



**UNIVERSIDAD NACIONAL MAYOR DE SAN MARCOS**

Universidad del Perú. Decana de América  
Facultad de Ingeniería de Sistemas e Informática  
Vicedecanato de Investigación y Posgrado  
Unidad de Posgrado

**MAESTRÍA EN INGENIERÍA DE SISTEMAS E INFORMÁTICA CON  
MENCIÓN EN INGENIERÍA DE SOFTWARE**

***SÍLABO***

Nombre de la asignatura : Arquitectura de Software

Profesor responsable : Mg. Ing. Efrain Ricardo Bautista Ubillús

Correo electrónico : ebautistau@unmsm.edu.pe

2024-2



## UNIVERSIDAD NACIONAL MAYOR DE SAN MARCOS

Universidad del Perú. Decana de América  
Facultad de Ingeniería de Sistemas e Informática  
Vicedecanato de Investigación y Posgrado  
Unidad de Posgrado

### 1. INFORMACIÓN GENERAL

- 1.1. Nombre de la asignatura : Arquitectura de Software
- 1.2. Tipo de asignatura : Obligatoria
- 1.3. Profesor(a) : Mg. Ing. Efrain Ricardo Bautista Ubillús
- 1.4. Programa : Maestría en Ingeniería de Sistemas e Informática con  
mención en Ingeniería de Software
- 1.5. Código de asignatura : R3S2A112
- 1.6. Créditos : 4
- 1.7. N° de horas semanales : 3
- 1.8. N° de horas por semestre : 48
- 1.9. Semestre académico : 2024-2
- 1.10. Duración : 16 semanas
- 1.11. Fecha de inicio : 14/09/2024
- 1.12. Fecha de finalización : 28/12/2024
- 1.13. Horario : Sábado 08:00 AM - 11:00 AM

### 2. FUNDAMENTOS DE LA ASIGNATURA

#### 2.1. Sumilla.

Asignatura que corresponde al periodo de profundización, es de naturaleza teórica-práctica de modalidad presencial. Tiene como propósito diseñar, implementar y evaluar las arquitecturas de software. Conocer soluciones arquitectónicas relacionadas con diferentes tipos de problemas tecnológicos. Entender los conceptos relacionados con el desarrollo de soluciones de software en la nube, los casos de aplicación, las problemáticas habituales que enfrentan las organizaciones para iniciar con proyectos basados en la nube, y el diseño de soluciones de software híbridas entre modelos tradicionales y de nube. Al finalizar el estudiante deberá presentar un informe aplicando los tópicos tratados.

#### 2.2. Competencias del programa.

El curso contribuye al logro de competencias, tales como:

- 2.2.1. Aplica el pensamiento crítico y creativo en su desempeño profesional permitiendo solucionar los problemas complejos que se presentan en las organizaciones y la sociedad adoptando la mejor alternativa.
- 2.2.2. Diseña y desarrolla software de calidad.
- 2.2.3. Conoce y trabaja con las herramientas, plataformas de desarrollo de software y entornos integrados de desarrollo (IDE) con las que se trabaja actualmente en la industria.
- 2.2.4. Identifica y selecciona innovaciones tecnológicas, manteniendo la competitividad de los recursos informáticos en su organización.
- 2.2.5. Desarrolla aplicaciones en la nube y despliega o migra aplicaciones o servicios en la nube.
- 2.2.6. Desarrolla investigaciones científico tecnológicas en una de las líneas de investigación del programa, orientadas al cumplimiento de los objetivos de desarrollo sostenible (ODS), a través de actividades innovadoras y creativas.

#### 2.3. Competencias de la asignatura.

- 2.3.1. Comprende los conceptos básicos de arquitectura de software y sus elementos.



## UNIVERSIDAD NACIONAL MAYOR DE SAN MARCOS

Universidad del Perú. Decana de América  
Facultad de Ingeniería de Sistemas e Informática  
Vicedecanato de Investigación y Posgrado  
Unidad de Posgrado

- 2.3.2. Analiza requisitos arquitecturales de software, aplica patrones y tácticas para dar solución a problemas concretos.
- 2.3.3. Aplica adecuadamente buenas prácticas para diseño de arquitecturas de software.
- 2.3.4. Entiende adecuadamente cada uno de los elementos arquitecturales y los documenta.
- 2.3.5. Comprende las arquitecturas de software desarrolladas.
- 2.3.6. Implementa arquitecturas de software.
- 2.3.7. Diseña soluciones nativas para la nube.
- 2.3.8. Diseña soluciones híbridas.
- 2.3.9. Despliega soluciones en la nube.

