

# PONTIFICIA UNIVERSIDAD CATÓLICA DEL PERÚ

## ESTUDIOS GENERALES CIENCIAS

### PROGRAMA ANALÍTICO

CURSO	: FUNDAMENTOS DE PROGRAMACIÓN (FPRO)
CLAVE	: 1INF01
TIPO	: OBLIGATORIO PARA TODAS LAS ESPECIALIDADES
CRÉDITOS	: 3.00
HORAS DE:	
TEORÍA	: 2 SEMANALES
LABORATORIO	: 2 SEMANALES
REQUISITOS	: (1FIS02)
SEMESTRE	: 2018-2

---

#### I. Objetivos del curso

Al terminar el semestre el alumno comprenderá y aplicará la lógica de proposiciones empleando la negación, la conjunción y la disyunción en la representación del conocimiento. Reconocerá lo que se entiende por algoritmo identificando las principales características del mismo. Comprenderá y aplicará los pasos clásicos para la resolución de problemas usando el computador así como el funcionamiento de las estructuras selectivas e iterativas. Será capaz de explicar cómo funcionan los algoritmos expresados en diagrama de flujo y podrá diseñar algoritmos expresados en pseudocódigos. Implementará algoritmos que contengan estructuras selectivas e iterativas usando un lenguaje de programación imperativo y entenderá la importancia del código limpio para la selección de nombres de identificadores y comentarios en la implementación de los algoritmos.

#### II. Metodología

El curso es de carácter expositivo con resolución de trabajos prácticos presentados en la computadora. Para el desarrollo del curso se utilizarán recursos audiovisuales y guías de estudio como complemento a lo expuesto durante las clases. En determinadas clases, se hará uso de un equipo de cómputo que permita mostrar diversas situaciones prácticas.

En las sesiones de prácticas, los alumnos resuelven de forma individual o grupal, según se indique, un tema preparado por la coordinación del curso, el cual se basa en los materiales de enseñanza disponibles para el curso. En estas sesiones, se relacionan los conocimientos teóricos con la práctica.

En las sesiones de práctica, se busca que el estudiante utilice un entorno de desarrollo integrado que permita la implementación de algoritmos usando el paradigma de programación imperativo. Las consultas que el estudiante necesite realizar al profesor del curso las puede hacer durante la clase, en los horarios de asesoría que el profesor proporcione, por correo electrónico o algún otro medio electrónico que indique el profesor del curso.

#### III. Sumilla

El curso es de naturaleza teórico-práctica y tiene como propósito desarrollar las capacidades para resolver diversos problemas de las ciencias e ingeniería planteando alternativas de solución que utilicen como herramienta la programación imperativa. Se desarrollan las nociones básicas de lógica proposicional con énfasis en la negación, la conjunción y la disyunción. Asimismo, se estudian los conceptos necesarios para el diseño de algoritmos enfatizando la representación de estructuras algorítmicas selectivas e iterativas por medio del diagrama de flujo y el pseudocódigo. Finalmente, se implementarán los algoritmos usando un lenguaje de programación imperativo.

#### IV. Descripción del programa

##### **CAPÍTULO 1.** Lógica proposicional (4 horas)

Proposiciones simples, proposiciones compuestas, negación, disyunción, conjunción, tablas de verdad, tautología, contradicción, contingencia. Equivalencias lógicas: leyes de identidad, leyes de dominación, leyes de idempotencia. Ley de la doble negación, leyes conmutativas, leyes asociativas, leyes distributivas, leyes de Morgan. Reglas de inferencia: reglas de adición, reglas de simplificación, reglas de silogismo disyuntivo, reglas de silogismo hipotético, reglas de conjunción, reglas de ponens.

##### **CAPÍTULO 2.** Algoritmia (1 hora)

Definición de algoritmia, definición de algoritmo, tipos de algoritmos, formas de representación de los algoritmos.

**CAPÍTULO 3. Resolución de problemas usando el computador (1 hora)**

Fases del desarrollo de problemas usando el computador: definición del problema, análisis, diseño, implementación, ejecución y pruebas.

