



Desarrollo de Software

PROGRAMA DE LA UNIDAD DIDÁCTICA:
Diseño y arquitectura de software

Clave:

TSU
16142525

LICENCIATURA
15142424

México, Ciudad de México, marzo del 2024

Universidad Abierta y a Distancia de México





ÍNDICE

Datos de identificación	3
Presentación	3
PROBLEMA PROTOTÍPICO	5
CONOCIMIENTOS PREVIOS	6
COMPETENCIA GENERAL	6
COMPETENCIAS ESPECÍFICAS	7
COMPETENCIAS TRANSVERSALES	8
LOGROS	9
RELACIÓN CON EL PERFIL DE EGRESO	10
RELACIÓN CON OTRAS UNIDAD DIDÁCTICAS	10
METODOLOGÍA	13
TEMARIO	14
EVALUACIÓN	16
Fuentes de consulta	18



Datos de identificación

Nombre de la carrera:	Desarrollo de Software
Nombre de la unidad didáctica	Diseño y arquitectura de software
Clave de unidad didáctica:	15142424 / 16142525
Seriación:	No aplica
Semestre	Cuarto
Horas contempladas:	72

Presentación

Para construir una aplicación práctica es necesario conocer los requerimientos funcionales y no funcionales del usuario, y seleccionar la mejor opción de arquitectura de software para modelar los elementos. Por ello, con el estudio de la unidad didáctica, se observará que la arquitectura de software es una etapa fundamental dentro del ciclo de vida del desarrollo de software, ya que a partir de la decisión de qué arquitectura aplicar, el éxito de una plataforma de software se verá reflejado en la agilidad de ésta para adaptarse a los cambios naturales que se dan en cualquier organización en cuanto a sus reglas de negocio, la compatibilidad con tecnologías anteriores, contemporáneas y futuras, así como en la facilidad de uso para el público usuario y la cantidad de esfuerzo/inversión necesarios para el mantenimiento de dicha plataforma.

En la unidad didáctica *Diseño y arquitectura de software* se abordarán conceptos dirigidos hacia la comprensión de qué es, cómo, cuándo y qué tipo de arquitectura es la adecuada respecto a la necesidad específica sobre la base de los modelos y patrones de diseño de software. Asimismo, observarás lo que abarca la conceptualización y descripción gráfica de los componentes de un sistema de software y la relación entre ellos, por lo cual su desarrollo conlleva el diseño de la estructura que soportará la solución de software.



Esta unidad didáctica está relacionada con la representación y composición de los módulos de un sistema de software, por lo que el diseño se propone como una actividad conciliatoria entre los requerimientos del problema y la factibilidad técnica de una solución en términos de un sistema de software. Al final se tendrá una visión amplia, completa y humana del mismo.

