Ministerio de Cultura y Educación **(Programa del año 2026)**
Universidad Nacional de San Luis
Facultad de Ciencias Físico Matemáticas y Naturales
Departamento: Informatica
Area: Area IV: Pr. y Met. de Des. del Soft.

I - Oferta Académica

Materia	Carrera	Plan	Año	Período
INGENIERIA WEB	TCO.UNIV.EN WEB	08/13	2026	1° cuatrimestre

II - Equipo Docente

Docente	Función	Cargo	Dedicación
PEREZ, NORMA BEATRIZ	Prof. Responsable	P.Adj Exc	40 Hs
GATICA, CLAUDIA RUTH	Auxiliar de Práctico	A.1ra Semi	20 Hs

III - Características del Curso

Credito Horario Semanal				
Teórico/Práctico	**Teóricas**	**Prácticas de Aula**	**Práct. de lab/ camp/ Resid/ PIP, etc.**	**Total**
3 Hs	2 Hs	3 Hs	Hs	8 Hs

Tipificación	Periodo
E - Teoria con prácticas de aula, laboratorio y campo	1º Cuatrimestre

Duración			
Desde	**Hasta**	**Cantidad de Semanas**	**Cantidad de Horas**
11/03/2026	19/06/2026	15	120

IV - Fundamentación

El desarrollo de aplicaciones web contemporáneo se despliega en un ecosistema de alta complejidad, caracterizado por la convergencia de tecnologías heterogéneas y una evolución constante de la infraestructura de red. Este escenario ha desplazado el foco del desarrollo tradicional hacia el diseño arquitectural, posicionándolo no solo como una etapa técnica, sino como la médula estratégica que garantiza la escalabilidad y sostenibilidad del software.

En la actualidad, las metodologías de desarrollo modernas reconocen este rol central, integrando la arquitectura desde las fases tempranas del ciclo de vida como el eje conductor que articula las actividades de ingeniería. Paralelamente, el avance tecnológico ha permitido la maduración y estandarización de patrones arquitecturales, los cuales actúan como soluciones probadas que optimizan la eficiencia y la calidad del diseño.

Sin embargo, se observa una disonancia crítica: a pesar del valor reconocido de la arquitectura, los marcos metodológicos actuales no han logrado integrar formalmente estos patrones en sus procesos de trabajo. Esta omisión, derivada de la necesidad de mantener la generalidad de los métodos o del desfase temporal entre la innovación tecnológica y la adaptación metodológica, genera una brecha que limita la aplicación sistemática de soluciones arquitectónicas estandarizadas en la industria.

V - Objetivos / Resultados de Aprendizaje

Objetivo General: Capacitar a los estudiantes en la concepción, diseño y desarrollo de arquitecturas de software robustas para entornos web, mediante la integración formal de patrones y metodologías orientadas a objetos, garantizando la escalabilidad, seguridad y eficiencia de los sistemas en ecosistemas digitales heterogéneos.

Objetivos Específicos:
-Jerarquizar el Diseño Arquitectural: Desarrollar la capacidad de posicionar la arquitectura como la médula del proceso de desarrollo, permitiendo que el estudiante tome decisiones técnicas fundamentadas en atributos de calidad (disponibilidad, mantenibilidad y rendimiento) desde las fases tempranas del proyecto.
- Sistematizar el Uso de Patrones: Instruir en la identificación, selección e implementación de Patrones de Diseño y Arquitecturales estandarizados, fomentando una práctica profesional que reduzca la complejidad accidental y la deuda técnica en aplicaciones empresariales distribuidas.
- Dominar Tecnologías de Grado Industrial y Seguridad: Proveer las competencias necesarias para implementar soluciones avanzadas bajo el estándar Jakarta EE, integrando servicios web, gestión de persistencia y protocolos de comunicación, con un enfoque crítico en la identificación y mitigación de vulnerabilidades en escenarios de IoT y redes complejas.