### 3. CONTENIDO TEMÁTICO

#### 3.1. Unidad de Aprendizaje 1: Diseño de Soluciones Arquitectónicas

##### Competencias de la asignatura

- 3.1.1. Comprende los conceptos básicos de arquitectura de software y sus elementos.
- 3.1.2. Analiza requisitos arquitecturales de software, aplica patrones y tácticas para dar solución a problemas concretos.
- 3.1.3. Aplica adecuadamente buenas prácticas para diseño de arquitecturas de software.
- 3.1.4. Entiende adecuadamente cada uno de los elementos arquitecturales y los documenta.
- 3.1.5. Comprende las arquitecturas de software desarrolladas.

##### Logros de aprendizaje:

- 3.1.6. Conoce y comprende los conceptos y definiciones de arquitectura de software.
- 3.1.7. Diseña arquitecturas de software basado en el método ADD v3.
- 3.1.8. Documenta arquitecturas de software.
- 3.1.9. Demuestra capacidad de investigación y aplicación práctica de los tópicos tratados en la unidad de aprendizaje.

#### Semanas Nº : 1 a 8

<b>Temas</b>	<b>Actividades:</b>
1. Ingeniería de software.	a) Presentación del curso.
2. Fundamentos de arquitectura de software.	b) Revisión del sílabo.
3. Ciclo de desarrollo de la arquitectura.	c) Elección del delegado del curso.
4. Drivers arquitecturales.	d) Conformación de equipos.
5. Atributos de calidad.	e) Debatir sobre los drivers arquitecturales.
6. QAW.	f) Explicar el taller de atributos de calidad.
7. Utility tree.	g) Explicar el método ADD.
8. Attribute driven design v3 (ADD).	h) Aplicar ADD a un caso.
9. Modelo C4 y UML.	i) Debatir sobre la aplicación de DDD.
10. Domain driven design (DDD).	j) Exposición de trabajo parcial (semana 8).
<b>Lectura selecta</b> Cervantes, H., & Kazman, R. (2024). Designing software architectures: a practical approach. Second edition. Addison-Wesley.	



## UNIVERSIDAD NACIONAL MAYOR DE SAN MARCOS

Universidad del Perú. Decana de América  
Facultad de Ingeniería de Sistemas e Informática  
Vicedecanato de Investigación y Posgrado  
Unidad de Posgrado

<b>Estrategias didácticas a emplear:</b>
--

- |   |
|---|
| <ul style="list-style-type: none"><li>- Clase Magistral</li><li>- Foros</li></ul> |
|---|

<b>Equipos y materiales:</b> Computadora, Plataforma de Aula Virtual, Videoconferencia, Recursos didácticos.
--

### 3.2. Unidad de Aprendizaje 2: Desarrollo de Soluciones de Software

#### Competencias de la asignatura

- 3.2.1. Comprende las arquitecturas de software desarrolladas.
- 3.2.2. Implementa arquitecturas de software.

#### Logros de aprendizaje:

- 3.2.3. Implementa arquitecturas de software conforme al diseño definido.
- 3.2.4. Demuestra capacidad de investigación y aplicación práctica de los tópicos tratados en la unidad de aprendizaje.

#### Semanas Nº : 9 a 11

<b>Temas</b> <ul style="list-style-type: none"><li>1. Implementación de patrones arquitecturales.</li><li>2. Arquitectura dirigida por eventos.</li><li>3. CQRS.</li><li>4. Event sourcing.</li><li>5. Principios SOLID.</li><li>6. Clean code.</li><li>7. Patrones GoF.</li><li>8. Refactoring.</li></ul>	<b>Actividades:</b> <ul style="list-style-type: none"><li>a) Talleres prácticos.</li></ul>
<b>Lectura selecta</b> <p>E. Gamma, R. Helm, R. Johnson, and J. Vlissides, Design patterns: elements of reusable object-oriented software. Pearson Education, 1994.</p>	
<b>Estrategias didácticas a emplear:</b> <ul style="list-style-type: none"><li>- Clase Magistral</li><li>- Foros</li></ul>	
<b>Equipos y materiales:</b> Computadora, Plataforma de Aula Virtual, Videoconferencia, Recursos didácticos.	