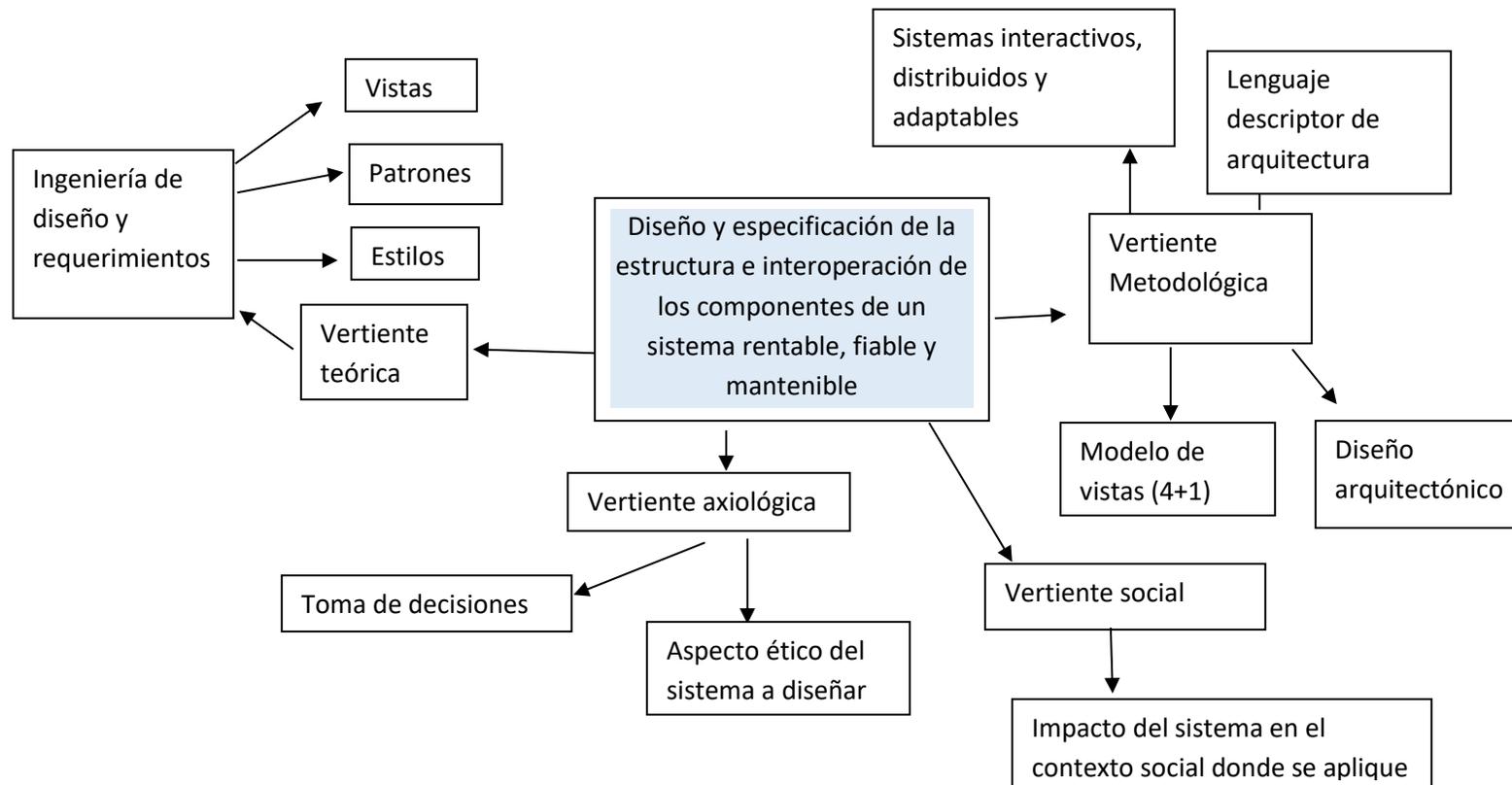
Esta unidad didáctica se encuentra conformada por tres unidades:

1. Arquitectura de software
2. Elementos de diseño de la arquitectura de software
3. Arquitectura de sistemas informáticos

En la primera unidad se analizarán los elementos de la arquitectura de software y su importancia en el diseño del mismo para, posteriormente, aplicar patrones de arquitectura y diseñar vistas mediante el lenguaje descriptor de arquitectura. En la segunda y tercera unidad se diseñará una propuesta de arquitectura para el diagnóstico de información de los usuarios mediante el análisis y uso de herramientas de diseño de la arquitectura en diferentes tipos de sistema.



PROBLEMA PROTOTÍPICO





CONOCIMIENTOS PREVIOS

Con la finalidad de alcanzar las competencias establecidas para la unidad didáctica es necesario contar con las siguientes habilidades y conocimientos:

- Conocimiento y aplicación de las diversas metodologías asociadas a los ciclos de vida del software.
- Capacidad de análisis para la determinación y documentación de los requerimientos funcionales y no funcionales de un sistema, derivados de la etapa de análisis y que satisfagan las necesidades del usuario final.
- Conocimiento de tecnologías informáticas. Por ejemplo, procesadores de texto y Visual Studio, plataforma que permite diseñar la arquitectura de la aplicación en modelos UML. También es necesario conocer los mecanismos de descarga de programas y contenidos e instalación de software para el desarrollo y documentación de la arquitectura de software de un sistema.
- Capacidad de investigación, toma de decisiones, participación en proyectos de desarrollo de software.