Al finalizar el dictado de la materia, se espera que el estudiante haya alcanzado los siguientes niveles de desempeño:
- Capacidad de abstracción y modelado.
- Presentación de diagramas de secuencia y clases que reflejen con precisión la lógica de una solución web empresarial.
- Implementación de soluciones basadas en Patrones. Aplicar un pensamiento crítico basado en patrones para resolver problemas recurrentes de software, seleccionando la solución óptima (Creacional, Estructural o Comportamental) según el contexto del problema.
- Competencia en ecosistemas Java/Jakarta EE. Conocimiento de como se desarrollan aplicaciones web de grado industrial utilizando el estándar Jakarta EE, gestionando correctamente el ciclo de vida de Servlets, la inyección de dependencias (CDI) y la persistencia de datos (JPA).
- Pensamiento Crítico en Seguridad y Entornos Heterogéneos. Identificar proactivamente vectores de ataque y vulnerabilidades en el diseño de sistemas conectados (IoT y Escenarios Digitales), proponiendo arquitecturas que sigan el principio de Seguridad por Diseño.

VI - Contenidos

Unidad 1: Fundamentos y Evolución de la Ingeniería Web
Se situa la arquitectura como el eje del desarrollo y comprender la transición de lo monolítico a lo distribuido.
- Evolución del Software Web: De aplicaciones estáticas a ecosistemas de aplicaciones distribuidas y heterogéneas (Internet como plataforma).
- Diseño Centrado en la Arquitectura (ACD): La arquitectura como médula del proceso de desarrollo. Calidad sistémica y atributos de calidad (escalabilidad, mantenibilidad).
- Taxonomía de Patrones: Diferenciación y jerarquía entre Patrones Arquitecturales (Macro), Patrones de Diseño (Micro) y Patrones de Solución.
- Paradigmas de Componentes y Servicios: Arquitecturas Orientadas a Servicios (SOA).
- Introducción a Microservicios y contenedores (Docker/Kubernetes) como evolución del componente tradicional.
- Nivelación Técnica: Revisión avanzada de Programación Orientada a Objetos, tipos de datos (JSON/XML) y modelado con UML 2.0.

Unidad 2: Patrones de Diseño y su Aplicación Metodológica
Fundamentación y Semántica de Patrones: donde utilizarlos y la formalización de patrones en el ciclo de vida del software.
- El Catálogo de GoF (Gang of Four) en el Contexto Web: Creacionales: (Singleton, Factory, Builder) aplicados a la gestión de recursos web. Estructurales: (Adapter, Proxy, Facade) para la integración de sistemas heterogéneos. Comportamentales: (Observer, Strategy, Command) en la gestión de eventos y flujos de datos.
-Metodología Orientada por Patrones: Identificación de estructuras y colaboraciones en el dominio del problema.
-Impacto de los patrones en la reducción de la deuda técnica y la complejidad accidental.
-Uso de patrones en frameworks modernos (Inyección de Dependencias y de Control).

Unidad 3: Arquitecturas Empresariales y Desarrollo de Aplicaciones Web
- El Ecosistema Jakarta EE (anteriormente J2EE): Arquitectura multicapa, contenedores de servlets y contenedores EJB.
- Arquitectura de Presentación y Lógica Web: Patrón MVC (Model-View-Controller): Implementación profunda. Separación de preocupaciones (Separation of Concerns).
- Componentes de Control: Servlets, Filtros y Listeners. Ciclo de vida y gestión de estado (Sesiones vs. Stateless).
- Componentes de Vista: Del JSP tradicional a los motores de plantillas y la comunicación con SPAs (Single Page

Applications).
- Capa de Negocio y Persistencia: Enterprise JavaBeans (EJB 3.x): Session Beans (Stateless/Stateful) y Message-Driven Beans para procesamiento asíncrono.
- Persistencia y JNDI: El rol de la inyección de recursos y el acceso a servicios de directorio.
- Comunicaciones y Servicios Web: RMI vs. REST/JSON: El cambio de paradigma hacia servicios livianos.
- Arquitecturas orientadas a eventos: Introducción a servicios de mensajería (JMS) en entornos distribuidos.
-Arquitecturas Web para dispositivos de recursos limitados (MQTT, protocolos ligeros).

VII - Plan de Trabajos Prácticos

Práctico 1: Ingeniería de Requisitos y Modelado de Ecosistemas Digitales
- Objetivo: Analizar la complejidad de sistemas web modernos y heterogéneos (IoT, APIs, Microservicios).
- Ejercitación de diagramas UML (CU, de secuencia, de actividad, entre otros). Exploración y análisis de una realidad Web concreta aplicando lo visto de la teoría.

