



Ministerio de Cultura y Educación
Universidad Nacional de San Luis
Facultad de Ciencias Físico Matemáticas y Naturales
Departamento: Informatica
Area: Area IV: Pr. y Met. de Des. del Soft.

(Programa del año 2023)

I - Oferta Académica

Materia	Carrera	Plan	Año	Período
ARQUITECTURA DE SOFTWARE	ING. INFORM.	026/1 2- 08/15	2023	1° cuatrimestre

II - Equipo Docente

Docente	Función	Cargo	Dedicación
RIESCO, DANIEL EDGARDO	Prof. Responsable	P.Asoc Exc	40 Hs

III - Características del Curso

Credito Horario Semanal				
Teórico/Práctico	Teóricas	Prácticas de Aula	Práct. de lab/ camp/ Resid/ PIP, etc.	Total
3 Hs	Hs	Hs	2 Hs	6 Hs

Tipificación	Periodo
B - Teoria con prácticas de aula y laboratorio	1° Cuatrimestre

Duración			
Desde	Hasta	Cantidad de Semanas	Cantidad de Horas
13/03/2023	24/06/2023	15	75

IV - Fundamentación

La arquitectura de software se ha vuelto indispensable con el crecimiento en complejidad y tamaño de los sistemas de software que se desarrollan. Esta describe la organización fundamental del sistema, y esta descripción ilumina decisiones de diseño de alto nivel: su composición y partes interactuantes, sus interacciones y patrones de comunicación presentes, y las propiedades claves de sus partes en las que la totalidad del sistema descansa o impone al sistema. Se convierte entonces en un modelo relativamente pequeño con respecto al sistema real, que permite a los distintos interesados comunicarse, y tratar decisiones de diseño que tendrán un profundo impacto en el trabajo subsiguiente, y en el éxito del sistema.

V - Objetivos / Resultados de Aprendizaje

Al finalizar el curso se espera que el alumno sea capaz de:

- * Describir las arquitecturas de software con precisión y de manera efectiva.
- * Desarrollar las habilidades para reconocer patrones arquitectónicos en sistemas de software existentes.
- * Proponer arquitecturas alternativas para encarar un determinado problema.
- * Desarrollar módulos de software correspondientes a una definición de arquitectura determinada.
- * Aplicar herramientas para definir arquitecturas.
- * Utilizar el “dominio de conocimiento” para instanciar una arquitectura para una familia de aplicaciones en particular.
- * Diseñar arquitecturas que cumplan con requisitos de seguridad establecidos.

VI - Contenidos

Unidad 1 - Introducción

La Arquitectura del Software. Definición. Que es y que no es Arquitectura. El ciclo de negocio de la arquitectura. Procesos de software y el ciclo de negocio de la arquitectura. El papel de la arquitectura en proyectos de software de alta complejidad. Niveles de diseño de software y el nivel arquitectónico del diseño de software. Características de una buena arquitectura.

Unidad 2 - Lenguajes de Descripción Arquitectónica

Lenguajes para el modelado, la descripción y prueba de la arquitectura. Documentación. IEEE 42010. Distintas Vistas Arquitectónicas. Vista de Modelos y Métodos Formales. Vista de Módulo. Estilos. Vista de Componente y Conector. Vista de Asignación. Estilos. Despliegue. Instalación.

Unidad 3 – Análisis y Diseño Arquitectónico

Arquitectura lógica: conjunto de patrones y abstracciones coherentes que proporcionan el marco de referencia necesario para guiar la construcción del software para un sistema de información. La Arquitectura de Software como fundamento para que analistas, diseñadores, programadores trabajen coherentemente. Arquitecturas y atributos de Calidad. Disponibilidad. Interoperabilidad. Modificabilidad. Performance. Seguridad. Prueba. Usabilidad. Criterios de análisis y diseño. Evaluación de la Arquitectura.

Unidad 4 – Patrones Arquitectónicos

Caracterización de patrón arquitectónico. Patrones, Modelos de Referencia y Arquitecturas de Referencia. Catálogos de patrones arquitectónicos. Patrones de arquitecturas de sistemas de software de mayor difusión. Catálogos de Patrones Arquitectónicos. Relaciones entre patrones. Descripción del patrón. Capas. Multi-Tier. Flujos de Datos. Sistemas distribuidos. Broker. Sistemas interactivos. Modelo-Vista-Controlador. Sistemas Adaptables. Microkernel. Reflexivos. Web. Basados en Eventos. Basados en Recursos. Servicios. Computación en la Nube.