COMPETENCIA GENERAL

- Construye una arquitectura de software para establecer las bases del desarrollo de un sistema que cubra las necesidades del usuario mediante el uso de patrones de arquitectura de software.



COMPETENCIAS ESPECÍFICAS

- Analiza las herramientas de arquitectura de software para utilizarlas en el proceso de diseño de una aplicación, propiciando la implementación de vistas, UML y lenguajes descriptores de arquitectura.
- Aplica un estilo de diseño para generar una propuesta de modelo arquitectónico que permita el diagnóstico de información de los usuarios mediante el análisis y uso de herramientas de diferentes tipos y la descripción del comportamiento del sistema planteado en respuesta a un caso concreto.
- Diseña un modelo arquitectónico que aplique e identifique los componentes de los sistemas distribuidos, interactivos y adaptables para el desarrollo de una aplicación informática dirigida a un objetivo específico mediante un patrón de arquitectura de software.





COMPETENCIAS TRANSVERSALES

Comunicación	Gestión de información	Pensamiento crítico	Trabajo colaborativo	Sociales	Solución de problemas y toma de decisiones
<ul style="list-style-type: none">• Capacidad de comunicación oral y escrita.• Capacidad de comunicación en un segundo idioma.	<ul style="list-style-type: none">• Capacidad de investigación.• Capacidad de aprender y actualizarse permanentemente.• Habilidades para buscar, procesar y analizar información procedente de diversas fuentes.	<ul style="list-style-type: none">• Capacidad de actuar ante nuevas situaciones.• Capacidad crítica y autocrítica.• Capacidad de abstracción, análisis y síntesis.	<ul style="list-style-type: none">• Capacidad de trabajar en equipo.• Habilidades interpersonales.• Capacidad de motivar y conducir hacia metas comunes.• Capacidad para formular y gestionar proyectos.	<ul style="list-style-type: none">• Responsabilidad social y compromiso ciudadano.• Compromiso con la preservación del medio ambiente.• Compromiso con su medio socio-cultural.• Valoración y respeto por la diversidad y la multiculturalidad.• Compromiso ético.• Compromiso con la calidad.	<ul style="list-style-type: none">• Capacidad creativa.• Capacidad para tomar decisiones.• Capacidad para identificar, plantear y resolver problemas.• Capacidad de organizar y planificar el tiempo.• Capacidad de aplicar los conocimientos en la práctica.



LOGROS

En la unidad didáctica *Diseño y arquitectura de software* se desarrollarán, por unidad temática, los siguientes contenidos:

Primera unidad

- Definición, enfoque e importancia de la arquitectura de software
- Vistas de la arquitectura de software
- Patrones de diseño
- Ubicación de la arquitectura de software en el proceso de desarrollo

Tiempo estimado para su desarrollo 22 horas

Segunda unidad

- Estilos y patrones de arquitectura
- Capas de modelos arquitectónicos
- Tuberías y filtros
- Tableros

Tiempo estimado para su desarrollo 22 horas

Tercera unidad

- Sistemas distribuidos: características, ventajas y desventajas
- Sistemas interactivos: MVC y PAC
- Sistemas adaptables: Proxy y administración de los sistemas

Tiempo estimado para su desarrollo 24 horas

Se identificarán las metodologías, modelos y patrones existentes aplicables al diseño para la generación de una arquitectura de software que provea las bases para el desarrollo de un sistema informático.



Se analizarán las ventajas y las desventajas para la aplicación de cada uno de estos modelos y patrones existentes con base en la naturaleza propia de la problemática a resolver.

Se definirán las estructuras que componen una arquitectura de software y su documentación apropiada para su comunicación a los demás miembros involucrados en el desarrollo de la aplicación informática.

Se utilizarán programas, contenidos e instalación de software para el desarrollo y documentación de la arquitectura de software de un sistema; entre ellos, ArchStudio bajo ambiente Windows y plataforma Eclipse.

