



Campus de la UNA
SAN LORENZO-PARAGUAY

UNIVERSIDAD NACIONAL DE ASUNCIÓN
FACULTAD POLITÉCNICA
CONSEJO DIRECTIVO

RESOLUCIÓN 24/25/23-00
ACTA 1207/09/12/2024

“POR LA CUAL SE APRUEBA EL PROGRAMA DE ESTUDIO DE LA ASIGNATURA LENGUAJES DE PROGRAMACIÓN 1, DE LA CARRERA INGENIERÍA EN INFORMÁTICA – PLAN 2023 DE LA FP-UNA”

VISTO: El Memorando DA/2379/2024 del Director Académico de la FP-UNA, Prof. MSc. Felipe Santiago Uzabal Escurra, con el cual remite el Memorando CCPTCC/034/2024 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 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 1”, de la carrera Ingeniería en Informática – Plan 2023, cuyo plan de estudio ya fue aprobado por el Consejo Superior Universitario.

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

24/25/23-01 APROBAR el Programa de Estudio de la Asignatura “Lenguajes de Programación 1”, de la carrera Ingeniería en Informática – Plan 2023 de la FP-UNA, detallado en el ANEXO 09 de la presente Acta.

24/25/23-02 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 24/25/23-00 Acta 1207/09/12/2024
ANEXO 09

DEPARTAMENTO DE ENSEÑANZA DE INFORMATICA
PROGRAMA DE ESTUDIO

I. IDENTIFICACIÓN

Asignatura	Lenguajes de Programación 1				
Carrera	Plan	Sede/Filial	Carácter	Semestre	Prerrequisitos
Ingeniería en Informática	2023	Sede San Lorenzo	Obligatoria	Primero	Ninguno
Horas semanales	4				
Total de horas teóricas semestral	30				
Total de horas prácticas semestral	42				
Total de horas semestral	72				
Valor en créditos académicos	La valoración en créditos académicos será comunicada en su oportunidad, ajustada al reglamento para la aplicación del Sistema Nacional de Créditos Académicos – Paraguay en la UNA; ajuste que 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

Los lenguajes de programación son componentes esenciales en el mundo de la informática. Proporcionan las herramientas y la sintaxis necesarias para comunicar instrucciones a una computadora y lograr que realice tareas específicas. La comprensión profunda de los lenguajes de programación es fundamental para cualquier persona que desee trabajar en el campo de la programación y el desarrollo de software.

La asignatura Lenguaje de Programación 1 se enfoca en brindar a los estudiantes las habilidades y los conocimientos necesarios para comprender y utilizar lenguajes de programación de manera efectiva. A lo largo de la asignatura, los estudiantes desarrollarán habilidades de resolución de problemas, pensamiento lógico y programación que les permitirán enfrentar desafíos del mundo real en el desarrollo de software y la automatización de procesos. Además, se fomentará la creatividad y la innovación en la programación, preparando a los estudiantes para un futuro exitoso en el campo de la tecnología de la información.

La asignatura tiene un enfoque teórico-práctico. La relación entre los ejes temáticos y las unidades es como sigue:

- Lenguajes de programación, sintaxis y semántica: se vincula con las Unidades 1, 2, 3, 11 y 13 que cubren la historia y evolución de los lenguajes de programación, sintaxis de un lenguaje de

programación, las estructuras de control del programa, manejo de programas por consola y compilación.

- Estructuras de control: están directamente relacionadas con la Unidad 3, que se centra en las estructuras de control del programa.
- Variables y tipos de datos, simples y estructurados: se reflejan en las Unidades 4, 5, 7, 9, 10 y 12 que abordan arreglos, datos básicos y agregados, punteros, cadenas, estructuras de datos y archivos.
- Funciones y procedimientos: corresponden a la Unidad 6 y 8, dedicada a funciones y modularidad.
- Gestión de memoria dinámica para tipos de datos: se trata en la Unidad 14, que aborda la gestión de memoria dinámica.