Práctico 2: Patrones de Diseño y Refactorización
- Objetivo: Aplicar patrones para resolver problemas de acoplamiento y mantenibilidad.
- Ejercitación sobre Patrones de Diseño.

Práctico 3: Laboratorio de Arquitectura Web y Seguridad
- Objetivo: conocer y tener una experiencia de diseñar una arquitectura real del mercado que sea robusta, escalable y segura utilizando Jakarta EE, integrando patrones complejos para resolver problemas de diseño en entornos distribuidos.
- Actividad: Desarrollo integral de un caso de estudio real aplicando Modelado UML y el Modelo Integrado de Patrones de la cátedra. El estudiante realizará un estudio sistemático para definir una arquitectura web real (puede ser de grado industrial), aplicando técnicas avanzadas de diseño que orienten la construcción del software hacia la alta disponibilidad y la seguridad desde la concepción (Security by Design).

Requisitos para la aprobación del Laboratorio:
Para alcanzar la aprobación de esta unidad práctica, los estudiantes deberán cumplir con los siguientes hitos de desempeño:
- Selección y Definición de Escenarios: Identificar y describir al menos tres (3) escenarios críticos de Casos de Uso. Estos podrán ser seleccionados del repositorio de la cátedra o propuestos por el estudiante (sujetos a aprobación), asegurando que representen desafíos arquitecturales significativos (ej. gestión de concurrencia, integración de APIs o seguridad de datos).
- Modelado Arquitectural en UML: Desarrollar de manera exhaustiva el Modelo de Casos de Uso, el Modelo de Clases de Dominio y los Diagramas de Interacción (Secuencia/Comunicación).
-Especificación Técnica de los Diagramas de Secuencia: Cada diagrama debe alcanzar un nivel de detalle donde:
--Sea explícita la aplicación de patrones (Arquitecturales y de Diseño) integrados según el enfoque de la materia.
--Se evidencie una estructura desacoplada que satisfaga los atributos de calidad definidos (escalabilidad y mantenibilidad).
--La lógica de diseño sea unívoca y ejecutable, permitiendo la transición directa hacia un prototipo funcional en Jakarta EE sin ambigüedades técnicas.
-Defensa de Seguridad: Identificar en el diseño los puntos críticos de exposición y justificar cómo la arquitectura propuesta mitiga posibles vulnerabilidades.

- Formato de Entrega de los prácticos: Presentar informe técnico, incluyendo toda la documentación técnica y, opcionalmente, el prototipo de implementación para su validación en entorno de ejecución.

VIII - Regimen de Aprobación

Condiciones de regularización:
- Aprobación de los Trabajos Prácticos.
- Presentación y aprobación de los proyectos de Laboratorio.
- Asistencia mínima de 75% de las clases Teóricas/Prácticas y de Laboratorio.
- Superar 5 Temáticas Teórico/prácticas en un porcentaje mayor o igual al 70% (setenta por ciento) o sus correspondientes recuperaciones.
- Se otorgan, tal como lo expresa la reglamentación vigente, dos recuperaciones para cada temática.

Condiciones de aprobación:
- Por promoción. los estudiantes deben:
-- Contar con las condiciones de regularización.
-- Haber superado además cada Temática en un 80% (ochenta por ciento) o más.
-- Contar con una asistencia mínima de 80% de las clases Teórico/Prácticas y de Laboratorio y
-- Aprobar un coloquio Integrador.
-- Examen Final, el cual podrá ser oral o escrito.
Condiciones de estudiantes libres no se admiten debido a prácticos y exposiciones.