RELACIÓN CON EL PERFIL DE EGRESO

La unidad didáctica *Diseño y arquitectura de software* aporta al perfil de egreso de los estudiantes de Ingeniería de Software y de Técnico Superior Universitario en Desarrollo de Software los conocimientos y las habilidades necesarias para la correcta definición de una arquitectura de software mediante la aplicación de estándares, patrones existentes y lenguajes descriptores de arquitectura con base en los requerimientos funcionales y no funcionales establecidos en la fase de análisis del proyecto en coordinación con el usuario, lo que permite el establecimiento de las bases de diseño para la materialización y puesta en marcha de una solución computacional.

RELACIÓN CON OTRAS UNIDAD DIDÁCTICAS

Para el desarrollo de la siguiente unidad didáctica es requisito indispensable haber cursado y acreditado las siguientes unidad didácticas que forman parte del plan de estudios de la carrera Desarrollo de Software:

Fundamentos de programación



Mediante el diseño de algoritmos, el estudiante analiza los problemas planteados y define los pasos que permiten su resolución, para codificarlos posteriormente en cualquier lenguaje de programación, atendiendo la sintaxis, reglas y herramientas propias de cada uno. Estas bases permitirán definir la mejor manera de resolver los requerimientos presentados por el usuario, resultado de la fase de análisis y traducidos en una arquitectura de software.

Habilidades del pensamiento

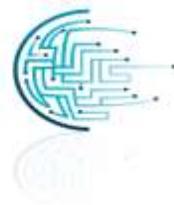
Facilita el desarrollo de competencias del estudiante, las cuales le permiten tener un pensamiento estructurado que repercute en una toma de decisiones, lo que simplifica la comprensión de todas las demás unidades didácticas del programa académico y su integración como competencias en el ejercicio de su profesión. Las habilidades mencionadas constituyen una base primordial para el desarrollo de cualquier aplicación informática, ya que desde la presentación del problema es necesario determinar los requerimientos necesarios tanto para el diseño (arquitectura de software) como para la construcción del sistema que satisfaga las necesidades del usuario, de manera confiable y segura.

Bases de datos

Las aplicaciones informáticas tienen como fin principal el procesamiento de la información y la presentación de resultados que permitan una adecuada toma de decisiones. La unidad didáctica *Bases de datos* establece los fundamentos para la realización de prototipos documentales que sirven de modelo para dar solución a problemas relacionados con la gestión de la información. En el diseño de la arquitectura de software es necesario definir cómo será integrada la información que maneja el sistema en el diseño final, constituyendo un componente imprescindible del mismo.

Análisis y diseño orientado a objetos

Proporciona las herramientas para el diseño de un programa de computadora bajo el enfoque orientado a objetos, paradigma que se utiliza en los lenguajes modernos de



programación, por lo que el diseño de la arquitectura de software no debe perder de vista este enfoque que permitirá aprovechar los beneficios para un desarrollo más sencillo, eficiente y encaminado a la obtención de mejores resultados.

Programación orientada a objetos

Similar al punto descrito anteriormente, esta unidad didáctica permite el desarrollo de habilidades para el diseño y, en este caso específico, la implementación de aplicaciones de software en un lenguaje orientado a objetos (Java) que satisfaga las necesidades de procesamiento de la información, objetivo primordial de todo sistema. Estos conceptos no se deben perder de vista al momento de elegir los patrones para el diseño de la arquitectura de software.

Introducción a la ingeniería de software

Considera los principios, conceptos y métodos que permiten observar el proceso de desarrollo de software. Se abordan las fases comunes de los ciclos de vida de software: análisis, diseño, codificación, pruebas y mantenimiento, así como los paradigmas de desarrollo. Es necesario identificar debidamente cada etapa del ciclo de vida antes mencionado para ubicar el proceso en el cual se encuentra inmerso el diseño de la arquitectura del software a construir, así como su importancia.

Métodos y modelos de desarrollo de software

La unidad didáctica *Diseño y arquitectura de software* tiene relación con esta unidad didáctica al dar a conocer ampliamente el proceso de desarrollo de software y las tecnologías que se utilizan para su modelado, estableciendo de esta manera un diseño que permita mejorar la estructura de la organización.