### 3.3. Unidad de Aprendizaje 3: Proyectos Basados en la Nube

#### Competencias de la asignatura

- 3.3.1. Diseña soluciones nativas para la nube.
- 3.3.2. Despliega soluciones en la nube.

#### Logros de aprendizaje:

- 3.3.3. Diseña e implementa arquitecturas de software en la nube.



## UNIVERSIDAD NACIONAL MAYOR DE SAN MARCOS

Universidad del Perú. Decana de América  
Facultad de Ingeniería de Sistemas e Informática  
Vicedecanato de Investigación y Posgrado  
Unidad de Posgrado

- 3.3.4. Demuestra capacidad de investigación y aplicación práctica de los tópicos tratados en la unidad de aprendizaje.

### Semanas N° : 12 y 13

<b>Temas</b> 1. Cloud computing. 2. Despliegue en AWS, Azure y Google Cloud.	<b>Actividades:</b> a) Debatir sobre computación en la nube. b) Talleres prácticos.
<b>Lectura selecta</b> C. Richardson, Microservices Patterns. Manning Publications Co, 2018.	
<b>Estrategias didácticas a emplear:</b> <ul style="list-style-type: none"><li>- Clase Magistral</li><li>- Foros</li></ul>	
<b>Equipos y materiales:</b> Computadora, Plataforma de Aula Virtual, Videoconferencia, Recursos didácticos.	

### 3.4. Unidad de Aprendizaje 4: Diseño de Soluciones Híbridas

#### Competencias de la asignatura

- 3.4.1. Diseña soluciones híbridas.

#### Logros de aprendizaje:

- 3.4.2. Diseña arquitecturas de software en la nube.  
3.4.3. Demuestra capacidad de investigación y aplicación práctica de los tópicos tratados en la unidad de aprendizaje.

### Semanas N° : 14 a 16

<b>Temas</b> 1. Soluciones híbridas.	<b>Actividades:</b> a) Debatir sobre modelos híbridos. b) Exposición del trabajo final (semana 15). c) Examen final (semana 16).
<b>Lectura selecta</b> C. Richardson, Microservices Patterns. Manning Publications Co, 2018.	
<b>Estrategias didácticas a emplear:</b> <ul style="list-style-type: none"><li>- Clase Magistral</li></ul>	
<b>Equipos y materiales:</b> Computadora, Plataforma de Aula Virtual, Videoconferencia, Recursos didácticos.	



## UNIVERSIDAD NACIONAL MAYOR DE SAN MARCOS

Universidad del Perú. Decana de América  
Facultad de Ingeniería de Sistemas e Informática  
Vicedecanato de Investigación y Posgrado  
Unidad de Posgrado