Unidad 5 – Orientado a Servicios

Evolución. Topología. Orientación a Objetos vs. Orientación a Servicios. Service Oriented Architecture (SOA). Componentes y su colaboración en SOA. Macro servicios o micro servicios. Stack de servicios Web. WSDL. SOAP. UDDI. Coreografía. Estilo Arquitectónico REpresentational State Transfer (RESTfull). Direccionalidad. Interfaz uniforme. Orientado a Recursos. Servicios vs Microservicios.

Unidad 6 – Arquitectura con Seguridad

Introducción. Atributos de Seguridad. Dimensiones de Seguridad. Seguridad con SOA. Protocolo SSL. Estandars de Seguridad de los Servicios Web. Amenazas. Mitigación. Seguridad en Spark.

Unidad 7 - Transacciones en SOA

Introducción. ACID. Protocolo COMMIT en dos fases. Patrón de Transacción compuesta. Patrón de Transacción de larga duración. Patrón de Negociación.

Unidad 8 - Arquitectura para Aplicaciones Blockchain

Definiciones. Funcionamiento. Red de Pares. Ethereum. Características. Estructura. Dinámica. Protocolos de Consenso. Hashing. Firma digital. Actores en Blockchain. Libro Mayor. Cadena de bloques. Transacción. Mineros. Modelos de Gobernanza. Arquitectura como Almacenamiento. Arquitectura como Infraestructura Digital. Propiedades y requisitos no funcionales. Arquitectura: Análisis y Diseño. Diseño de Aplicaciones basadas en Blockchain. Criptomonedas. Contratos inteligentes. Casos de Uso con la Blockchain Federal Argentina.

VII - Plan de Trabajos Prácticos

1. Análisis, diseño e implementación de distintos problemas usando REST.
2. Análisis, diseño e implementación de distintos problemas usando Modelo-Vista-Controlador y Servicios Web.
3. Análisis, diseño e implementación de distintos problemas usando SOA con WSDL.
4. Análisis y Diseño Arquitectónico para distintos tipos de problemas. Definir Estilos Arquitectónicos.
5. Análisis, diseño e implementación de distintos problemas usando el patron arquitectónico Reflexivo.

Laboratorio: Aplicar los conceptos aprendidos para instanciar una arquitectura de una familia de aplicaciones y construir el software para la definición de dicha arquitectura, combinando distintos patrones arquitectónicos. (40 hs).

VIII - Regimen de Aprobación

Existen dos niveles para su aprobación:

a) Regularización solamente: Para regularizar la materia se deberá:

- 1.- Tener como mínimo un 80% de asistencia a clases prácticas.
- 2.- Tener los trabajos solicitados por la cátedra aprobados.
- 3.- Presentación y aprobación del proyecto integrador de laboratorio con nota mayor o igual a 7 (siete).
- 4.- Tener aprobado con el 60% todos los proyectos solicitados por la cátedra, en las fechas solicitadas. En caso de no cumplir, tendrá recuperatorios con fechas diferenciadas.

b) Promoción sin examen final: Para regularizar y aprobar la materia se deberá:

- 1.- Cumplir con los requisitos a.1, a.2 y a.3.
- 2.- Haber aprobado los proyectos solicitados con el 70%.
- 3.- Aprobar la presentación integral final ante la presencia del resto de los alumnos, con una nota mayor o igual a 7 (siete).

Aquellos alumnos que sólo regularicen la materia deberán rendir un examen final, en los turnos establecidos.

Alumnos Libres: Se aceptan alumnos libres debiendo previamente el alumno entregar y aprobar el proyecto integrador de laboratorio.

