



1. Datos Generales de la asignatura.

Nombre de la asignatura:	Arquitecturas Web.
Clave de la asignatura:	DAD-2402
SATCA¹:	2-3-5
Carrera:	Ingeniería en Desarrollo de Aplicaciones.

2. Presentación.

Caracterización de la asignatura.
<p>Esta asignatura aporta al perfil del egresado las competencias que le faciliten el uso de tecnologías y herramientas actuales y emergentes acordes a las necesidades del entorno que le permitan desarrollar e implementar API's y Microservicios que agilicen los procesos de integración y comunicación entre sistemas, optimizando la eficiencia y la productividad en el ámbito empresarial mejorando la experiencia del usuario.</p> <p>Con relación al perfil del egresado realiza contribuciones a la competencia que señala que este:</p> <ul style="list-style-type: none">● 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.● 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. <p>Es de suma importancia porque permite al estudiante conocer y aplicar las principales especificaciones de arquitecturas de servicios web, así como los diferentes protocolos que intervienen en la interoperabilidad de las diversas plataformas, con el objetivo de crear sistemas seguros y estables que permitan la comunicación entre diferentes entornos de desarrollo.</p> <p>En esta asignatura se abordan las bases teóricas y prácticas referentes a las arquitecturas web, a la arquitectura de servicios SOAP, a la arquitectura de servicios REST, a los microservicios y a los tópicos avanzados de desarrollo; aplica conocimientos de otras asignaturas.</p>

¹ Sistema de Asignación y Transferencia de Créditos Académicos



Intención didáctica.

Esta asignatura con el fin de obtener las competencias debe centrarse en la realización de prácticas e implementación de casos de estudio que ofrezcan diversos escenarios que ayuden en la aplicación de los conceptos, técnicas, arquitecturas, modelos y marcos de trabajo relacionados con las API's y los microservicios para la resolución de los mismos.

La asignatura está organizada en cinco temas:

El primer tema se centra en los conceptos, arquitectura, modelos e integración de SOA que agilicen los procesos de integración y comunicación entre sistemas.

El segundo aborda, la arquitectura de servicios SOAP y los elementos que lo conforman, con la finalidad de que el estudiante pueda establecer la comunicación entre las aplicaciones que se diseñan con diferentes lenguajes y en distintas plataformas.

La arquitectura de servicios REST, es el tercer tema que se aborda en esta asignatura, la cual ofrece un buen desempeño, escalabilidad y abstracción de recursos, donde cada petición HTTP contiene toda la información necesaria para responder a la petición, sin necesidad que el cliente ni el servidor tenga que recordar el estado de su comunicación.

En el cuarto tema el estudiante, se centra en la arquitectura basada en microservicios, la cual es una arquitectura que permite diseñar aplicaciones cuyos elementos funcionan de forma independiente, pero de manera coordinada.

El estudiante en el quinto tema, se enfoca en estudiar las alternativas para la entrega y despliegue del software, considerando temas avanzados como son: integración continua, ambientes de desarrollo, distribución y patentado.

La importancia de la asignatura se centra en conocer y desarrollar cada uno de los temas que la conforman, por lo que se recomienda que el estudiante desarrolle diversas prácticas en cada tema visto en clase generando un proyecto integrador y poniendo atención en los avances de los mismos.

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 Básica 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
Desarrolla servicios web modernos y escalables utilizando arquitecturas orientadas a servicios y aplica metodologías de entrega y despliegue continuo que garantice un ciclo de desarrollo eficiente y de alta calidad en la creación de software robusto y adaptable.

5. Competencias previas

Diseña, con calidad, una aplicación web, aplicando conceptos, filosofías, estrategias y metodologías del diseño de software, para su posterior construcción.
Crea aplicaciones web interactivas y dinámicas, utilizando un lenguaje de programación del lado del servidor; técnicas avanzadas de acceso a datos; el intercambio de datos mediante JSON, XML y AJAX; servicios web apropiados; así como interfaces de usuario atractivas y funcionales; para la satisfacción de necesidades de la sociedad y las organizaciones.