### 3.5. Bibliografía.

- 3.5.1. E. Gamma, R. Helm, R. Johnson, and J. Vlissides, Design patterns: elements of reusable object-oriented software. Pearson Education, 1994.
- 3.5.2. Clements. P., Kazman. R. & Klein. M. (2002). Evaluating Software Architecture: Methods and Case Studies, Addison Wesley.
- 3.5.3. M. Fowler, Patterns of enterprise application architecture. Addison-Wesley Longman Publishing Co., Inc, 2002.
- 3.5.4. E. Evans, Domain-driven design: tackling complexity in the heart of software. Addison-Wesley Professional, 2004.
- 3.5.5. Taylor, R., Medvidovic, N. & Dashofy, E. (2010). Software Architecture: Foundations, Theory and Practice, Wiley.
- 3.5.6. Clements, P., Bachmann, Felix., Bass, L., Garlan, D., Ivers, J., Little, R., Merson, P., Nord, R., & Stafford, J. (2011). Documenting Software Architecture: Views and Beyond, MA: Addison-Wesley.
- 3.5.7. Dugan, D. (2012). Enterprise Software, Architecture and Design: Entities, Services and Resources, Wiley.
- 3.5.8. Rozanski, N. & Woods, E. (2012). Software Systems Architecture: Working with Stakeholders Using Viewpoints and Perspectives, Addison Wesley.
- 3.5.9. Feiler, P. & Gluch, D. (2013). Model-Based Engineering with AADL, Addison Wesley.
- 3.5.10. Richards, M. (2013). Software Architecture Patterns: Understanding Common Architecture Patterns and When to Use Them. Hoboken, O'reilly.
- 3.5.11. IEEE, SWEBOK, Guide to the Software Engineering Body of Knowledge, Version 3.0. IEEE Computer Society, USA, 2014.
- 3.5.12. Bass, L., Weber, I. & Zhu, L. (2015). DevOps: A Software Architecture's Perspective, Addison Wesley.
- 3.5.13. Vaughn Vernon, Domain-Driven Design Distilled. Addison-Wesley Pearson Education, Inc, 2016.
- 3.5.14. R. C. Martin, Clean Architecture: A Craftsman's Guide to Software Structure and Design. Prentice Hall, 2017.
- 3.5.15. C. Richardson, Microservices Patterns. Manning Publications Co, 2018.
- 3.5.16. Richards, M., & Ford, N. (2020). Fundamentals of Software Architecture: An Engineering Approach. O'Reilly Media.
- 3.5.17. L., Clements, P., & Kazman, R. (2021). Software Architecture in Practice.
- 3.5.18. H. Cervantes, and R. Kazman, Designing Software Architectures: A Practical Approach. Second edition. Addison-Wesley Professional, 2024.

### Recursos Electrónicos

- 3.5.19. IEEE Xplore: <https://ieeexplore.ieee.org/Xplore/home.jsp>
- 3.5.20. ACM: <https://dl.acm.org>
- 3.5.21. ScienceDirect: <https://www.sciencedirect.com>
- 3.5.22. SEI: <https://www.sei.cmu.edu>
- 3.5.23. InfoQ: <https://www.infoq.com/architecture-design>



## UNIVERSIDAD NACIONAL MAYOR DE SAN MARCOS

Universidad del Perú. Decana de América  
Facultad de Ingeniería de Sistemas e Informática  
Vicedecanato de Investigación y Posgrado  
Unidad de Posgrado

#### 4. ESTRATEGIAS METODOLÓGICAS

La metodología de enseñanza es no presencial, los contenidos y material de enseñanza se publicará en el Aula Virtual de la Unidad de Posgrado, la sesión de clase se desarrollará mediante videoconferencias en línea (síncronas) según la programación horaria del curso y publicada como material de enseñanza en el Aula Virtual. Entre las estrategias de enseñanza a utilizar se tiene: La clase magistral, discusión grupal, aprendizaje basado en problemas ABP, aprendizaje basado en proyectos, asesoría, foros y talleres. El presente curso utilizará herramientas colaborativas.

#### 5. ESTRATEGIAS DE EVALUACIÓN

##### 5.1. Modalidades de evaluación:

La modalidad de evaluación comprende desempeño, examen parcial, examen final y sustentación del proyecto final, los cuales deberán ser presentados mediante la plataforma del Aula Virtual, asimismo las exposiciones de trabajos deberán realizarse mediante videoconferencias en línea cuyos enlaces a sus contenidos deberán registrarse en la plataforma del Aula Virtual.

##### 5.2. Criterios de evaluación

Ponderación de cada una de las modalidades.

Modalidades	Porcentaje
1. Desempeño.	15%
2. Trabajo parcial (TP). Semana 8.	20%
3. Trabajo final (TF). Semana 15.	30%
4. Examen final (EF). Semana 16.	35%
Total	100%

Obtención del promedio final:  $\text{Promedio de } 1(15\%) + 2(20\%) + 3(30\%) + 4(35\%)$

**Nota aprobatoria mínima: 14**

##### 5.3. Requisitos para aprobar la asignatura:

Es un requisito contar con asistencia mayor al 70% las cuales serán registradas por el docente en la plataforma del Aula Virtual según la programación académica.

*Ciudad Universitaria, Septiembre del 2024*