



Campus de la UNA
SAN LORENZO-PARAGUAY

UNIVERSIDAD NACIONAL DE ASUNCIÓN
FACULTAD POLITÉCNICA
CONSEJO DIRECTIVO

RESOLUCIÓN 25/18/129-00
ACTA 1226/25/08/2025

“POR LA CUAL SE APRUEBA EL PROGRAMA DE ESTUDIO DE LA ASIGNATURA ARQUITECTURA DE SOFTWARE, DE CARRERAS DE GRADO, SEDE SAN LORENZO FILIAL VILLARRICA Y FILIAL CORONEL OVIEDO”

VISTO: El Memorando DA/1707/2025 del Director Académico de la FP-UNA, Prof. MSc. Felipe Santiago Uzabal Escurra, con el cual remite el Memorando CCPTCC/030/2025 de la Comisión Coordinadora del Proyecto de Transformación Curricular de Carreras de Grado de la FP-UNA, en el que presenta la propuesta de Programa de Estudio de Asignatura de las Carreras de Grado.

CONSIDERANDO: La Ley 4995/2013 de Educación Superior, el Estatuto de la Universidad Nacional de Asunción y las deliberaciones sobre el tema.

Que la Comisión Coordinadora del Proyecto de Transformación Curricular de Carreras de Grado, solicita la aprobación del Programa de Estudio de la asignatura **“Arquitectura de Software”**, la cual es común entre Carreras de Grado.

Que los programas fueron elaborados conforme a las disposiciones establecidas por el Consejo Nacional de Educación Superior (CONES) en materia de **créditos académicos**, según lo dispuesto en la Resolución CONES N.º 221/2024, que regula el *Sistema de Créditos Académicos – Paraguay* y los criterios para su publicación en las carreras de grado.

**EL CONSEJO DIRECTIVO DE LA FACULTAD POLITÉCNICA
RESUELVE:**

25/18/129-01 APROBAR el Programa de Estudio de la Asignatura **“Arquitectura de Software”**, de carreras de Grado, Sede San Lorenzo, Filial Villarrica y Filial Coronel Oviedo, detallado en el ANEXO 91 de la presente Acta.

25/18/129-02 COMUNICAR, copiar y archivar.

Prof. Abg. Joel Arsenio Benítez Santacruz
Secretario



Prof. Ing. Silvia Teresa Leiva León, MSc.
Presidenta



Campus de la UNA
SAN LORENZO-PARAGUAY

UNIVERSIDAD NACIONAL DE ASUNCIÓN
FACULTAD POLITÉCNICA
CONSEJO DIRECTIVO

Resolución 25/18/129-00 Acta 1226/25/08/2025
ANEXO 91

DEPARTAMENTO DE ENSEÑANZA DE INFORMÁTICA
PROGRAMA DE ESTUDIO

I. IDENTIFICACIÓN

Nivel				Grado							
Asignatura				Arquitectura de Software							
Carrera				Plan		Sede/Filial		Carácter		Semestre	Prerrequisitos
Ingeniería en Informática				2026		Sede San Lorenzo		Electiva 1		Séptimo	Ingeniería de Software 1, Bases de Datos 2, Programación de Aplicaciones 2, Seguridad Informática.
Licenciatura en Ciencias Informáticas				2026		Sede San Lorenzo/Filial Villarrica/Filial Coronel Oviedo		Electiva		***	Ingeniería de Software I, Base de Datos II, Seguridad Informática, Programación Frontend.
Semanal					Periodo						
HT	HP	HTD	HTI	HS	PA	THTD	THTI	THA	CA-PY		
2	2	4	4	8	18	72	72	144	5		

- *HT: Horas Teóricas semanales.
- *HP: Horas Prácticas semanales.
- *HTD: Horas semanales de Trabajo académico con acompañamiento Docente.
- *HTI: Horas semanales de Trabajo académico Independiente del estudiante.
- *HS: Horas Semanales (HTD+HTI).
- *PA: Periodo Académico en semanas.
- * THTD: Total de Horas de Trabajo académico con acompañamiento Docente (HTD*PA).
- * THTI: Total de Horas de Trabajo académico Independiente del estudiante (HTI*PA).
- * THA: Total de Horas de trabajo Académico (THTD+THTI).
- * CA-PY: Créditos académicos de la asignatura.