6. Temario

No.	Temas	Subtemas
1	Arquitectura Web.	<ul style="list-style-type: none">1.1. Computación orientada al servicio.1.2. Arquitectura orientada al servicio.1.3. Modelos de servicios.1.4. Integración de aplicaciones: EAI (Enterprise Application Integration) v.s. Middleware.1.5. Ingeniería de software orientada al servicio.1.6. Formatos de representación de recursos: XML, JSON, Atom.



2	Arquitectura de servicios SOAP.	<ul style="list-style-type: none">2.1. SOAP.2.2. Definición de servicios: WSDL.2.3. Formato de mensajes y protocolo de mensajería: SOAP.2.4. Descripción, publicación, descubrimiento e integración: UDDI.2.5. Invocación de servicios. gestión del estado del servicio.2.6. Soporte de una plataforma para el desarrollo de servicios Web SOAP.
3	Arquitectura de servicios REST.	<ul style="list-style-type: none">3.1. El estilo arquitectónico REST.3.2. La Arquitectura orientada a recursos (ROA).3.3. Definición del modelo de recursos.3.4. Diseño de URIs y Clientes REST.3.5. Diseño e implementación de servicios RESTful orientados a recursos.3.6. Mashups de servicios.3.7. Soporte de una plataforma para el desarrollo de Servicios Web RESTful.3.8. Marcos de trabajo.
4	Microservicios.	<ul style="list-style-type: none">4.1. Principios arquitectónicos de los microservicios.<ul style="list-style-type: none">4.1.1. Enfoques de modularización.4.1.2. Ventajas y desventajas.4.1.3. Técnicas de modularización.4.1.4. Equilibrio entre redundancia e integridad.4.1.5. Orquestación vs Coreografía.4.2. Orquestación de servicios.4.3. Integración de servicios usando coreografía.4.4. Registro y auto-reconocimiento de servicios.4.5. Balanceo de carga en cliente.4.6. Patrón Circuit Breaker (corto circuito).4.7. Contenedores.4.8. Marcos de trabajo.



5	Tópicos avanzados de desarrollo.	5.1. Manejo de ambientes. 5.1.1. Desarrollo. 5.1.2. Testeo. 5.1.3. Preproducción. 5.1.4. Producción. 5.2. Integración continua de software. 5.3. Entrega continua de software. 5.4. Configuración del despliegue. 5.5. Empaquetado y distribución. 5.6. Patentes y licenciamientos de software.
---	----------------------------------	--

7. Actividades de aprendizaje de los temas

1. Arquitectura Web	
Competencias	Actividades de aprendizaje
<p>Específica(s): Identifica y reconoce la importancia de la arquitectura orientada al servicio.</p> <p>Genéricas:</p> <ul style="list-style-type: none"> • Capacidad de abstracción, análisis y síntesis. • Comunicación oral y escrita. • Habilidad para buscar, procesar y analizar información procedente de fuentes diversas. • Capacidad de trabajo en equipo. • Compromiso ético. • Capacidad de aplicar los conocimientos en la práctica. • Habilidad para trabajar en forma autónoma. 	<ul style="list-style-type: none"> • Investigar los conceptos fundamentales, objetivos, usos y aplicaciones de la computación orientada al servicio y elaborar un mapa conceptual. • Realizar una investigación acerca de las arquitecturas orientadas al servicio. • Identificar los modelos de servicios a través de una investigación y elaborar una tabla comparativa que incluya su definición, características, ventajas y desventajas. • Elaborar una presentación electrónica acerca de la Integración de Aplicaciones e Ingeniería de Software Orientada al Servicio, y exponerla en el grupo. • Investigar acerca de los formatos de representación de recursos: XML, JSON, Atom y elaborar una tabla comparativa que incluya su definición, características y requerimientos.



