



SÍLABO DE CURSO

PONTIFICIA UNIVERSIDAD CATÓLICA DEL
PERÚ
FACULTAD DE CIENCIAS E INGENIERÍA
DEPARTAMENTO DE CIENCIAS DE LA COMPUTACIÓN

FUNDAMENTOS DE PROGRAMACIÓN

INF144

Semestre Académico:	2025-1
Profesor(es):	Dr. Carlos Mendoza
Créditos:	4
Horas semanales:	5
Prerrequisitos:	Ninguno

1 INFORMACIÓN GENERAL

Clave	: codigo
Créditos	: credits
Tipo	: Obligatorio
Horas de dictado	:
Clase	: 3,00 horas semanales
Laboratorio	: 2,00 horas semanales
Horario	: Lunes y Miércoles de 10:00 a 12:00
Profesor(es)	: profesor
Departamento	: Ciencias de la Computación
Requisitos	: prerequisites

2 SUMILLA

Descripción General del Curso

El curso "Fundamentos de Programación" introduce a los estudiantes en los conceptos básicos y esenciales de la programación de computadoras. Se exploran las herramientas fundamentales para el diseño de algoritmos, como diagramas de flujo y pseudocódigo, y se desarrollan habilidades para la resolución sistemática de problemas mediante el pensamiento algorítmico. El curso cubre operaciones básicas, estructuras de control, tipos de datos fundamentales, y mecanismos de entrada/salida, sentando las bases para cursos más avanzados de la carrera.

3 OBJETIVOS Y COMPETENCIAS

Objetivos del curso

Al finalizar el curso, el estudiante estará en capacidad de:

- Comprender la evolución histórica de la programación y su importancia en el contexto actual.
- Diseñar algoritmos utilizando diagramas de flujo y pseudocódigo como herramientas de representación.
- Implementar soluciones algorítmicas usando tipos de datos fundamentales y operaciones básicas.
- Aplicar estructuras de control de secuencia, selección e iteración para resolver problemas.
- Desarrollar programas que interactúen con el usuario mediante operaciones de entrada y salida.
- Analizar y depurar programas simples identificando errores lógicos y sintácticos.

Competencias

El curso contribuye al desarrollo de las siguientes competencias profesionales:

- **Pensamiento algorítmico:** Capacidad para formular soluciones computacionales paso a paso.
- **Abstracción:** Habilidad para identificar aspectos esenciales de un problema y modelarlos en términos computacionales.
- **Razonamiento lógico:** Destreza para construir secuencias lógicas de instrucciones que resuelvan problemas específicos.
- **Comunicación técnica:** Capacidad para expresar soluciones algorítmicas en forma clara y precisa utilizando representaciones estándar.

4 CONTENIDO DEL CURSO

UNIDAD 1: Introducción a la Programación

- **Tema 1.1:** Introducción e historia (1,00 hora)
- **Tema 1.2:** Diagramas de flujo y Pseudocódigo (3,00 horas)

Duración total: 4,00 horas teóricas

UNIDAD 2: Conceptos Fundamentales

- **Tema 2.1:** Operaciones básicas (aritméticas, booleanas) (2,00 horas)
- **Tema 2.2:** Entrada y Salida de datos (con formato) (2,00 horas)
- **Tema 2.3:** Variables y Tipos de Datos (3,00 horas)

Duración total: 7,00 horas teóricas

UNIDAD 3: Estructuras de Control

- **Tema 3.1:** Estructuras secuenciales (2,00 horas)
- **Tema 3.2:** Estructuras condicionales (if-else, switch) (3,00 horas)
- **Tema 3.3:** Estructuras iterativas (for, while, do-while) (4,00 horas)

Duración total: 9,00 horas teóricas

UNIDAD 4: Modularización

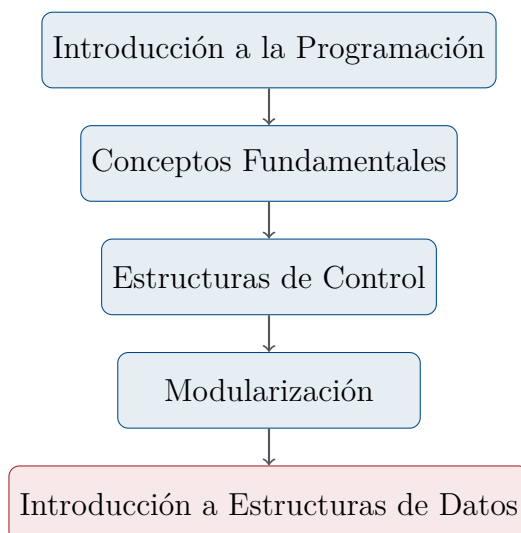
- **Tema 4.1:** Funciones y procedimientos (4,00 horas)
- **Tema 4.2:** Paso de parámetros (3,00 horas)
- **Tema 4.3:** Ámbito de variables (2,00 horas)

Duración total: 9,00 horas teóricas

UNIDAD 5: Introducción a Estructuras de Datos

- **Tema 5.1:** Arreglos unidimensionales (3,00 horas)
- **Tema 5.2:** Arreglos multidimensionales (3,00 horas)
- **Tema 5.3:** Introducción a estructuras/registros (2,00 horas)

Duración total: 8,00 horas teóricas

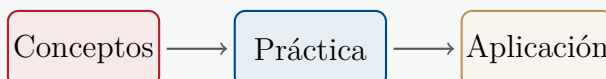


5 METODOLOGÍA

Enfoque Metodológico

El curso se desarrolla mediante una combinación de:

- **Clases teóricas:** Presentación de conceptos fundamentales de programación con ejemplos prácticos.
- **Sesiones de laboratorio:** Implementación de algoritmos y resolución de problemas en un entorno de programación.
- **Resolución de problemas:** Ejercicios progresivos que refuerzan los conceptos aprendidos.
- **Proyectos guiados:** Desarrollo de aplicaciones simples que integran los conceptos del curso.
- **Demostraciones:** Ejemplificación de técnicas de depuración y buenas prácticas de programación.



