



Servicio de Instrucción	Oficialización: Director Edwin Maraví
<h1>Architect</h1>	Año:2018
	Código: C05_2018

Sílabo Architect

I. DATOS GENERALES

Año de vigencia	:	2018
Número de sesiones	:	20 (4 horas por sesión)
Nº de Horas académicas	:	80
Requisitos	:	Conocimientos de Desarrollo Web de Java (Expert). 06 meses de experiencia en desarrollo Java

II. FUNDAMENTACIÓN

El curso orienta a los participantes en el correcto uso de las librerías, frameworks Java y servidores de aplicaciones para poder establecer mecanismos de comunicaciones entre aplicaciones heterogéneas. El curso orienta al participante en uso de herramientas y conocimientos de tecnologías necesarias para poder establecer procedimientos de Integración de Aplicaciones.

III. UNIDAD DE COMPETENCIA

Integrar aplicaciones heterogéneas basadas en arquitectura orientada a servicios.

IV. CAPACIDADES TERMINALES Y CRITERIOS DE EVALUACIÓN

CAPACIDAD TERMINAL	CRITERIOS DE EVALUACIÓN
Usar mecanismos y componentes para integración de aplicaciones heterogéneas.	Reconoce conceptos de Sistemas distribuidos.
	Reconoce todos los estándares de Servicios Web
	Reconoce sistemas basados en mensajería.
	Reconoce conceptos de SOA: BPM y ESB
	Aplica métricas de rendimiento de Software

V. CONTENIDOS

Comunicación en sistemas distribuidos

- Comunicación bajo protocolo HTTP.



Servicio de Instrucción	Oficialización: Director Edwin Maraví
<h1>Architect</h1>	Año:2018
	Código: C05_2018

- Normativas XML y JSON.
- Comparativa XML vs JSON.
- Herramientas Java para el manejo de XML y JSON.

Arquitecturas y estándares Web Services

- Arquitectura de Web Services.
- Interoperabilidad de Web Services.
- Estándares para implementación de SOAP Web Services: SOAP, WSDL, UDDI y WS-Security.
- Estándares para implementación de RESTful Web Services: HTTP, WADL, URL/URI y Internet Media Types.
- SOAP Web Services vs REST Web Services. Java API for XML-based Web Services (JAX-WS)
 - Especificación JSR 224 – Java API for XML-Based Web Services.
- Especificación JAX-WS – Metro Reference Implementation.
- SOAP 1.1/1.2.

Anotaciones.

- XML/HTTP Binding.
- JAXB. Java API for RESTful Web Services (JAX-RS)

Arquitectura REST.

- JSR 311 – Java API for RESTful Web Services.
- JAX-RS Jersey Reference Implementation.
- JSR 339 – JAX-RS 2.0.

Arquitectura orientada a mensajes

- Message Oriented Middleware (MOM).
- Modelos de mensajería: Point to Point y Publish/Subscribe. □ Procesamiento síncrono vs asíncrono. Java Message Service (JMS) □ Especificación JSR 914 – Java Message Service API.
- Principales implementaciones del mercado.



Servicio de Instrucción	Oficialización: Director Edwin Maraví
<h1>Architect</h1>	Año:2018
	Código: C05_2018

- JMS – ActiveMQ.
 - Point to Point (Queues).
 - Publish/Subscribe (Topic).
- Tipos de mensajes.
 - Text Message.
 - Stream Message.
 - Object Message.
- Java Message Services con Message-Driven Beans □ Message-Driven Beans.
- Enterprise JavaBeans.
- Implementación JMS Message-Driven Beans.

(ESB) Enterprise Services Bus

- Arquitectura ESB.
- Arquitectura JBoss ESB Server.
- JBoss ESB.
 - Providers.
 - Message Interface.
 - Service Listeners.
 - Service Actions.
 - Custom Actions.
 - Base Web Service.
- Business Rule Management Systems

API Drools.

- Implementación de reglas con Drools Expert: reglas básicas, validaciones y transformaciones. Implementación de flujos con JBPM5.
- Editor BPMN.
- Events, actions, tasks y gateways.



Servicio de Instrucción	Oficialización: Director Edwin Maraví
Architect	Año:2018
	Código: C05_2018

Métricas de rendimiento de software

- Normativa de medición del rendimiento de software.
- Herramientas de medición.
 - JMeter.
 - Selenium IDE.
 - Selenium Driver.
 - SOAP UI.
 - Bases para implementación de una herramienta manual.
- Dimensionamiento y pruebas de rendimiento de software

Aplicaciones de monitoreo

- Monitoreo de Rendimiento (APM).
 - Experiencia del usuario final.
 - Arquitectura de ejecución.
 - Transacciones de negocio.
 - Monitoreo de componentes.
- Monitoreo de Actividades (BAM).
 - Alertas y visualización.
 - Procesos y análisis.
 - Bases de datos operacionales

Service Oriented Architecture

- ¿Qué es SOA? .
- Ciclo de vida. Beneficios.
- Arquitectura de software de aplicaciones.
 - Estilo arquitectónico.
 - Patrón arquitectónico.
 - Patrón de diseño.



Servicio de Instrucción	Oficialización: Director Edwin Maraví
<h1>Architect</h1>	Año:2018
	Código: C05_2018

- Service Oriented Architecture.
 - Arquitectura EDA.
 - SOA 2.0.
- Arquitectura distribuida.
- Teorema de Brewer (CAP).

VI. METODOLOGÍA

- Curso teórico-práctico.
- Exposición del profesor, ayudas audiovisuales, diálogo abierto y debate.
- Realización de casos prácticos en laboratorio.
- Realización de tareas en aula virtual.
- Generación de conocimiento mediante investigación y foros de discusión en aula virtual.

VII. EVALUACIÓN

La evaluación, como constatación del logro de los objetivos que es alcanzar las capacidades mencionadas se dará mediante la siguiente ponderación:

04 Tareas (se elimina la menor)	15 %
04 Evaluaciones Virtuales (se elimina la menor)	10 %
04 Investigaciones (se elimina la menor)	15 %
Evaluaciones prácticas 1 y 2 (se elimina la menor)	20 %
Evaluaciones prácticas 3 y 4	40 %
TOTAL	100 %

VIII. BIBLIOGRAFÍA

- AspectJ in Action. Practical Aspect-Oriented Programming. Ramnivas Laddad
- Introduction to Spring Framework. Rod Johnson.



Servicio de Instrucción	Oficialización: Director Edwin Maraví
Architect	Año:2018
	Código: C05_2018

IX. PÁGINAS WEB DE APOYO:

- <http://www.tutorialspoint.com/hibernate/>
- <http://docs.oracle.com/javaee/6/tutorial/doc/bnbtg.html>
- <http://mybatis.github.io/mybatis-3/es/getting-started.html>
- <http://www.springhispano.org/?q=node/613>
- <http://www.springbyexample.org/pdf/SpringByExample.pdf>