Modelado de negocios

La unidad didáctica hace énfasis en la correcta definición de requerimientos por parte del ingeniero de software para determinar los procesos (implícitos y explícitos) de un negocio



y comenzar con el desarrollo de la aplicación informática. Incorpora el Lenguaje Unificado de Modelado UML y Notación para el Modelado de Procesos de Negocio BPMN con el fin de representar las funciones de la organización a través de los diagramas. El dominio de estos conceptos, mecanismos y habilidades permitirán una adecuada determinación de requerimientos que constituyen la base para el diseño y construcción de la arquitectura de software.

Las competencias adquiridas en cada una de las unidades didácticas antes mencionadas permitirán elegir la mejor solución acorde con los requerimientos funcionales y no funcionales de un problema determinado, transcribiendo esta solución en una arquitectura de software mediante la identificación de los patrones de diseño disponibles.

METODOLOGÍA

La evaluación de la unidad didáctica en el marco del programa de la UnADM se conceptualiza como un proceso participativo, sistemático y ordenado, en el que se identifica un problema para resolver mediante las situaciones de aprendizaje de las unidades temáticas y obtener el resultado esperado: la arquitectura de software.

La metodología se denomina Aprendizaje basado en la resolución de problemas; es decir, el estudiante parte del planteamiento de un problema susceptible de ser resuelto mediante el desarrollo de un sistema de software, desarrolla las competencias esperadas a partir de la identificación y la correcta aplicación de los diversos patrones, y genera la arquitectura de software para resolver de forma integral los requerimientos planteados.

El procedimiento a seguir en la metodología seleccionada es el siguiente:

- Presentación de un problema: Definición del mismo con la posibilidad de integrar uno por unidad temática, estableciendo las condiciones de trabajo con base en las indicaciones de la figura académica.



- Identificación de las necesidades: Determinación del patrón de arquitectura de software aplicable para su resolución con base en los requerimientos derivados del problema planteado,
- Obtención de la información: Investigación y recopilación en fuentes confiables la información adicional requerida para el correcto desarrollo de la arquitectura de software.
- Resolución del problema: Presentación de la propuesta de arquitectura de software.

Además, se utilizarán recursos de organización de información y análisis de conceptos y se promoverá el trabajo colaborativo en los foros de discusión para favorecer la comunicación e integración entre estudiantes.

