



UNIVERSIDAD DISTRITAL FRANCISCO JOSÉ DE CALDAS

**SYLLABUS DE
ALGORITMOS Y PROGRAMACIÓN EN C/C++**

FACULTAD: CIENCIAS Y EDUCACIÓN

PROYECTO CURRICULAR: LICENCIATURA EN
FÍSICA

COMPONENTE: Fundamentos generales

*Cursos teóricos prácticos - Componente práctico y/o
experimental (Componente práctico).*

Obligatorio

CÓDIGO: 24314

Pre-requisitos: 24308 – TIC en Enseñanza de la Física II
4706 – Álgebra Lineal

Co-requisitos: Ninguno

NÚMERO DE CREDITOS: 3

I. JUSTIFICACIÓN DEL ESPACIO ACADÉMICO

Las nuevas tecnologías y las computadoras han ampliado la forma de resolver problemas en el mundo actual. La enseñanza de las ciencias se ha beneficiado por este hecho; existe ahora una gran cantidad de lenguajes de programación y de Software con los que es posible elaborar y resolver problemas en diferentes áreas de la física.

Conscientes de la necesidad que representa la computación para el Licenciado en Física y los cambios metodológicos que ello implica, este curso proporciona un primer acercamiento a una de las herramientas más fundamentales del uso de la informática como lo son los lenguajes de programación. Uno de los lenguajes que se ha convertido en paradigma de la programación es el C; sin embargo, la constitución de rutinas que resuelven problemáticas concretas exige el conocimiento del problema en cuestión, y el planteamiento de algoritmos que den la solución requerida. De allí, la necesidad de construir los elementos que permitan al estudiante diseñar los algoritmos, los que incluyen aspectos lógicos, diagramáticos, entre otros. Buscando constituirse como el espacio, por excelencia, donde dichos aspectos sean conocidos y desarrollados.

**II. PROGRAMACION DEL CONTENIDO
OBJETIVO GENERAL**

Fundamentar al estudiante en el análisis de problemas, el diseño y construcción de algoritmos secuenciales que resuelven dicho problema, y correspondiente codificación en C/C++.

OBJETIVOS ESPECÍFICOS

- Formar al estudiante en el reconocimiento y análisis de problemas solubles a través de algoritmos.
- Estructurar y fundamentar a los estudiantes en la estructura de algoritmos secuenciales.

- Utilizar los algoritmos como una estructura de tipo lógico, buscando que se conviertan en un mecanismo de solución de problemas y en especial de la física.
- Utilizar el pseudocódigo como elemento facilitador en el diseño y codificación de los algoritmos en C/C++.
- Aplicar y desarrollar las sentencias, y estructuras de control fundamentales del C/C++ para la codificación de los algoritmos.
- Fundamentar al estudiante en la programación modular y el uso de la recursividad.
- Impulsar y promover el aprendizaje autónomo, con capacidad lógica y creativa con el fin de fomentar una actitud positiva hacia la ciencia y en particular, hacia la Física.
- Uso del compilador gcc/g++ de linux, y escritura de programas en editores de texto plano.

UNIDADES TEMÁTICAS Y/O PROBLEMÁTICAS

- **UNIDADES Y MEDIDAS:**

Por ser la física una de las ciencias naturales, requiere de unas bases teóricas y experimentales suficientes para poder entender distintas fenomenologías que están en nuestro entorno. Para abordar coherentemente ese estudio se requiere de: a) Estudio de magnitudes y cantidades físicas, unidades y sistemas de unidades, proceso de medición, y análisis dimensional, b) Tratamiento básico de datos experimentales que incluye la medición, errores, precisión instrumental, incertidumbre experimental y el tratamiento estadístico de datos, c) Representación gráfica de datos experimentales y su análisis a través del análisis de gráficas lineales, no lineales, en escalas logarítmicas y semilogarítmicas, y d) La obtención de la relación funcional de los datos desde el proceso de linealización de las gráficas y desde el análisis de datos a partir del ajuste de curvas usando métodos de regresión.

- **LÓGICA:**

Estructura del argumento, identificación del argumento, argumentos complejos, lógica formal y lógica informal, evaluación del argumento, verdad de las premisas, proposiciones lógicas, formas argumentales, formalización y semántica de los operadores lógicos.

- **ALGORITMOS:**

¿Qué es un algoritmo?, paradigma de la programación estructurada, el lenguaje C, etapas del desarrollo e implementación de un programa: (Análisis del problema, diseño del algoritmo, codificación, compilación y ejecución, verificación y depuración, documentación), representación gráfica de algoritmos, estructura secuencial, variables y constantes, estructura de decisión, estructura de repetición, inicialización, contadores, acumuladores, estructura modular, pseudocódigo.

- **FUNDAMENTOS DE PROGRAMACIÓN ESTRUCTURADA EN C:**

caracteres de C, identificadores y palabras claves, tipos de datos, variables y arreglos, casting, declaraciones, operadores y expresiones, operadores relacionales y lógicos, operadores unarios, jerarquía de los operadores, compendio básico de funciones

matemáticas, entrada y salida de datos, estructuras de control: if, if-else, while, do-while, for, break, continue, switch.

- Algoritmos matemáticos:

Algoritmo de Euclides, polinomios, números aleatorios (método congruencial lineal, método congruencial aditivo, prueba de aleatoriedad), ajuste de curvas: (interpolación polinomial, método de los mínimos cuadrados), integración (métodos de cuadratura simple, método del trapezoide, método de Simpson, método de Romberg (**NOTA: LEER LA METODOLOGÍA**)).