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. 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.
4. 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. Introducción.	1.1. Breve historia de los lenguajes de programación. 1.2. Lenguaje de nivel medio y estructurado. 1.3. Compiladores e intérpretes 1.4. Partes de un programa. 1.5. Bibliotecas. 1.6. Compilación de un programa.	1. Explica la evolución de los lenguajes de programación a lo largo de la historia. 2. Diferencia entre lenguajes de programación de nivel medio y lenguajes estructurados. 3. Diferencia entre compiladores e intérpretes y explicar cómo funcionan.
2. Sintaxis de un lenguaje de programación.	2.1. Los tipos de datos básicos. 2.2. Modificaciones de identificadores. 2.3. Nombres de identificadores. 2.4. Variables. 2.5. Modificadores del tipo de acceso. 2.6. Especificadores de clase de almacenamiento. 2.7. Inicializaciones de variables. 2.8. Constantes. 2.9. Operadores. 2.10. Expresiones.	1. Identifica los tipos de datos básicos en programación, como enteros, números de punto flotante, caracteres y booleanos. 2. Define variables en un programa y asignar nombres de identificadores siguiendo las mejores prácticas de nomenclatura. 3. Utiliza operadores matemáticos, lógicos y relacionales en combinación con variables y constantes para construir expresiones que realicen cálculos y tomen decisiones en un programa.
3. Estructuras de control de programa.	3.1. Verdadero y falso. 3.2. Sentencias de: 3.2.1. Iteración. 3.2.2. Salto. 3.2.3. Expresión.	1. Evalúa las declaraciones como verdaderas o falsas en un contexto de programación. 2. Aplica de manera efectiva diferentes tipos de sentencias de

Unidades	Contenidos	Resultados de aprendizaje
	3.2.4. Bloque.	control en un programa, incluyendo sentencias de iteración, salto, expresión y bloque, para controlar el flujo de ejecución de un programa de acuerdo con las necesidades específicas.
4. Arreglos y cadenas.	4.1. Arreglos unidimensionales. Generación de un puntero a un arreglo. 4.2. Paso de arreglos unidimensionales a funciones. 4.3. Cadenas. 4.4. Arreglos bidimensionales y multidimensionales a funciones. 4.5. Indexación de punteros. 4.6. Inicialización de arreglos.	1. Implementa programas utilizando arreglos. 2. Manipula los datos almacenados en arreglos mediante punteros. 3. Diseña funciones que aceptan arreglos como argumentos. 4. Realiza operaciones de concatenación, búsqueda y manipulación de caracteres en cadenas.
5. Punteros a memoria interna.	5.1. Introducción a los punteros. 5.2. Variables punteras. 5.3. Los operadores de punteros. 5.4. Expresiones de punteros. 5.5. Punteros y arreglos. 5.6. Indirección múltiple. 5.7. Inicialización de punteros. 5.8. Punteros a funciones. 5.9. Funciones de asignación dinámica.	1. Explica los conceptos básicos de punteros. 2. Utiliza punteros para asignar valor, acceder y manipular datos en la memoria. 3. Emplea operadores de punteros para evaluar expresiones. 4. Aplica punteros en arreglos y funciones dinámicas.
6. Funciones.	6.1. Forma general de una función. 6.2. Reglas de ámbito de las funciones. 6.3. Argumentos de funciones. 6.4. La sentencia return. 6.5. Funciones que devuelven valores no enteros. 6.6. Prototipos de funciones. 6.7. Devolución de punteros. 6.8. Funciones del tipo void. 6.9. Lo que devuelve main. 6.10. Recursividad. 6.11. Declaración de listas de parámetros de longitud variable. 6.12. Declaración de parámetros de funciones clásicas frente a las modernas. 6.13. Aspectos de implementación. 6.14. Bibliotecas y archivos.	1. Explica la estructura y función de las funciones en programación. 2. Identifica las reglas de ámbito de las funciones. 3. Explica conceptos avanzados como prototipos de funciones, devolución de punteros y recursividad.
7. Manejo de caracteres y cadenas.	7.1. Fundamentos de cadenas y caracteres. 7.2. Funciones de entrada y salida.	1. Explica los conceptos fundamentales relacionados con cadenas y caracteres en programación.