**CAPÍTULO 4. Diseño de algoritmos (8 horas)**

Pseudocódigos, diagramas de flujo. Estructura selectiva simple, estructura selectiva doble. Ciclo iterativo con entrada controlada, ciclo iterativo con salida controlada. Resolución de problemas con estructuras selectivas e iterativas. Tabla de iteración.

**CAPÍTULO 5. Implementación de algoritmos (14 horas)**

Tipos de estructuras selectivas: simples, dobles y múltiples. Diseño de algoritmos para resolver problemas utilizando estructuras selectivas. Desarrollo de programas utilizando estructuras selectivas.

**CAPÍTULO 6. Estructuras algorítmicas iterativas (6 horas)**

Tipos de datos y sus operaciones. Constantes, variables y expresiones, Ámbito de variables. Diseño de algoritmos y desarrollo de programas para resolver problemas basados en tratamiento de datos. Estructura selectiva simple, estructura selectiva doble. Ciclo iterativo con entrada controlada, ciclo iterativo con salida controlada. Resolución de problemas con estructuras selectivas e iterativas. Implementación de funciones.

**V. Descripción de las sesiones de laboratorio**

**SESIÓN 1. Programación imperativa**

Estructura básica de un programa.

**SESIÓN 2. Programación imperativa**

Elaboración de programas secuenciales.

**SESIÓN 3. Estructuras algorítmicas selectivas**

Elaboración de programas con estructuras selectivas simples.

**SESIÓN 4. Estructuras algorítmicas selectivas**

Elaboración de programas con estructuras selectivas dobles.

**SESIÓN 5. Estructuras algorítmicas selectivas**

Elaboración de programas con estructuras selectivas simples y dobles.

**SESIÓN 6. Estructuras algorítmicas iterativa**

Elaboración de programas con ciclos iterativos.

**SESIÓN 7. Estructuras algorítmicas iterativa**

Elaboración de programas con ciclos iterativos.

**SESIÓN 8. Estructuras algorítmicas iterativa**

Elaboración de programas con ciclos iterativos y selectivos.

**SESIÓN 9. Reutilización**

Implementación de funciones.

**SESIÓN 10. Depuración de programas**

Ejecución de programas paso a paso. Inspección de variables.

**VI. Bibliografía**

• Textos guía

DEITEL, Harvey M.

2004 *C: how to program*. Cuarta edición. Upper Saddle River, N.J.: Prentice-Hall.

KERNIGHAN, Brian W.

1991 *El lenguaje de programación C*. Segunda edición. México D.F.: Prentice-Hall.

LÓPEZ ROMAN, Leobardo

2005 *Programación estructurada en lenguaje C*. México, D.F.: Alfaomega.

• Texto complementario

CORREA URIBE, Guillermo

1992 *Desarrollo de algoritmos y sus aplicaciones en Basic, Pascal, Cobol y C con su respectivo pseudocódigo*. Tercera edición. Bogotá: McGraw-Hill.

## VII. Sistema de evaluación

Los promedios de prácticas se calculan con aproximación hasta las décimas. Cualquiera sea la cifra de las centésimas, no se tomará en cuenta.

La nota final del curso se expresa solo en números enteros. Si el cálculo de la nota final da un total con decimales, debe convertirse esa cifra a enteros (se añade un punto a la nota si el primer decimal es cinco o más; se elimina el decimal si es menor de 5).

La nota final del curso se calculará utilizando la fórmula que a continuación se detalla. En ella se usa la siguiente nomenclatura:

$N_f$  : nota final

$E_1$  : nota del primer examen (medio ciclo)

$E_2$  : nota del segundo examen (final)

$Pr_b$  : promedio de prácticas de tipo Pb, eliminando la nota más baja.

$$N_f = \frac{2E_1 + 3E_2 + 5Pr_b}{10}$$

Para los alumnos que rindan el examen especial, este reemplazará al examen al cual el alumno faltó según los artículos 5° y 41° del Sistema de Evaluación.

San Miguel, agosto de 2018