2. Arquitectura de servicios SOAP	
Competencias	Actividades de aprendizaje
<ul style="list-style-type: none"> • Específica(s): • Adquiere la capacidad del desarrollo de servicios web con arquitectura SOAP. • Genéricas: • Capacidad de abstracción, análisis y síntesis. • Comunicación oral y escrita. • Habilidad para buscar, procesar y analizar información procedente de fuentes diversas. • Capacidad de trabajo en equipo. • Compromiso ético. • Capacidad de aplicar los conocimientos en la práctica. • Habilidad para trabajar en forma autónoma. 	<ul style="list-style-type: none"> • Elaborar un mapa conceptual acerca de la arquitectura de servicios SOA, considerando conceptos básicos, características, usos y aplicaciones. • Investigar acerca de los formatos de mensajes y protocolo de mensajería: SOAP, y elaborar una tabla comparativa que incluya su definición, características y requerimientos. • Elaborar una presentación electrónica acerca de la descripción, publicación, descubrimiento e integración: UDDI, invocación de servicios, gestión del estado del servicio, soporte de una plataforma para el desarrollo de Servicios Web SOAP, y exponerla en el grupo. • Seleccionar ejemplos donde se apliquen servicio Web basado en arquitectura SOAP y exponerlo haciendo uso de las herramientas computacionales disponibles. • Resolver ejercicios de servicio Web basado en arquitectura SOAP según lo requiera el caso y elaborar documentación de resultados.
3. Arquitectura de servicios REST	
Competencias	Actividades de aprendizaje
<p><i>Específica(s):</i> Adquiere la capacidad del desarrollo de servicios web basados en la arquitectura REST.</p> <p><i>Genérica(s):</i></p> <ul style="list-style-type: none"> • Capacidad de abstracción, análisis y síntesis. • Comunicación oral y escrita. • Habilidad para buscar, procesar y analizar información procedente de fuentes diversas. • Capacidad de trabajo en equipo. • Compromiso ético. • Capacidad de aplicar los conocimientos en la práctica. • Habilidad para trabajar en forma autónoma. 	<ul style="list-style-type: none"> • Elaborar un cuadro sinóptico acerca de estilo arquitectónico REST. • Realizar un reporte de investigación acerca de la Arquitectura Orientada a Recursos (ROA) y del modelo de recursos. • Elaborar una presentación electrónica acerca del diseño de URIs y Clientes REST, diseño e implementación de Servicios RESTful orientados a recursos, y exponerla en el grupo. • Investigar acerca de mashups de servicios y del soporte de una plataforma para el desarrollo de Servicios Web RESTful, y elaborar un mapa conceptual. • Identificar los marcos de trabajo a través de una investigación y elaborar una tabla comparativa que incluya su definición, características, ventajas y desventajas.