Unidades	Contenidos	Resultados de aprendizaje
	7.3. Procesamiento de cadenas.	2. Aplica funciones de entrada y salida para leer y mostrar cadenas, así como realizar procesamiento de cadenas, como la concatenación, búsqueda, reemplazo y manipulación de caracteres dentro de cadenas en un programa.
8. Tipo abstracto de datos y modularidad.	8.1. El papel de la abstracción. 8.2. Abstracción en los lenguajes de programación. 8.3. Módulos. 8.3.1. La estructura de un módulo. 8.3.2. Reglas de modulación. 8.4. Diseño de módulos. 8.4.1. Acoplamiento de módulos. 8.4.2. Cohesión de módulos. 8.5. Tipo de datos. 8.6. Tipo abstracto de datos TAD. 8.7. Creación de un TAD. 8.8. Relación de TAD y objetos.	1. Identifica el papel de la abstracción en la programación. 2. Aplica conceptos de abstracción en un lenguaje de programación. 3. Implementa módulos en un lenguaje de programación aplicando los conceptos de acoplamiento y cohesión.
9. Datos básicos y agregados.	9.1. Estructuras. 9.2. Arreglos de estructuras. 9.3. Paso de estructuras a funciones. 9.4. Punteros a estructuras. 9.5. Arreglos y estructuras dentro de estructuras. 9.6. Campo de bits, uniones y enumeraciones.	1. Explica el concepto y uso de estructuras en programación. 2. Implementa arreglos de estructuras y pasar estructuras a funciones como argumentos. 3. Aplica los conceptos de campos de bits para gestionar datos compactos, uniones para compartir espacio de memoria y enumeraciones para definir conjuntos de valores discretos en programación.
10.Estructuras de datos.	1.1. Estructuras lineales estáticas y dinámicas. 1.1.1. Listas enlazadas. 1.1.2. Pilas. 1.1.3. Colas. 1.2. Estructuras no lineales estáticas y dinámicas. 1.2.1. Árboles. 1.2.2. Grafos.	1. Diferencia entre estructuras lineales estáticas y dinámicas. 2. Resuelve problemas específicos en programación utilizando estructuras lineales, como listas enlazadas, pilas y colas. 3. Resuelve problemas de mayor complejidad utilizando las estructuras no lineales estáticas y dinámicas, incluyendo árboles y grafos.
2. Manejo de programas por consola.	2.1. Escritura y lectura de caracteres. 2.2. Escritura y lectura de cadenas. 2.3. E/S por consola con formato.	1. Identifica como funcionan las operaciones de entrada y salida de caracteres en lenguajes de programación, y aplicando estas habilidades en la manipulación de datos de carácter individual. 2. Utiliza la escritura y lectura de cadenas y E/S por consola con formato.



Unidades	Contenidos	Resultados de aprendizaje
3. Manejo de memoria externa	3.1. Entrada y Salida por archivos. 3.2. Archivos secuencias. 3.3. Archivos aleatorios. 3.4. Elementos básicos del sistema de archivos.	1. Aplica las operaciones de entrada y salida sobre archivos secuencial y aleatorio en un programa. 2. Explica los elementos básicos del sistema de archivos, como directorios, rutas y permisos.
4. Compilación condicionada	4.1. El pre-procesador de C. 4.2. Directivas de compilación condicional. 4.3. Los operadores del pre-procesador # y ##. 4.4. Nombres de macros predefinidos. 4.5. Comentarios.	1. Utiliza las directivas del pre-procesador de C de compilación condicional, para gestionar la inclusión o exclusión de partes del código en función de condiciones específicas. 2. Utiliza los operadores del pre-procesador y nombres de macros predefinidos.
5. Gestión de memoria dinámica	5.1. Duración del almacenamiento asignado. 5.2. Gestores de memoria. 5.3. Malloc. 5.4. Calloc. 5.5. Free. 5.6. Estados de memoria. 5.7. Arreglos de elementos flexibles.	1. Comprende la duración del almacenamiento asignado y los gestores de memoria. 2. Utiliza funciones como malloc, calloc y free en la gestión de memoria.

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 un proyecto que involucre todos los resultados de aprendizaje de la materia. 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 plan de clases, de acuerdo con el perfil de los estudiantes, los recursos disponibles y el contexto educativo.

VI. ESTRATEGIAS EVALUATIVAS

Evaluaciones continuas de teoría y práctica, obtenida con la media de los controles realizados durante el curso a través de test semanales y ejercicios de laboratorio computados en los exámenes parciales, evaluaciones formativas como cuestionarios, exámenes cortos y ejercicios, para que puedan medir su progreso y realizar ajustes, examen final compuesto por un test escrito y trabajo práctico final.

Con fines de calificación y promoción se aplicará la normativa sobre evaluación 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

- Seacord, RC (2020). Effective C: Una introducción a la programación profesional en C. No Starch Press.
- Deitel, HM, & Deitel, PJ (última edición). Como programar en C/C++/Java. Prent
- Aguilar, F., & Martínez, R. (última edición). Estructura de datos: Una perspectiva en C. McGraw-H
- Kernighan, BW y Ritchie, DM (1991). El lenguaje de programación C.
- Joyanes Aguilar, L. (última edición). Fundamentos de programación. McGraw-Hill
- Prata, S. (2014). Imprimación C plus. Addison
- Alcalde, E., & García, M. (última edición). Metodología de la programación. McGraw
- Schildt, H. (1994). C: Guía de autoenseñanza. McGraw
- Schildt, H. (1989). C: Manual de referencia. McGraw-Hill