Se utilizará la plataforma virtual institucional para distribución de materiales, entregas de trabajos y comunicación fuera del horario de clases.

6 SISTEMA DE EVALUACIÓN

La nota final (NF) se calculará de la siguiente manera:

Sistema de Evaluación

Tipo de Evaluación	Descripción	Peso (%)
Prácticas de Laboratorio (PL)	Promedio de prácticas semanales	20%
Prácticas Calificadas (PC)	Promedio de 3 evaluaciones escritas	15%
Proyecto de curso (PR)	Desarrollo de aplicación integradora	15%
Examen Parcial (EP)	Evaluación escrita (Unidades 1-3)	25%
Examen Final (EF)	Evaluación escrita integral	25%
TOTAL		100%

Políticas de Evaluación

- Para aprobar el curso se requiere una nota final mínima de 11 sobre 20.
- Es obligatoria la asistencia a las sesiones de laboratorio.
- La presentación del proyecto de curso es requisito indispensable para aprobar.
- Los exámenes no rendidos se califican con nota cero (0).
- Casos de plagio serán sancionados según el reglamento de disciplina de la universidad.
- Las prácticas de laboratorio son evaluadas tanto por funcionalidad como por estilo y calidad de código.

7 CRONOGRAMA DEL CURSO

Cronograma del Curso

Semana	Contenido	Actividades
1	Unidad 1: Introducción e historia	Instalación del entorno de desarrollo
2	Unidad 1: Diagramas de flujo y Pseudocódigo	Laboratorio: Diseño de algoritmos simples
3	Unidad 2: Operaciones básicas	Práctica Calificada 1
4	Unidad 2: Entrada y Salida de datos	Laboratorio: Implementación de E/S
5	Unidad 2: Variables y Tipos de Datos	Laboratorio: Manejo de variables y tipos
6	Unidad 3: Estructuras secuenciales	Introducción al proyecto de curso
7	Unidad 3: Estructuras condicionales	Práctica Calificada 2
8	EXAMEN PARCIAL	Evaluación escrita (Unidades 1-3)
9	Unidad 3: Estructuras iterativas	Laboratorio: Ciclos y repeticiones
10	Unidad 4: Funciones y procedimientos	Laboratorio: Implementación de funciones
11	Unidad 4: Paso de parámetros	Avance de proyecto (Primera revisión)
12	Unidad 4: Ámbito de variables	Práctica Calificada 3
13	Unidad 5: Arreglos unidimensionales	Laboratorio: Manipulación de vectores
14	Unidad 5: Arreglos multidimensionales	Laboratorio: Matrices y aplicaciones
15	Unidad 5: Introducción a estructuras/registros	Presentación final de proyectos
16	EXAMEN FINAL	Evaluación escrita integral

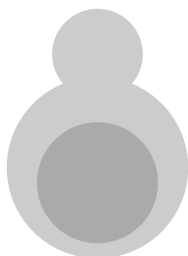
8 BIBLIOGRAFÍA

Bibliografía

- [1] Deitel, P. J., & Deitel, H. M. (2016). *C How to Program* (8th ed.). Pearson.
- [2] Stroustrup, B. (2013). *The C++ Programming Language* (4th ed.). Addison-Wesley.
- [3] Cormen, T. H., Leiserson, C. E., Rivest, R. L., & Stein, C. (2009). *Introduction to Algorithms* (3rd ed.). MIT Press.
- [4] Sebesta, R. W. (2015). *Concepts of Programming Languages* (11th ed.). Pearson.
- [5] Kernighan, B. W., & Ritchie, D. M. (1988). *The C Programming Language* (2nd ed.). Prentice Hall.
- [6] Joyanes Aguilar, L. (2008). *Fundamentos de Programación: Algoritmos, Estructuras de Datos y Objetos* (4ta ed.). McGraw-Hill.

9 INFORMACIÓN DEL DOCENTE

Perfil del Docente



Nombre	Dr. Carlos Mendoza
Correo Electrónico	cmendoza@pucp.edu.pe
Horario de Asesoría	Martes y Jueves 14:00 - 15:00
Formación	Doctor en Ciencias de la Computación, Universidad de Illinois
Especialización	Lenguajes de programación, algoritmos y educación en computación
Investigación	Métodos efectivos para la enseñanza de programación, lenguajes educativos

10 POLÍTICA INSTITUCIONAL CONTRA EL PLAGIO

Integridad Académica

Para la corrección y evaluación de todos los trabajos del curso se va a tomar en cuenta el debido respeto a los derechos de autor, castigando severamente cualquier indicio de plagio con la nota CERO (00).

Estas medidas serán independientes del proceso administrativo de sanción que la facultad estime conveniente de acuerdo a cada caso en particular. Se recomienda revisar el Reglamento de Disciplina de la PUCP, disponible en la página web institucional.

La originalidad de los trabajos es un valor esencial de la comunidad académica. Se espera que todos los estudiantes citen adecuadamente las fuentes utilizadas y presenten trabajos que reflejen su propio análisis y reflexión.

