



Campus de la UNA  
SAN LORENZO-PARAGUAY

UNIVERSIDAD NACIONAL DE ASUNCIÓN  
FACULTAD POLITÉCNICA  
CONSEJO DIRECTIVO

RESOLUCIÓN 24/26/83-00  
ACTA 1208/16/12/2024

**“POR LA CUAL SE APRUEBA EL PROGRAMA DE ESTUDIO DE LA ASIGNATURA DESARROLLO AVANZADO II – FULL STACK, DE LA CARRERA LICENCIATURA EN CIENCIAS INFORMÁTICAS – PLAN 2023 DE LA FP-UNA”**

**VISTO:** El Memorando DA/2437/2024 del Director Académico de la FP-UNA, Prof. MSc. Felipe Santiago Uzabal Ecurra, con el cual remite el Memorando CCPTCC/036/2024 de la Comisión Coordinadora del Proyecto de Transformación Curricular de Carreras de Grado de la FP-UNA, en el que presenta la propuesta de Programas de Estudio de las Asignaturas de la Carrera Licenciatura en Ciencias Informáticas.

**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 **“Desarrollo Avanzado II – Full Stack”**, de la carrera Licenciatura en Ciencias Informáticas – Plan 2023, cuyo plan de estudio ya fue aprobado por el Consejo Superior Universitario.

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

**24/26/83-01** APROBAR el Programa de Estudio de la Asignatura **“Desarrollo Avanzado II – Full Stack”**, de la carrera Licenciatura en Ciencias Informáticas – Plan 2023 de la FP-UNA, detallado en el ANEXO 75 de la presente Acta.

**24/26/83-02** 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 24/26/83-00 Acta 1208/16/12/2024  
ANEXO 75

DEPARTAMENTO DE ENSEÑANZA DE INFORMÁTICA  
PROGRAMA DE ESTUDIO

I. IDENTIFICACIÓN

Asignatura	Desarrollo Avanzado II – Full Stack				
Carrera	Plan	Sede/Filial	Carácter	Semestre	Prerrequisitos
Licenciatura en Ciencias Informáticas	2023	Sede San Lorenzo / Filial Villarrica / Filial Coronel Oviedo	Electiva	***	Haber acumulado la cantidad de créditos académicos que corresponda a la aprobación de todas las asignaturas hasta el 5° semestre, resultante de la aplicación del Sistema Nacional de Créditos Académicos– Paraguay en la UNA.
Horas semanales	4				
Total de horas teóricas semestral	36				
Total de horas prácticas semestral	36				
Total de horas semestral	72				
Valor en créditos académicos	La valoración en créditos académicos será comunicada en su oportunidad ajustada al Reglamento General del Sistema de Créditos Académicos de la UNA, el cual se encuentra en proceso de elaboración conforme a las disposiciones de la Resolución CONES N° 221/2024, en su artículo N° 10.				
Actualización	Al egreso de la primera cohorte.				

II. FUNDAMENTACIÓN

En un entorno donde las aplicaciones web son cada vez más complejas y la demanda por soluciones escalables y eficientes es creciente, los desarrolladores full stack son clave para integrar frontend y backend, garantizando un flujo de trabajo ágil y eficaz. Este curso proporciona a los estudiantes las habilidades necesarias para diseñar, implementar, probar y documentar sistemas completos, aplicando las últimas tecnologías tanto en frontend como en backend. Además, se enfoca en la utilización de metodologías ágiles, técnicas de gestión de proyectos, y herramientas de colaboración para desarrollar software de calidad que satisfaga las necesidades actuales del mercado.

En relación a la naturaleza de la asignatura, se aborda de manera teórico-práctico, se combinarán conceptos teóricos con ejercicios prácticos. La organización de la asignatura se basa en los ejes temáticos, se incluyen conceptos fundamentales como: Diseño, implementación, prueba y documentación de un sistema de software, aplicando metodologías de desarrollo de software, técnicas de gestión de proyectos y herramientas de colaboración. Tecnologías de última generación para desarrollo backend y frontend.

III. COMPETENCIAS DEL PERFIL DE EGRESO ASOCIADAS

1. Comunicarse en las lenguas oficiales del país y en una lengua extranjera.

2. Liderar y trabajar en equipo con eficacia y responsabilidad tomando decisiones basadas en evidencias.
3. Aplicar en la práctica profesional los valores humanos, la ética y los mecanismos de seguridad laboral.
4. Actuar proactivamente frente a los problemas sociales y ambientales.
5. Adaptarse respetuosamente a contextos nuevos o adversos, así como a diversidades personales, disciplinares y culturales.
6. Actualizarse permanentemente mediante la obtención y gestión autónoma de información de calidad, utilizando tecnología de la información y comunicación.
7. Evaluar el comportamiento de diversos fenómenos disciplinares e interdisciplinares relacionados con la ingeniería en informática con una visión de sistema, mediante modelos teóricos validados y actualizados, capaces de abarcarlos integralmente en un contexto de incertidumbre.
8. Seleccionar, utilizar y construir instrumentos innovadores asociados al ejercicio de la ingeniería informática.
9. Aplicar, producir y difundir conocimientos técnicos y científicos en el área de la ingeniería informática.
10. Planificar, proyectar, diseñar y ejecutar proyectos sostenibles e integrales para la resolución de problemas, la mejora y la innovación en el ámbito de la ingeniería informática.
11. Interpretar, modelar y comunicar información referida a la ingeniería en informática en forma gráfica.