II. FUNDAMENTACIÓN

La misión de la arquitectura de software es aplicar soluciones de alto nivel para una tipificación de problemas que comúnmente se presentan en el desarrollo de software. Es importante que el futuro profesional de la informática maneje los conceptos relacionados con las diferentes arquitecturas de referencia, los patrones de diseño, atributos de calidad del software, patrones de datos distribuidos para una arquitectura de micro servicios y también orientada a eventos.

No se espera solamente la implementación del software como tal, sino la optimización de recursos tanto de hardware como de software, tener un *time to market* acorde a las necesidades del negocio y la administración del presupuesto disponible.

La incorporación de Arquitectura de Software en la carrera obedece al propósito de ayudar al futuro profesional en la concreción de sus conocimientos técnicos, adquiriendo capacidades de análisis, diseño e implementación de arquitecturas de software que maximicen las características del software y aprovechen

aprovechen los recursos disponibles para la gestión de todo el ciclo de vida, según los retos técnicos que el mercado competitivo, dinámico y global exige.

En relación a la naturaleza de la asignatura, se aborda de manera teórico-práctica, se combinarán conceptos teóricos con ejercicios prácticos. La organización de la asignatura se basa en los ejes temáticos, se incluyen conceptos fundamentales como: diseño de sistemas; patrones de arquitectura; microservicios; arquitecturas orientadas a eventos.

III. COMPETENCIAS DEL PERFIL DE EGRESO ASOCIADAS

- 1. Seleccionar, utilizar y construir instrumentos innovadores asociados al ejercicio de la informática.
- 2. Planificar, proyectar, diseñar y ejecutar proyectos sostenibles e integrales para la resolución de problemas, la mejora y la innovación en el ámbito de la informática.

IV. ORGANIZACIÓN DE LA ASIGNATURA

Unidades	Contenidos	Resultados de aprendizaje
1. Introducción.	1.1. Qué es una arquitectura de software. 1.2. Objetivos y beneficios de una arquitectura de software. 1.3. Tareas y responsabilidades de un arquitecto de software.	1. Identifica, entiende y documenta los requerimientos de sistemas de información. 2. Identifica las oportunidades creadas por las innovaciones tecnológicas. 3. Aprecia las relaciones entre la estrategia de negocio y los sistemas de información, arquitectura e infraestructura.
2. Atributos de calidad.	2.1. Introducción a los atributos de calidad. 2.2. Disponibilidad. 2.3. Performance. 2.4. Testability. 2.5. Seguridad. 2.6. Usabilidad.	1. Comprende los principales atributos de calidad en el software, y su impacto en el diseño, desarrollo y mantenimiento de sistemas. 2. Evalúa y aplica prácticas y estrategias para garantizar atributos de calidad específicos, adaptando soluciones a las necesidades del sistema y las expectativas de los usuarios. 3. Desarrolla soluciones de software que integren de manera equilibrada los atributos de calidad, utilizando técnicas y herramientas apropiadas para monitorear, medir y mejorar continuamente aspectos clave.
3. Diseño y desarrollo de una arquitectura de software.	3.1. Enfoques para el desarrollo de una arquitectura de software. 3.2. Factores y restricciones. 3.3. Conceptos transversales. 3.4. Patrones. 3.5. Principios de diseño.	1. Comprende los enfoques, principios y metodologías para el desarrollo de una arquitectura de software, analizando cómo los factores y restricciones del sistema influyen en las decisiones de diseño y en la selección de patrones arquitectónicos adecuados. 2. Identifica y aplica los patrones de arquitectura de software más comunes para resolver problemas

