



Facultad de Ingeniería  
Escuela Profesional de Ingeniería Informática

**SÍLABO 2023-II**

**I. DATOS ADMINISTRATIVOS**

1. Asignatura	: ARQUITECTURA Y EVOLUCIÓN DE SOFTWARE
2. Código	: IF 0705
3. Naturaleza	: Teórico-práctica
4. Condición	: Obligatorio
5. Requisitos	: IF0604 Análisis de sistemas y diseño de Software
6. Nro. Créditos	: 3.5
7. Nro. de horas	: 5 (Teoría= 2 Taller = 3)
8. Semestre Académico	7
9. Docente	: Miguel Arrunátegui
Correo Institucional	: gipsy.arrunategui@urp.edu.pe

**II. SUMILLA**

El curso de Arquitectura y evolución de Software corresponde al VII ciclo de formación de la Escuela Académico Profesional de Ingeniería Informática. El curso es de naturaleza teórico-práctica. Tiene como propósito desarrollar en el estudiante las capacidades necesarias para definir, probar especificar y documentar la Arquitectura del software y gestionar su evolución. Tiene como objeto desarrollar las capacidades para seleccionar, evaluar, diseñar, probar y documentar la arquitectura del software. Asimismo, desarrollar las capacidades para llevar una adecuada Administración de la configuración del software, permitiendo su adecuada evolución.

**III. COMPETENCIAS GENÉRICAS A LAS QUE CONTRIBUYE LA ASIGNATURA**

- Pensamiento crítico y creativo
- Autoaprendizaje

**IV. COMPETENCIAS ESPECÍFICAS A LAS QUE CONTRIBUYE LA ASIGNATURA**

- Diseño en ingeniería
- Soluciona problemas de Ingeniería
- Práctica moderna de la ingeniería

**V. DESARROLLA EL COMPONENTE DE: INVESTIGACIÓN ( ) RESPONSABILIDAD SOCIAL ( )**

**VI. LOGRO DE LA ASIGNATURA**

Al finalizar la asignatura el estudiante: Comprende los conceptos de la arquitectura del software y su racionalidad, Aplica la POO y los patrones de diseño, conoce, comprende y aplica los estilos de la arquitectura de software, diseña e implementa arquitecturas modernas basadas u orientada a servicios, diseña e implementa API's de servicios SOAP y REST, documenta la arquitectura del software, Planifica y realiza la Administración de la configuración del Software, gestiona el control de cambios, realiza auditoria Física y Funcional a la configuración del software durante su ciclo de vida.



## VII. PROGRAMACIÓN DE CONTENIDOS

<b>UNIDAD II: INTRODUCCIÓN A LA ARQUITECTURA DE SOFTWARE Y DISEÑO ORIENTADO A OBJETOS</b>	
<b>LOGRO DE APRENDIZAJE:</b> Al finalizar la unidad, el estudiante Comprende los conceptos de la arquitectura del software y su racionalidad, Aplica la Programación Orientada a Objetos y los patrones de diseño.	
Semana	Contenido
1	Introducción a la arquitectura de software, aspectos históricos, conceptos clave. Programación orientada a Objetos: Encapsulamiento, herencia y polimorfismo. Interfaces, clases abstractas, métodos estáticos
2,3,4	Patrones de diseño. Patrones de diseño Gang of Four. Implementación de patrones más usados: Singleton, Iterator, Chain of responsibility, Decorator, Memento, Adapter, Abstract Factory, Prototype, Facade y Bridge
5,6,7	Patrones de arquitectura de empresas (PoEAA) de Martin Fowler. Expresiones regulares. Programación de filtros usando expresiones regulares.
8	Monitoreo y Retroalimentación. Evaluación del Logro

<b>UNIDAD III: DISEÑO Y DOCUMENTACIÓN DE LA ARQUITECTURA DE SOFTWARE</b>	
<b>LOGRO DE APRENDIZAJE:</b> Al finalizar la unidad, el estudiante conoce, comprende y aplica los estilos de la arquitectura de software, diseña e implementa arquitecturas modernas. Documenta la arquitectura de software.	
Semana	Contenido
9	Método de diseño y atributos de la calidad de la Arquitectura de software Decisiones Arquitectónicas. Documentación de la Arquitectura: Modelo de Vistas, Modelo C4. Definición de patrones arquitectónicos. Implementación de patrones arquitectónicos: Tuberías y filtros
10	Implementación de patrones arquitectónicos: Patrón de capas. Migración de un software monolítico a capas y tiers. Monitoreo y Retroalimentación. Evaluación del Logro