IX - Bibliografía Básica

- [1] Software Architecture in Practice, 4th Edition. Len Bass, Paul Clements, Rick Kazman. Addison Wesley (2021).
- [2] Documenting Software Architectures: Views and Beyond (2nd Edition), Paul Clements, Felix Bachmann, Len Bass, David Garlan, James Ivers, Reed Little, Paulo Merson, Robert Nord, Judith Stafford. Addison-Wesley (2011).
- [3] SOA Patterns. Arnon Rotem, Gal, Oz. (2012). Manning Publications.
- [4] ISO 25000 Calidad del Producto de Software. <http://iso25000.com/index.php/normas-iso-25000/iso-25010>
- [5] Software Modeling and Design: UML, Use Cases, Patterns, and Software Architectures, Hassan Gomaa, Cambridge University Press (2011).
- [6] Cloud Computing, Nayan B. Ruparelia. The MIT Press (2016).
- [7] NIST Special Publication 500-291 y NIST Special Publication 500-292. <http://www.nist.gov/itl/cloud/index.cfm>
- [8] ISO/IEC/IEEE 42010: Systems and software engineering — Architecture description: <http://www.iso-architecture.org/>
- [9] Fundamentals of Software Architecture: An Engineering Approach. Mark Richards, Neal Ford. O'Reilly Media (2020).
- [10] Software Architecture Patterns, Mark Richards, Mark Richards. O'Reilly (2015).
- [11] Pattern-Oriented Software Architecture: A System of Patterns, Volume 1, Frank Buschmann, Regine Meunier, Hans Rohnert, Peter Sornmerlad, Michael Stal. JOHN WILEY & SONS (2001).
- [12] Software Architecture: Foundations, Theory, and Practice. R. N. Taylor, N. Medvidovic, E. M. Dashofy. Wiley (2009).
- [13] Security for Software Engineers. James Helfrich. CRC Press (2019).
- [14] Java™ Coding Guidelines: 75 Recommendations for Reliable and Secure Programs. Fred Long, Dhruv Mohindra, Robert C. Seacord, Dean F. Sutherland, David Svoboda. Addison Wesley (2014).
- [15] Application Security Program Handbook: A guide for software engineers and team leaders, Version 2. Derek Fisher. Manning Publications (2021).
- [16] Hands-on cybersecurity for architects plan and design robust security architectures. Aslaner, Milad_Rerup, Neil. Packt Publishing (2018).
- [17] Blockchain for Dummies. Tiana Laurence. Wiley (2019).
- [18] Ethereum Smart Contract Development in Solidity. Gavin Zheng, Longxiang Gao, Liqun Huang, Jian Guan. Springer (2021).
- [19] Blockchain Federal Argentina, bfa.ar
- [20] Architecture for Blockchain Applications. Xiwei Xu, Ingo Weber, Mark Staples. Springer (2019)

X - Bibliografía Complementaria

- [1] Evaluating Software Architectures: Methods and Case Studies. Paul Clements, Rick Kazman, Mark Klein (2005). Addison-Wesley
- [2] Software Architecture: Perspectives on an Emerging Discipline. Mary Shaw, David Garland (1996). Prentice Hall
- [3] Models for Evaluating and Improving Architecture Competence, Len Bass, Paul Clements, Rick Kazman, Mark Klein (2008). http://resources.sei.cmu.edu/asset_files/TechnicalReport/2008_005_001_14972.pdf
- [4] Pattern-Oriented Software Architecture Volume 1: A System of Patterns. Frank Buschmann, Regine Meunier, Hans Rohnert, Peter Sommerlad, Michael Stal (1996). Wiley.
- [5] Patterns of Enterprise Application Architecture. Martin Fowler (2002). Addison-Wesley Professional.
- [6] MDA Explained: The Model Driven Architecture(TM): Practice and Promise, Anneke G. Kleppe, Jos Warmer, and Wim Bast (2003), Addison-Wesley Professional.
- [7] Pattern-Oriented Software Architecture For Dummies. Robert Hanmer. Wiley. 2012.
- [8] SOA Design Patterns, Thomas Erl (2009), Prentice Hall.
- [9] SOA with REST: principles, patterns & constraints for building enterprise solutions with REST. Erl, Thomas. (2013) Prentice Hall
- [10] Designing Software Architectures: A Practical Approach (SEI Series in Software Engineering). Humberto Cervantes, Rick Kazman. Addison-Wesley. 2016.
- [11] Service-Oriented Architecture: Analysis and Design for Services and Microservices. Thomas Erl. Prentice Hall. 2016
- [12] Building Microservices: Designing Fine-Grained Systems. Sam Newman. O'Reilly Media. 2015
- [13] Cloud Computing For Dummies. Judith Hurwitz, Robin Bloor, Marcia Kaufman, Fern Halper. Wiley Pub. 2010

XI - Resumen de Objetivos

Reconocer, describir, y desarrollar arquitecturas de software

XII - Resumen del Programa

Lenguajes de Descripción Arquitectónica
Análisis y Diseño Arquitectónico
Patrones Arquitectónicos
Arquitecturas Orientadas a Servicio
Arquitectura con Seguridad
Transacciones en SOA
Arquitectura para Aplicaciones Blockchain

XIII - Imprevistos

Modalidad de cursado Presencial.

XIV - Otros

.