Unidades	Contenidos	Resultados de aprendizaje
		específicos de diseño, optimizando la modularidad, flexibilidad y mantenimiento del sistema. 3. Aplica principios de diseño sólidos y conceptos transversales en la construcción de arquitecturas de software, garantizando la coherencia, la reutilización y la adaptabilidad del sistema ante cambios y futuras evoluciones tecnológicas.
4. Arquitectura de microservicios.	4.1. Qué es una arquitectura de microservicios. 4.2. Comunicación interprocesos. 4.3. Manejo de transacciones en una arquitectura de microservicios. 4.4. Implementación de consultas en una arquitectura de microservicios.	1. Comprende los principios fundamentales y los beneficios de la arquitectura de microservicios. 2. Aplica diferentes estrategias de comunicación interprocesos, para garantizar la integración eficiente y fiable entre microservicios en un sistema distribuido. 3. Implementa soluciones para gestionar transacciones distribuidas y consultas eficientes en una arquitectura de microservicios.
5. Arquitectura orientada a eventos.	5.1. ¿Qué es una arquitectura orientada a eventos? 5.2. Eventos. 5.3. Streams. 5.4. Kafka. 5.5. MessageQueue.	1. Comprende los principios fundamentales de una arquitectura orientada a eventos (EDA), incluyendo sus componentes clave, como eventos, productores, consumidores y la infraestructura de mensajería, y cómo estos se aplican para construir sistemas escalables y desacoplados. 2. Analiza y aplica el uso de eventos, streams y sistemas de mensajería para facilitar la comunicación asíncrona y la integración entre servicios en una arquitectura orientada a eventos. 3. Implementa soluciones prácticas para gestionar el flujo de eventos y datos en tiempo real.

V. ESTRATEGIAS DIDÁCTICAS

En el desarrollo del programa se aplicarán estrategias didácticas conducentes a la apropiación teórica y la ejecución práctica de procesos y procedimientos, a saber:

- **Debate:** exposición por parte del docente de los conceptos básicos por unidad, con materiales de lectura y ejemplos orientados a la enseñanza de las competencias específicas de la asignatura. El docente asume el rol de expositor y buscará generar el debate a través de preguntas sobre lo expuesto y desde la participación de los estudiantes.
- **Clase invertida:** con materiales didácticos dispuestos en el aula virtual previamente y aplicar en clases presenciales, analizando y respondiendo a planteamientos con estudio de casos a través de trabajos grupales, orientadas especialmente al desarrollo de métodos para la organización, tales como: la definición de la misión, la visión, la elaboración de funciones, objetivos, organigramas y planificación de trabajos colaborativos, para asimilar los contenidos de la asignatura.
- **Ponencia docente:** el docente introducirá el tema, estableciendo su importancia y relevancia para la materia. Desarrollará el contenido de manera secuencial, explicando conceptos, resaltando puntos clave y relacionando la información con contextos prácticos o ejemplos reales. Utilizará recursos visuales y auditivos para enriquecer y variar la exposición, facilitando la comprensión de los estudiantes. El docente fomentará la participación activa de los estudiantes a través de preguntas, debates cortos o ejercicios de reflexión en determinados momentos de la exposición.
- **Aprendizaje basado en problemas:** las clases prácticas serán utilizadas para que los estudiantes resuelvan diversos ejercicios de diseño de arquitecturas utilizando los conceptos aprendidos. Esto los anima a aplicar lo que han aprendido de manera práctica y a enfrentarse a desafíos reales que podrían encontrar en el campo de la arquitectura de sistemas.

La elección particular de la estrategia didáctica aplicada será explícita en Planeamiento de la Asignatura, de acuerdo con el perfil de los estudiantes, los recursos disponibles y el contexto educativo.

VI. ESTRATEGIAS EVALUATIVAS

Procesos de análisis y diseño de arquitecturas grupales e individuales, pruebas individuales orales y/o escritas durante el desarrollo de las unidades con diálogos e interpretaciones que los estudiantes realicen sobre los contenidos, debates, retroalimentación en casos necesarios y actividades que amplíen el conocimiento.

Con fines de calificación y promoción se aplicará el Reglamento Académico vigente en la institución que prevé valoraciones de proceso y final.

VII. MEDIOS AUXILIARES

Pizarras, marcadores, computadoras, proyector, parlantes para multimedia, plataforma virtual, sala de laboratorio equipada para prácticas con sistemas operativos Linux o Windows y acceso a Internet.

VIII. BIBLIOGRAFÍA

- Bass, L., Clements, P., & Kazman, R. (2022). Software Architecture in Practice. Carnegie Mellon University.
- Indrasiri, K., & Siriwardena, P. (2018). Microservices for the Enterprise. Apress.
- Martin, R. C. (2017). Clean Architecture. Prentice Hall.
- Richardson, C. (2018). Microservices Patterns: With examples in Java. Manning.
- Starke, G., & Lorz, A. (2021). Software Architecture Foundation. Van Haren Publishing.
- Volkhover, A (2019). Become an awesome Software Architect
- Biehl, M (2016). API Design. API-University Press