- ALGORITMOS DE ORDENAMIENTO:

Ordenamiento elemental, ordenamiento Shell, ordenamiento de burbuja, ordenamiento rápido básico, ordenamiento Radix (**NOTA: LEER LA METODOLOGÍA**)

- Funciones y otras estructuras en C:

Definición y acceso a funciones, recursividad, paso de argumentos por valor y por referencia, arreglos unidimensionales y cadenas, arreglos de más dimensiones (**NOTA: LEER LA METODOLOGÍA**).

- ARCHIVOS O FICHEROS:

Tipos de archivos, apertura y cierre de archivos, punteros y archivos, operaciones de lectura y escritura, otras operaciones.

- Algoritmos geométricos:

Métodos geométricos elementales: puntos, líneas y polígonos, camino simple cerrado, inclusión en un polígono. Algoritmo de la envolvente convexa: algoritmo de envase, algoritmo de Graham (**NOTA: LEER LA METODOLOGÍA**).

- Algoritmos de problemas cotidianos:

Estaciones de gas, redares de inspección, estacionamientos (**NOTA: LEER LA METODOLOGÍA**).

III. ESTRATEGIAS

METODOLOGÍA PEDAGÓGICA Y DIDÁCTICA

La metodología debe favorecer el desarrollo de las habilidades relacionadas con el planteamiento y solución de problemas. Además, debe motivar una actitud proactiva, en la que el estudiante asuma su rol desde una perspectiva participativa, crítica, responsable y comprometida con su formación profesional. Y puede incluir clases magistrales, seminarios, Seminario – Talleres, Talleres, Proyectos tutoriados, laboratorios, entre otros.

Es importante contar con el aspecto metodológico relacionado en el contenido del curso referente al desarrollo de software, es decir, se debe trabajar permanente con el estudiante en el reconocimiento y análisis del problema, el diseño y construcción del algoritmo, y la posterior prueba de este último —la prueba de escritorio— ya sea a través de diagramas de flujo o pseudocódigo. Y en un último estadio, una vez validado el algoritmo, con la codificación y compilación.

Finalmente, con respecto del listado de algoritmos matemáticos, de ordenamiento, geométricos y cotidianos, y en honor de la vastedad temática de los mismos, se recomienda tomar por lo menos un caso de cada uno de ellos; en otras palabras, que todos los algoritmos listados en el contenido deben tomarse como una guía.

La distribución del trabajo de los estudiantes se muestra a continuación:

Tipo de Curso	Horas			Horas profesor/semana	Horas Estudiante/semana	Total Horas Estudiante/semestre	Créditos
	TD	TC	TA	(TD + TC)	(TD + TC +TA)	X 16 semanas	
Teórico	2	4	3	6	9	144	3
Práctico							

Trabajo Presencial Directo (TD). Trabajo Mediado _ cooperativo (TC) Trabajo Autónomo (TA).

IV. RECURSOS

Estos deben ser garantizados por la universidad en su totalidad, es decir la institución debe responsabilizarse por el personal docente, auxiliar de laboratorios, espacios físicos y las demás condiciones pertinentes para el desarrollo del curso.

BIBLIOGRAFÍA

TEXTOS BÁSICOS

- JOYANES A., LUIS.; ZAHONERO M., IGNACIO. (2004). Algoritmos y estructuras de datos una perspectiva en C. McGraw Hill.
- FARRÉ, R.; SIERRA, J.; OLIVERAS, A.; NIEUWENHUIS, R.; RODRÍGUEZ, E.; NIVELA, P. (2012). Lógica para informáticos. Alfaomega, Marcombo.
- KNUTH, D. (1980). El arte de programar ordenadores; T.1 Algoritmos fundamentales. Reverté.
- NOLT, J.; ROHATYN, D.; VARZI, A.(2011). Schaum's Outline of Logic; second edition. McGraw Hill.
- VEGA, A.; JEDER, I.; LOPEZ, G. (2009). Análisis y diseño de algoritmos. Buenos Aires: Alfaomega.
- CORMEN; THOMAS, H. (2001). Introduction to Algorithms; 2nd edition. MIT Press / McGraw-Hill.
- LIPSCHUTZ, S. (1988). Matemáticas Para Computación, serie Schaum. McGraw Hill.
- Mancilla Herrera, A.; Ebratt Gómez, R.; Capacho Portilla, José. (2015). Diseño y construcción de algoritmos. Ediciones de la U.
- DEITEL; DEITEL. (1995). Cómo programar en C. Ed. Prentice Hall.
- RODRIGUEZ A.; MIGUEL, A. (1991). Metodología de programación a través de pseudocódigo. McGraw Hill, 1ª. Edición.
- SEDGEWICK R. (1995). Algoritmos en C++. Addison Wesley.

- CEBALLOS SIERRA, FCO. JAVIER. (2015). C/C++. Curso de programación. 4ª edición. RA-MA editorial.

EVALUACIÓN

- Se plantea una metodología de seguimiento al desarrollo de los estudiantes de acuerdo con los objetivos propuestos.
- Se plantean las técnicas utilizadas para hacer mediciones tanto de corte cuantitativo como cualitativo que permiten definir el nivel de desarrollo del estudiante.
- Se define la ponderación o distribución en porcentaje de valoración de las diferentes estrategias de medición utilizadas, en concordancia con la normatividad vigente.