



Campus de la UNA
SAN LORENZO-PARAGUAY

UNIVERSIDAD NACIONAL DE ASUNCIÓN
FACULTAD POLITÉCNICA
CONSEJO DIRECTIVO

RESOLUCIÓN 25/18/94-00
ACTA 1226/25/08/2025

“POR LA CUAL SE APRUEBA EL PROGRAMA DE ESTUDIO DE LA ASIGNATURA LENGUAJES DE PROGRAMACIÓN 3, DE LA CARRERA INGENIERÍA EN INFORMÁTICA – PLAN 2026, SEDE SAN LORENZO”

VISTO: El Memorando DA/1707/2025 del Director Académico de la FP-UNA, Prof. MSc. Felipe Santiago Uzabal Ecurra, 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 Programas de Estudio de las Asignaturas de la Carrera Ingeniería en Informática.

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 **“Lenguajes de Programación 3”**, de la carrera Ingeniería en Informática – Plan 2026.

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:**

25/18/94-01 APROBAR el Programa de Estudio de la Asignatura **“Lenguajes de Programación 3”**, de la carrera Ingeniería en Informática – Plan 2026, Sede San Lorenzo, detallado en el ANEXO 56 de la presente Acta.

25/18/94-02 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

Resolución 25/18/94-00 Acta 1226/25/08/2025
ANEXO 56

DEPARTAMENTO DE ENSEÑANZA DE INFORMÁTICA
PROGRAMA DE ESTUDIO

I. IDENTIFICACIÓN

Nivel			Grado							
Asignatura			Lenguajes de Programación 3							
Carrera			Plan		Sede/Filial		Carácter	Semestre	Prerrequisitos	
Ingeniería en Informática			2026		Sede San Lorenzo		Obligatoria	Tercero	Algoritmos y Estructuras de Datos 2, Lenguajes de Programación 2.	
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

Existe una necesidad de crear componentes de software que hagan uso eficiente de los recursos computacionales que son administrados por un sistema operativo y sus servicios. En la actualidad, los sistemas se han vuelto complejos y sus componentes no funcionan aisladamente sino deben hacerlo de manera coordinada compartiendo datos y utilizando las facilidades de comunicación brindadas.

En este contexto, comprender las implicancias de las decisiones de programación, basadas en las características del lenguaje de programación, la arquitectura del hardware y los servicios del software que los administra, es fundamental para un profesional de ingeniería informática.

Este curso acercará al estudiante a la programación analizando y utilizando un lenguaje de programación apropiado para desarrollar componentes de software que accedan a los sistemas de servicios de bajo nivel, interactuando de manera directa con un sistema operativo, ya sea general o especializado.

La asignatura tiene un enfoque teórico-práctico. Los ejes temáticos y las unidades están conectados, proporcionando una base sólida para la programación a bajo nivel y la interacción directa con el hardware y el sistema operativo de la siguiente forma:

- Programación de sistemas: se aborda en la Unidad 1, que introduce conceptos esenciales para desarrollar software que interactúa directamente con el hardware y el sistema operativo.
- Características de lenguajes de programación de bajo nivel: se vincula con la Unidad 2, que explora lenguajes específicos y su uso en la programación cercana al hardware.

- Gestión de memoria: tratada en la Unidad 3, en la cual se estudian las técnicas de control de memoria requeridas para la eficiencia y seguridad en programación de sistemas.
- Programación a nivel de bits e interacción con el sistema operativo: abordadas en la Unidad 4, en la cual se detalla cómo los programas interactúan y manipulan recursos del sistema operativo.
- Concurrencia: corresponde a la Unidad 5, en la cual se examina la ejecución simultánea de tareas para la programación de sistemas.
- Programación de sockets: directamente relacionada con la Unidad 6, que cubre la comunicación en red mediante la programación de sockets.

III. COMPETENCIAS DEL PERFIL DE EGRESO ASOCIADAS

1. Comunicarse en las lenguas oficiales del país y en una lengua extranjera.
2. Seleccionar, utilizar y construir instrumentos innovadores asociados al ejercicio de la ingeniería informática.
3. 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.