TEMARIO

Unidad 1. Arquitectura de software

1.1. Introducción a la arquitectura de software

1.1.1. Origen y objetivo de la arquitectura de software

1.1.2. Lenguajes descriptores

1.1.3. Vistas de la arquitectura

1.1.4. Modelo vistas 4+1

1.2. La arquitectura en la fase de diseño de software

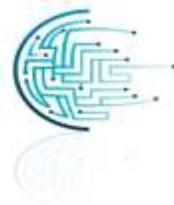
1.2.1. La metodología rup

1.2.2. Propiedades y principios del diseño de software

1.2.3. Proceso de diseño de software

Unidad 2. Elementos de diseño de la arquitectura de software

2.1. Estilos y patrones arquitectónicos



2.1.2 Enfoque

2.2. Categorías de estilos arquitectónicos

2.2.1 Basado en componentes

2.2.2 Capas

2.2.3 Cliente/servidor

2.2.4 Orientado a objetos

2.2.5 Tuberías y filtros

2.3 Patrones arquitectónicos emergentes

2.3.1 Patrón de arquitectura de dominio

2.3.1 Patrón de arquitectura de cebolla

2.3.1 Patrón de arquitectura de fachada

Unidad 3. Arquitectura de sistemas informáticos

3.1. Patrones arquitectónicos en sistemas distribuidos

3.1.1. Ámbito de aplicación

3.1.2. Arquitectura orientada a servicios

3.1.3 Patrón de arquitectura de nube

3.2. Patrones arquitectónicos en sistemas interactivos

3.2.1. Ámbito de aplicación

3.2.2. Modelo-vista-controlador

3.2.3. Presentación-abstracción-control

3.3. Patrones arquitectónicos en sistemas adaptables

3.3.1. Ámbito de aplicación

3.3.2. Proxy y sistemas adaptables



3.3.3. Microkernel

3.3.4. Reflection

EVALUACIÓN

La evaluación del aprendizaje es un proceso, a través del cual se observa, recoge y analiza información relevante del proceso de aprendizaje de los estudiantes, con la finalidad de reflexionar, emitir juicios de valor, así como tomar decisiones pertinentes y oportunas para optimizarlo (Díaz Barriga A.F. & Hernández R.G., 2005). Orienta la toma de decisiones, da pauta a determinar acciones en términos de valoración de conocimientos, nivel del desempeño, reorientaciones de aprendizaje, mejora del proceso educativo y adecuación de actividades, entre otras acciones.

De acuerdo con lo anterior, mediante la evaluación te brindaremos apoyo y seguimiento para identificar las dificultades en el desarrollo de conocimientos, habilidades y actitudes del proceso integral de aprendizaje.

En el marco del Modelo educativo de la UnADM, la evaluación de la unidad didáctica se realiza en los siguientes momentos: 1) formativa y 2) sumativa.

EVALUACIÓN FORMATIVA

Se realiza en paralelo al desarrollo del proceso de enseñanza-aprendizaje de cada unidad, y sirve para localizar dificultades cuando aún estás en posibilidad de remediarlas.

En este primer momento de evaluación, se aplican estrategias asociadas a las:

- **Actividades individuales** (tareas). Se trata de un primer momento de aprendizaje, en el cual se consideran tus perspectivas, experiencias, intereses, capacidades y necesidades.
- **Actividades colaborativas** (foros). El trabajo colaborativo fomenta y promueve el aprendizaje en contribución con otros compañeros, ya que eres responsable no sólo de tu aprendizaje, sino de contribuir a que los demás aprendan en equipo y se fomente un ambiente de confianza; por ende, que se logren las metas de aprendizaje.



EVALUACIÓN SUMATIVA

Se aplica al final del proceso de tu experiencia de aprendizaje, su propósito es verificar los resultados alcanzados y el grado de aprendizaje o nivel de conocimientos, habilidades y actitudes que hayas adquirido.

Este segundo y último momento de evaluación, se mide y valora a través de las siguientes actividades:

- **Evidencias de aprendizaje.** Son actividades que tienen como objetivo integrar el proceso de construcción de tu aprendizaje, la evaluación, la retroalimentación y la planeación de la nueva ruta de aprendizaje que seguirás de acuerdo con los resultados individuales obtenidos.
- **Actividad complementaria.** Esta actividad es planeada por la figura académica considerando las competencias y logros de la unidad didáctica, toda vez que identifica los conocimientos, habilidades y actitudes que te hizo falta desarrollar o potenciar (se realiza en una ocasión al finalizar la última unidad).
- **Actividad de reflexión.** Es un ejercicio de metacognición que permite que tomes conciencia de tu proceso de aprendizaje, el punto de partida son las experiencias del contexto académico y la reflexión sobre tu desempeño. Se trata de una acción formativa que parte de tu persona y no del saber teórico, que considera tu experiencia de aprendizaje (se realiza en una ocasión al finalizar la última unidad).

A continuación, se presenta el esquema general de evaluación correspondiente a esta unidad didáctica:

Esquema general de evaluación

Tipo de evaluación	Actividades	Puntaje
--------------------	-------------	---------



Formativa	Actividades individuales	30%
	Actividades colaborativas	10%
Sumativa	Evidencias de aprendizaje	40%
	Actividad complementaria	10%
	Actividad de reflexión	10%
Total		100%

Recuerda que la calificación final que te permitirá acreditar se asigna de acuerdo con los criterios e instrumentos de evaluación establecidos para cada actividad, los cuales son diseñados con base en las competencias y logros de esta unidad didáctica.