	<ul style="list-style-type: none"> • Seleccionar ejemplos donde se apliquen servicio Web basado en arquitectura REST y exponerlo haciendo uso de las herramientas computacionales disponibles. • Resolver ejercicios de servicio Web basado en arquitectura REST según lo requiera el caso y elaborar documentación de resultados.
4. Microservicios	
Competencias	Actividades de aprendizaje
<p><i>Específica(s):</i> Adquiere la capacidad del desarrollo de la arquitectura basada en microservicios.</p> <p><i>Genéricas:</i></p> <ul style="list-style-type: none"> • Capacidad de abstracción, análisis y síntesis. • Comunicación oral y escrita. • Habilidad para buscar, procesar y analizar información procedente de fuentes diversas. • Capacidad de trabajo en equipo. • Compromiso ético. • Capacidad de aplicar los conocimientos en la práctica. • Habilidad para trabajar en forma autónoma. 	<ul style="list-style-type: none"> • Elaborar un mapa conceptual acerca de microservicios, considerando conceptos básicos, características, usos y aplicaciones. • Elaborar una presentación electrónica acerca de principios arquitectónicos de los microservicios: enfoques de modularización, ventajas y desventajas, técnicas de modularización, equilibrio entre redundancia e integridad, orquestación y coreografía; y exponerla en el grupo. • Realizar un reporte de investigación acerca de orquestación de servicios, integración de servicios usando coreografía, registro y auto-reconocimiento de servicios. • Elaborar una presentación electrónica acerca de Balanceo de carga en client, Patrón Circuit Breaker (Corto circuito) y Contenedores; y exponerla en el grupo. • Identificar los marcos de trabajo a través de una investigación y elaborar una tabla comparativa que incluya su definición, características, ventajas y desventajas. • Elaborar una tabla comparativa acerca de SOA y microservicios, que incluya su definición, características, ventajas y desventajas. • Seleccionar ejemplos de servicio Web donde se aplique la arquitectura de microservicios y exponerlo haciendo uso de las herramientas computacionales disponibles. • Resolver ejercicios de servicio Web basado en la arquitectura de microservicios según lo requiera el caso y elaborar documentación de resultados.

5. Tópicos avanzados de desarrollo

Competencias	Actividades de aprendizaje
<p>Específica(s):</p> <p>Elabora los artefactos finales para la entrega y despliegue continuo del software.</p> <p>Genéricas:</p> <ul style="list-style-type: none">• Capacidad de abstracción, análisis y síntesis.• Comunicación oral y escrita.• Habilidad para buscar, procesar y analizar información procedente de fuentes diversas.• Capacidad de trabajo en equipo.• Compromiso ético.• Capacidad de aplicar los conocimientos en la práctica.• Habilidad para trabajar en forma autónoma.	<ul style="list-style-type: none">• Escenifica en equipo y en GitHub, mediante una práctica en un lenguaje de programación los ambientes de desarrollo, testeo, preproducción y producción de software.• Analiza un caso de estudio donde se aborda el tema de integración y despliegue continuo de software.• Realiza práctica de laboratorio de Git y GitHub para integración y despliegue de software.• Elabora un documento con la configuración de despliegue de una aplicación.• Investiga y discute ante el grupo el proceso para patentar software, registro de marca y registro IN-DAUTOR.• Elabora cuadro comparativo de los tipos de licenciamiento que existen.• Registra una marca, un manual o un software ante instancias legales.

8. Práctica(s)

<ul style="list-style-type: none">• Construcción de un servicio Web para validar el acceso a una aplicación.• Construcción de un servicio Web para reportar el clima de un estado.• Construcción de un servicio Web para reporte de cotizaciones en la Bolsa Mexicana de Valores (BMV).• Construcción de un servicio Web para validar tarjetas de crédito.• Construcción de un servicio Web de una certificación digital.• Construcción de un microservicio para reserva de habitaciones de un hotel.• Elabora la documentación final para la entrega y despliegue del software, entre ellos, manual de usuario, manual técnico, configuración de despliegue, presentación ejecutiva y logotipo del software.• Realiza un programa en colaboración con otros compañeros utilizando integración continua a través de la herramienta Git y la plataforma github.com, donde implementa las ramas de desarrollo, testeo, preproducción y producción.• Realiza un programa en colaboración con otros compañeros utilizando despliegue continuo a través de la herramienta Git y la plataforma github.com, donde implementa las ramas de desarrollo, testeo, preproducción y producción.• Registra con sus compañeros un software, manual o marca ante un organismo legal o mediante un licenciamiento Open Source.
--



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

La evaluación debe ser permanente y continua. Se debe hacer una evaluación diagnóstica, formativa y sumativa. Se debe aplicar autoevaluación, coevaluación y heteroevaluación.

Algunos de los instrumentos que se pueden utilizar:

- Mapa conceptual.
- Mapa mental.
- Guía de problemas de programación.
- Examen (teóricos y prácticos).
- Reportes de prácticas.
- Resúmenes.
- Preguntas guiadas.
- Plenaria.
- Cuadro comparativo.