IV. ORGANIZACIÓN DE LA ASIGNATURA

Unidades	Contenidos	Resultados de aprendizaje
1. Programación de sistemas.	1.1 Introducción. 1.2 Sistema computacional, organización del hardware y abstracciones del sistema operativo. 1.3 Representación y manipulación de información de enteros y flotantes. 1.4 POSIX estandarización y características. 1.5 Ley de Amdahl.	1. Identifica los componentes de un sistema computacional (hardware y software), describiendo su propósito y explicando su funcionamiento en una computadora actual. 2. Describe las abstracciones en un sistema operativo actual. 3. Identifica las representaciones internas de datos enteros y flotantes en una computadora. 4. Enuncia las características y aplicaciones del estándar POSIX. 5. Utiliza la Ley de Amdahl en situaciones específicas de análisis de rendimiento.
2. Lenguaje de bajo nivel.	2.1. Introducción. 2.2. Compilación, enlazado y depuración. 2.3. Tipo de datos y operadores. 2.4. Estructuras de control. 2.5. Estructuras de datos incorporadas. 2.6. Modelo de memoria. 2.7. Punteros y/o referencias. Representación y aritmética. 2.8. Interacción con instrucciones de bajo nivel (ensamblador). 2.9. Código generado y rendimiento.	1. Utiliza herramientas de construcción y depuración de programas en un lenguaje de bajo nivel. 2. Implementa correctamente programas generales en un lenguaje de bajo nivel. 3. Identifica las interacciones de las implementaciones de alto nivel con las instrucciones de bajo nivel a nivel de ensamblador. 4. Identifica las implicancias de rendimiento del uso de ciertas construcciones del lenguaje sobre

Unidades	Contenidos	Resultados de aprendizaje
		el código binario generado.
3. Gestión de memoria.	3.1 Introducción. 3.2 Memoria física y virtual. 3.3 Direccionamiento. 3.4 Asignación de memoria dinámica 3.5 Garbagecollector 3.6 Proyecto de programación.	1. Describe la organización de la memoria en una computadora moderna y su relación con las construcciones presentes en los lenguajes de programación. 2. Implementa programas en lenguaje de bajo nivel que permitan la utilización de los servicios de gestión de memoria.
4. Interacción con el Sistema Operativo (S.O.).	4.1 Excepciones. 4.2 Procesos. Creación y control. Señales. 4.3 Entrada/Salida. Dispositivos. Archivos y directorios. 4.4 Proyecto de programación	1. Describe las excepciones y su gestión en el contexto de la ejecución de un proceso. 2. Explica el sentido y propósito del control de procesos que utiliza el S.O. para la abstracción de procesos. 3. Describe los conceptos de dispositivos de Entrada/Salida, archivos y directorios. 4. Implementa correctamente programas en lenguaje de bajo nivel que crea, gestiona y finaliza procesos en un S.O. 5. Implementa correctamente programas en lenguaje de bajo nivel que utilizan servicios de Entrada/Salida y archivos.
5. Concurrencia.	5.1 Introducción. 5.2 Concurrencia con procesos. Pipes. IPC. 5.3 Hilos y sincronización. 5.4 Secciones críticas, semáforos y mutexes. 5.5 Proyecto de programación	1. Describe el funcionamiento de semáforos, mutexes y otras primitivas de sincronización. 2. Describe los problemas relacionados a la ejecución concurrente de múltiples procesos o hilos. 3. Implementa correctamente programas en lenguaje de nivel que crea, gestiona y finaliza procesos concurrentes utilizando hilos o IPC.
6. Comunicación socket.	6.1 Introducción. 6.2 Modelo Cliente/Servidor. 6.3 Internet. Modelo de direccionamiento. 6.4 Interface Socket (bind, listen, accept). 6.5 Proyecto de programación.	1. Describe correctamente el modelo y funcionamiento de la comunicación con sockets. 2. Implementa correctamente programas en lenguaje de nivel que permitan la comunicación vía sockets.

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:** 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 estudiante buscará resolver un problema a través del conocimiento que adquirió en el aula.
- **Aprendizaje basado en proyectos:** el docente propondrá la realización de proyectos que involucren los resultados de aprendizaje en las unidades que así se establezcan. De esta forma el estudiante participa activamente en su aprendizaje, desarrollando diferentes habilidades para solucionar un problema a través de este proyecto.

La elección particular de la estrategia didáctica aplicada será explícita en el Planeamiento de la Asignatura, 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, implementación de proyectos de programación y presentaciones de los hallazgos y experiencias. Todos estos serán valorados y que en su conjunto aportarán para la calificación y promoción, las que serán aplicadas según normativas institucionales.

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

Aula virtual, pizarrón, proyector, marcadores, equipo de audio, computadora.

VIII. BIBLIOGRAFÍA

- R. Bryant y David R. O'Hallaron (2016). ComputerSystems. A programmer'sperspective. 3ra. Edición. Pearson.
- W. S. Stevens y S. Rago (2013) AdvancedProgramming in the UNIX Environment. 3ra Edicion. Pearson Education.
- J. M. Hart (2010). Windows SystemProgramming. 4ta Edicion. Addison-Wesley.
- Kay A. Robbins, Steven Robbins (2003) UNIX™ SystemsProgramming: Communication, Concurrency, and Threads.