Fuentes de consulta

Bibliografía básica

- Bass, L., Clemens, P., & Kazman, R. (2003). *Software Architecture in Practice* (2ª ed.). Addison Wesley.
- De la Torre, C., Zorrilla, U., Calvarro, J., & Ramos, M. A. (2010). *Guía de arquitectura n-capas orientada al dominio con .NET 4.0*. Krasis Press.
- Buschmann, F., Meunier, R., Rohnert, H., Sommerlad, P., & Stal, M. (1996). *Pattern-Oriented Software Architecture*. Wiley.
- Kruchten, P. (1995). *Architectural blueprints – The “4+1” view model of software architecture*. *IEEE Software*, 12(6), 42-50.
- Qin, Z., Xing, J., & Zheng, X. (2008). *Software Architecture*. Zhejiang University Press, Hangzhou, Springer – Verlag GmbH Berlin Heidelberg.
- Reynoso, C., & Kicillo, N. (2004). *Estilos y patrones en la estrategia de arquitectura de Microsoft*. Universidad de Buenos Aires.



- Reynoso, C., & Kicillof, N. (2004). *Lenguajes de descripción de arquitectura*. Universidad de Buenos Aires.
- Sommerville, I. (2005). *Ingeniería del software* (7ª ed.). Pearson Education.
- Tanenbaum, A. (2006). *Distributed Systems: Principles and Paradigms* (2ª ed.). Prentice Hall.
- Taylor, R. N., Medvidovic, N., & Dashofy. (2009). *Software Architecture: Foundations, Theory, and Practice*. John Wiley.

Bibliografía complementaria

- C. A. B., Barai, M., & Caselli, V. (2008). *Service Oriented Architecture with Java*. Packt Publishing.
- Lauriat, S. M. (2007). *Advanced Ajax: Architecture and Best Practices*. Pearson Education.
- Marino, J., & Rowley, M. (2010). *Understanding SCA (Service Component Architecture)*. Pearson Education.
- Reese, G. (2009). *Cloud Application Architectures*. O'Reilly.
- Castro, L., Cervantes, H., & Velasco, P. (2015). *Arquitectura del Software. Conceptos básicos*. Editores SA de CV.
- Durango, A., & Arias, Á. (2014). *Ingeniería y Arquitectura de Software*. Createspace.
- Khan, S. M. A. (2023). *Popular software architecture used in software development*. ResearchGate. [Online]. Available: https://www.researchgate.net/publication/370715656_Popular_Software_Architecture_Used_In_Software_Development
- Kruchten, P. (1995). *Architectural Blueprints – The “4+1” View Model of software architecture*. IEEE Software, 12(6), 42-50. [Online]. Available: <https://www.cs.ubc.ca/~gregor/teaching/papers/4+1view-architecture.pdf>
- Martel, Y., Roßmann, A., Sultanow, E., Weiß, O., Wissel, M., Pelzel, F., & Sessler, M. (2020). *Software Architecture Best Practices for Enterprise*



Artificial Intelligence. GI-Jahrestagung, 165-181.

https://doi.org/10.18420/inf2020_16

- Pressman, R. (2010). *Software engineering: a practitioner's approach* (7^a ed.). Mc Graw Hill Higher Education.
- Reynoso, C., & Kicillof, N. (2004). *Lenguajes de descripción de arquitectura*. Universidad de Buenos Aires.
- Sommerville, I. (2005). *Ingeniería del Software* (7^a ed.). Pearson Education.
- Torvekar, N., & Game, P. (2019). *Microservices and its Applications: An overview*. International journal of computer sciences and engineering, 7(4), 803-809. <https://doi.org/10.26438/ijcse/v7i4.803809>
- Bass, L., Clements, P., & Kazman, R. (2012). *Software Architecture in Practice*. Addison-Wesley Professional.
- De la Torre, C., Zorrilla, U., Calvarro, J., & Ramos, M. A. (2010). *Guía de arquitectura N-Capas orientada al dominio con .NET 4.0*. Krasis Press.
- Evans, E. (2015). What is Domain-Driven Design?. https://www.dddcommunity.org/learning-ddd/what_is_ddd/
- Khalil, M., Ghani, K., & Khalil, W. (2016). *Onion Architecture: a new approach for XAAS (everything-as-a-service) based virtual collaborations*.