<b>UNIDAD IV: PATRONES ARQUITECTÓNICOS Y ARQUITECTURAS BASADAS EN SERVICIOS</b>	
<b>LOGRO DE APRENDIZAJE:</b> Al finalizar la asignatura, el estudiante diseña e implementa arquitecturas modernas basadas u orientada a servicios, diseña e implementa API's de servicios SOAP y REST	
Semana	Contenido
11	Arquitectura orientada a servicios: Servicios web SOAP Implementación de patrones arquitectónicos: Modelo, vista, controlador. Arquitectura orientada a servicios: Desarrollo de API's.
12	Patrón de Hexagonal o Ports y Adapters. Arquitectura basada en servicios: SOAP, REST, Microservicios. Monitoreo y Retroalimentación. Evaluación del Logro
13	Monitoreo y Retroalimentación. Evaluación del Logro

<b>UNIDAD I: AMINISTRACIÓN DE LA CONFIGURACIÓN Y EVOLUCIÓN DE SOFTWARE</b>	
<b>LOGRO DE APRENDIZAJE:</b> Al finalizar la unidad, el estudiante Planifica y realiza la Administración de la configuración del Software, gestiona el control de cambios, realiza auditoria Física y Funcional a la configuración del software durante su ciclo de vida.	
Semana	Contenido
14	La Ingeniería de software. La disciplina de Administración de la configuración. Los Procesos de la SCM.
15	Conoce los requerimientos de información que son capturados a través del Plan de SCM. Identifica y establece las relaciones entre los elementos de configuración para un proyecto de



	software. Definir el proceso de solicitudes de cambios, evaluar el impacto de las peticiones de cambio y gestionar las solicitudes para su desarrollo. Comprender la importancia de la auditoría Física / Funcional, adaptar el proceso de auditoría a una línea base y presentar reportes que muestren el estado actual del producto de software.
<b>16</b>	El mantenimiento del Software. Conocer el proceso y requerimientos a satisfacerse en el Plan de Mantenimiento del software. Monitoreo y Retroalimentación. Evaluación del Logro
<b>17</b>	<b>EVALUACIÓN SUSTITUTORIA CON PRODUCTO FINAL: RÚBRICA</b>

#### VIII. ESTRATEGIAS DIDÁCTICAS

Aula invertida, Aprendizaje Colaborativo, disertación, Aprendizaje basado en problemas, aprendizaje basado en juegos

#### IX. EVALUACIÓN

La modalidad no presencial se evaluará a través de productos que el estudiante presentará al final de cada unidad. Los productos son las evidencias del logro de los aprendizajes y serán evaluados a través de rúbricas cuyo objetivo es calificar el desempeño de los estudiantes de manera objetiva y precisa.

Retroalimentación. En esta modalidad no presencial, la retroalimentación se convierte en aspecto primordial para el logro de aprendizaje. El docente devolverá los productos de la unidad revisados y realizará la retroalimentación respectiva.

UNIDAD	INSTRUMENTOS	PORCENTAJE
<b>I</b>	PC1	10%
<b>II</b>	Examen Parcial	20%
<b>III</b>	PC2	15%
<b>IV</b>	Examen Final	30%
<b>V</b>	Trabajo Final	20%
<b>VI</b>	Participación	5%

#### X. RECURSOS

- Equipos: computadora, laptop, Tablet, celular
- Materiales: Presentaciones PowerPoint, Apuntes de clase del Docente, Guías de laboratorio, Foros de discusión, Tareas, Presentación de resultados.
- Plataformas: Moodle, Blackboard Collaborate Ultra, SmarterASP.net, Servidor remoto IIS, Servidor DB SQL Server 2017

#### XI. REFERENCIAS

##### Bibliografía Básica

1. *Gamma, E; Helm, R; Jhonson, R y Vissides, J. Patrones de Diseño.* Editorial Addison Wesley. 2003
2. *Bass, L., Clements, P. y Kazman, R. (2003). Software architecture in practice.* Boston: Addison-Wesley. SEI series in software engineering.
3. *Martin Fowler, David Rice, Matthew Foemmel, Edward Heatt, Robert Mee, Randy Stafford. Patterns Of Enterprise Application Architecture.* Addison-Wesley, November 05, 2002
4. *Kruchten, Philippe. The Rational Unified Process: An Introduction (2nd Edition).* Addison-Wesley Professional; 2 edition (March 14, 2000)
5. *Cerami, Ethan. Web services Essentials (O'Reilly XML).* O'Reilly Media, Inc.; 1 edition (February 2002)
6. *Berlack, h. Ronald. Software Configuration Management.* John Wiley & Sons 1992.
7. *Rational Software Corporation. Software Configuration Management: A Clear Case for IBM Rational ClearCase and ClearQuest UCM.* 2004
8. *Sommerville, Ian. Ingeniería de Software.* Addison Wesley.



## XII. REFERENCIAS DIGITALES

### Bibliografía ACM

July 2021

Published By ACM

**RESTest: automated black-box testing of RESTful web APIs**

Alberto Martin-Lopez, Sergio Segura, Antonio Ruiz-Cortés

ISSTA 2021: Proceedings of the 30th ACM SIGSOFT International Symposium on Software Testing and Analysis

July 2021, pp 682–685 <https://doi.org/10.1145/3460319.3469082>