IX - Bibliografía Básica

[1] Arquitectura y Fundamentos (Eje de la Materia) Bass, L., Clements, P., y Kazman, R. (2021). Software architecture in practice (4a ed.). Addison-Wesley Professional.
[2] Cervantes, H., y Kazman, R. (2016). Designing software architectures: A practical approach. Addison-Wesley.
[3] Martin, R. C. (2017). Clean architecture: A craftsman's guide to software structure and design. Prentice Hall.
[4] Buschmann, F., Meunier, R., Rohnert, H., Sommerlad, P., y Stal, M. (2007). Pattern-oriented software architecture: A system of patterns (Vol. 1). Wiley.
[5] Fowler, M. (2002). Patterns of enterprise application architecture. Addison-Wesley.
[6] Gamma, E., Helm, R., Johnson, R., y Vlissides, J. (1994). Design patterns: Elements of reusable object-oriented software. Addison-Wesley.
[7] Castillo, C. (2022). Desarrollo de aplicaciones web con Jakarta EE. Edición Independiente.
[8] Heffelfinger, D. R. (2019). Jakarta EE 8 cookbook. Packt Publishing.
[9] Richard, T. (2023). Jakarta EE: Desarrolle aplicaciones web en Java (Servlets, JSP, JSF, REST, WebSockets). Ediciones Eni.
[10] Aedo, I., y Montero, S. (2005). Ingeniería de la Web y patrones de diseño. Pearson Educación.
[11] Pressman, R. S., y Maxim, B. R. (2021). Ingeniería del software: Un enfoque práctico (9a ed.). McGraw-Hill.
[12] Apuntes creados especificamente para la materia por el profesor responsable a cargo.

X - Bibliografia Complementaria

[1] Unified Modeling Language User Guide - Grady Booch, James Rumbaugh and Ivar Jacobson.
[2] Design Patterns: Elements of Reusable Object-Oriented Software - Erich Gamma, Richard Helm, Ralph Johnson, John.
[3] The J2EE(TM) 1.4 Tutorial - http://java.sun.com/j2ee/1.4/docs/tutorial/doc/.
[4] Object-Oriented Software Engineering: A Use Case Driven Approach - Ivar Jacobson.
[5] Unified Modeling Language -http://www.omg.org/technology/documents/formal/uml.htm

XI - Resumen de Objetivos

El objetivo de la materia es que los estudiantes tengan la capacidad de liderar el desarrollo de ecosistemas digitales complejos bajo tres pilares:
(1). Diseño Arquitectural Estratégico: Posicionar la arquitectura como la médula del desarrollo para garantizar sistemas escalables, seguros y sostenibles.
(2). Sistematización de Patrones: Superar el uso empírico mediante la selección y aplicación formal de Patrones de Diseño y Arquitecturales (GoF, MVC, Microservicios).
(3). Implementación Empresarial y Seguridad: Dominar el estándar Jakarta EE para construir soluciones robustas, con especial énfasis en la mitigación de vulnerabilidades en entornos de IoT y alta heterogeneidad.

XII - Resumen del Programa

El desarrollo de aplicaciones web contemporáneo se despliega en un ecosistema de alta complejidad, caracterizado por la convergencia de tecnologías heterogéneas y una evolución constante de la infraestructura de red. Este escenario ha desplazado el foco del desarrollo tradicional hacia el diseño arquitectural, posicionándolo como la médula estratégica que garantiza la escalabilidad, seguridad y sostenibilidad del software.
La materia Ingeniería web propone un abordaje integral donde el diseño arquitectural se ubica en el centro de la escena, considerándolo desde las etapas tempranas del ciclo de vida del software. A través de un recorrido que va desde los

fundamentos de los sistemas distribuidos hasta la implementación avanzada en Jakarta EE, el programa busca que los estudiantes no solo dominen herramientas técnicas, sino que desarrollen el criterio para seleccionar e integrar formalmente patrones arquitecturales y de diseño. Frente a los desafíos actuales de los Escenarios Digitales y el Internet de las Cosas (IoT), la materia incorpora una visión crítica sobre la identificación de vulnerabilidades y la aplicación de soluciones estandarizadas. De este modo, se busca formar profesionales capaces de construir arquitecturas web robustas y eficientes, preparándolos para resolver problemas complejos en entornos de alta demanda tecnológica.

XIII - Imprevistos

Ante cualquier necesidad, se puede contactar con el equipo de cátedra a través de la siguiente dirección de correo electrónico:
Prof. Mg. Ing. Norma Beatriz Perez: nbperez@email.unsl.edu.ar

XIV - Otros

-