Herramientas:

- Rúbrica.
- Lista de cotejo.
- Matriz de valoración.
- Guía de observación.

Se debe generar un portafolio de evidencias, de preferencia en formato digital.

11. Fuentes de información

1. Building Web Services with Java: Making Sense of XML, SOAP, WSDL and UDDI. Steve Graham, Simeon Simeonov, Toufic Boubez, Doug Davis, Glen Daniels, Yuichi Nakamura, Ryo Neyama. Sams Publishing.
2. MBuilding Web Services with Java: Making Sense of XML, SOAP, WSDL and UDDI. Steve Graham, Simeon Simeonov, Toufic Boubez, Doug Davis, Glen Daniels, Yuichi Nakamura, Ryo Neyama. Sams Publishing.
3. Microsoft .NET XML Web Services Step by Step: Adam Freeman y Allen Jones. Microsoft Press.
4. Developing Java Web Services: Architecting and Developing Secure Web Services Using Java. Ramesh Nagappan, Robert Skoczylas, Rima Patel Sriganesh. John Wiley & Sons. 2003.
5. Tabor Robert. Servicios Web XML de Microsoft.NET. Pearson.
6. Castro. XML Guía de Aprendizaje. Pearson.
7. Subbu Allamaraju. RESTful Web Services Cookbook, O'REILLY, 2010.
8. Leonard Richardson, Sam Ruby. RESTful Web Services, O'REILLY, 2007.
9. Nicolai M. Josuttis, SOA in Practice: The Art of Distributed System Design, O'REILLY, 2007
10. Lonneke Dikmans, Ronald Van Luttikhuisen. SOA Made Simple, Packt Publishing, 2012.
11. Arnon Rotem-Gal-Oz. SOA Patterns, Manning, 2012.
12. Surianarayanan, C., Ganapathy, G., & Pethuru, R. (2019). Essentials of Microservices Architecture: Paradigms, Applications, and Techniques.
13. Bucchiarone, A., Dragoni, N., Dustdar, S., Lago, P., Mazzara, M., Rivera, V., & Sadovykh, A. (2020). Microservices. Science and Engineering. Springer.
14. Wolff, E. (2016). Microservices: flexible software architecture. Addison-Wesley Professional.
15. Larsson, M. (2019). Hands-on Microservices with spring boot and spring cloud: Build and deploy Java microservices using spring cloud, Istio, and Kubernetes. Packt Publishing Ltd.
16. Richardson, C. (2018). Microservices patterns: with examples in Java. Simon and Schuster.
17. Mercado-Ramos, V. H., Zapata, J., & Ceballos, Y. F. (2015). Herramientas y buenas prácticas para el aseguramiento de calidad de software con metodologías ágiles. Revista de Investigación, Desarrollo e Innovación, 6(1), 73-83.
18. Aguirre, L. G., & Palomino, N. L. S. (2015). Desarrollo de un marco metodológico del proceso de verificación y validación de software para pequeñas y medianas empresas. Industrial data, 18(2), 145-154.
19. Pasano, Michael; Software testing series vol. 2, Independently Published, 2020

Sitios web de referencia y consulta:

20. RESTful Web Service Developer's Guide. ORACLE. <http://docs.oracle.com /cd/E19776-01/820-4867/index.html>
21. Tutorial sobre JSON ORACLE. <http://www.oracle.com/technetwork/es/articles /java/api-java-para-json-2251318-esa.html>



22. <http://distance-ed.bcc.ctc.edu/media238/238p2/IntroDBconn.html>
23. http://www.webtaller.com/manual-xml/indice_manual_xml.php
24. <http://www.desarrolloweb.com/articulos/1535.php>
25. <https://f.hubspotusercontent30.net/hubfs/462636/68.pdf>