**IV. ORGANIZACIÓN DE LA ASIGNATURA**

Unidades	Contenidos	Resultados de aprendizaje
1. Metodologías de Desarrollo de Software y Gestión de Proyectos.	1.1. Introducción a las metodologías ágiles. 1.2. Técnicas de gestión de proyectos y herramientas de colaboración. 1.3. Planificación y seguimiento de proyectos de software. 1.4. Gestión de equipos de desarrollo en proyectos full stack.	1. Aplica metodologías ágiles y técnicas de gestión de proyectos en el desarrollo de software, para asegurar el cumplimiento de objetivos y plazos. 2. Utiliza herramientas de colaboración para coordinar el trabajo en equipo y gestionar eficientemente el ciclo de vida de un proyecto de software.
2. Desarrollo Frontend con Tecnologías Modernas	2.1. Tecnologías y frameworksfrontend. 2.2. Diseño de interfaces de usuario intuitivas y responsivas. 2.3. Componentización y reutilización de código en frontend. 2.4. Acceso a servicios y APIs desde el frontend.	1. Diseña e implementa interfaces de usuario eficientes y adaptativas utilizando frameworksfrontend modernos. 2. Desarrolla componentes reutilizables para garantizar la modularidad y mantenibilidad del frontend.
3. Desarrollo Backend con Tecnologías Modernas	3.1. Frameworksbackend. 3.2. Diseño de arquitecturas backend escalables. 3.3. Gestión de bases de datos relacionales y no relacionales. 3.4. Creación de APIsRESTful.	1. Implementa arquitecturas backend utilizando frameworks modernos, asegurando escalabilidad y eficiencia. 2. Diseña y desarrolla APIs que permitan la comunicación eficaz



*[Handwritten signature]*

*[Handwritten signature]*

Unidades	Contenidos	Resultados de aprendizaje
		entre el frontend y el backend.
4. Integración y Pruebas en el Desarrollo Full Stack.	4.1. Integración continua y despliegue continuo (CI/CD). 4.2. Pruebas unitarias, de integración y funcionales para frontend y backend. 4.3. Automatización de pruebas en entornos full stack. 4.4. Seguridad en aplicaciones web full stack (autenticación, autorización, cifrado).	1. Implementa procesos de integración continua para automatizar las pruebas y despliegue de aplicaciones full stack. 2. Aplica pruebas automatizadas y manuales en diferentes niveles del sistema para asegurar la calidad y seguridad del software.
5. Documentación y Mantenimiento de Sistemas Full Stack.	5.1. Mejores prácticas en la documentación de software. 5.2. Generación de documentación técnica y de usuario final. 5.3. Mantenimiento y evolución de sistemas de software. 5.4. Refactorización y optimización del código.	1. Documenta adecuadamente los componentes de un sistema full stack, proporcionando guías claras tanto para desarrolladores como para usuarios. 2. Implementa técnicas de refactorización y optimización del código, manteniendo la calidad y eficiencia del sistema a lo largo del tiempo.

## 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:

- **Aprendizaje basado en problemas:** estrategia de enseñanza donde se busca resolver un problema a través del conocimiento que adquirió en el aula, el estudiante toma liderazgo de su aprendizaje e identifica la importancia de su aprendizaje y el conocimiento.
- **Aprendizaje basado en proyectos:** metodología donde el estudiante participa activamente en su aprendizaje, desarrollando diferentes habilidades para solucionar un problema a través de un proyecto, y que pueda implementarse para la mejora del contexto.
- **Aprendizaje cooperativo colaborativo:** conjunto de métodos de instrucción para la aplicación en grupos pequeños, de entrenamiento y desarrollo de habilidades mixtas (aprendizaje, desarrollo personal y social) donde cada componente del grupo es responsable tanto de su aprendizaje como del de los restantes.
- **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.

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

## VI. ESTRATEGIAS EVALUATIVAS

Procesos de producción 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 de Evaluación vigente en la institución que prevé valoraciones de proceso y final.

## VII. MEDIOS AUXILIARES

Aula o plataforma virtual, pizarras acrílicas, proyector, marcadores, borrador de pizarra acrílica, equipo de audio, ordenadores, wifi, celulares, plataformas de videoconferencia y foros.

## VIII. BIBLIOGRAFÍA

- Erich Gamma, Richard Helm, Ralph Johnson, John Vlissides. Design Patterns: Elements of Reusable Object-Oriented Software. 2022, 2da edición, Addison-Wesley.
- Sam Newman. Building Microservices: Designing Fine-Grained Systems. 2022, 2da edición, O'Reilly Media.
- Robert C. Martin. Clean Code: A Handbook of Agile Software Craftsmanship. 2022, 2da edición, Prentice Hall.
- Steve McConnell. Code Complete: A Practical Handbook of Software Construction. 2023, 2da edición, Microsoft Press.
- Kurt Guntheroth. Optimized C++: Proven Techniques for Heightened Performance. 2022, 1ra edición, O'Reilly Media.
- Jez Humble, David Farley. Continuous Delivery: Reliable Software Releases through Build, Test, and Deployment Automation. 2023, 1ra edición, Addison-Wesley.

