerabilidades CSRF: Aprovechar los algoritmos de aprendizaje automático para una detección y prevención precisas. Ediciones Nuestro Conocimiento.
16. Reibel, J.-J. (2023). Software Engineering. Independently published.
17. Stephens, R. (2022). Beginning Software Engineering 2nd Edición. Wiley.
18. Suman, U. (2022). Software Engineering: Concepts and Practices, 2E. Cengage India.