



## 1. Información General

<b>Espacio Académico</b>	Programación			
<b>Código</b>	1134			
<b>Tipo</b>	Espacio teórico-práctico			
<b>Área</b>	Básicas de Ingeniería			
<b>Créditos académicos</b>	<b>HTD</b>	<b>HTC</b>	<b>HTA</b>	<b>Horas/semana</b>
	0	4	2	6
	<b>2 créditos</b>			

## 2. Justificación

En la actualidad los múltiples requerimientos de diseño, control y ejecución de procesos, que desarrollan los tecnólogos en sus labores diarias, exigen soluciones que requieren de lenguajes de programación capaces de brindar agilidad, seguridad y soluciones estándares, a empresas Colombianas que abren sus puertas, a un mundo global dinámico y sin fronteras. Por esta razón, el profesional debe estar en capacidad de resolver problemas propios de su disciplina (eléctrica, sistemas, electrónica, etc.) haciendo uso de programas en lenguaje C, que estén bien estructurados, que sean funcionales y de buena calidad.

## 3. Objetivos

- Sembrar en el tecnólogo los conocimientos y técnicas de la programación estructurada, mediante el uso de programas en lenguaje C.
- Dominar la programación estructurada modular, mediante la discusión detallada de una amplia variedad de algoritmos; asumiendo el dominio de los rudimentos de la computación.
- Lograr que el tecnólogo tenga la capacidad de resolver una amplia gama de problemas mediante las técnicas de programación, en las disciplinas que sean de su interés.
- Sentar las bases para que el tecnólogo adquiera en el futuro, el dominio de otros métodos de programación, tales como la programación orientada a objetos.

## 4. Requerimientos

No tiene ningún tipo de prerrequisito cognitivo.

## 5. Aspectos pedagógicos

La metodología se ha desarrollado desde el punto de vista general, lo que cabe destacar son las prácticas de laboratorio; las clases de teoría se apoyan de material escrito que le puede servir de guía sobre los contenidos más importantes de la asignatura, las explicaciones teóricas se intercalarán con la realización de problemas, ejemplos prácticos y aplicaciones siempre que el contenido lo requiera.

Se desarrollan actividades en grupos, solución de problemas y cuestiones teórico-prácticas relacionadas con la asignatura, de manera que se intente reforzar y aplicar los conceptos básicos a situaciones reales concretas y fomentar la capacidad de análisis y síntesis de los estudiantes. Adicionalmente se proponen prácticas de laboratorio que son implementadas en el curso. Las primeras sesiones se dedicarán a una introducción de programación, de tal forma que le permitir al estudiante hacer un reconocimiento claro del lenguaje de programación.

- Los trabajos complementarios: dichos trabajos incidirán en la nota de la asignatura y pueden ser teóricos, prácticos o teórico-prácticos y deberán realizarse de forma individual o por grupo de máximo dos estudiantes.
- Tutorías de atención al alumnado: El alumnado tiene a su disposición unas horas de tutorías en las cuales puede consultar cualquier duda relacionada con la asignatura, así como dudas concretas sobre el contenido de la asignatura.
- Horarios de disponibilidad de prácticas: Contiene información sobre las horas en las cuales los estudiantes pueden acudir a sus prácticas libres.
- Evaluación: Con el fin de que el alumno verifique si posee el nivel de conocimientos mínimo que le permita afrontar la elaboración de las prácticas de la asignatura con ciertas garantías de éxito.

## 6. Descripción de créditos

Distribución de las actividades		Horas semanales	Horas semestre	Número de créditos
Clase presencial (trabajo directo)	Introducción de concepto Ejemplificación del contenido Realización de ejercicios y problemas Talleres de refuerzo Evaluación y Diagnóstico de conocimientos	2	32	2
Acompañamiento (trabajo cooperativo)	Seguimiento a los talleres y consultas Talleres extra clase	1	16	
Actividades extractase (trabajo autónomo)	Lecturas previas Talleres y consultas extra clase Ejercicios y trabajos	1	16	
<b>TOTAL</b>		4	64	

## 7. Competencias e indicadores

Nombre de la unidad temática	Competencias	Indicadores de idoneidad
Capítulo 1: Conceptos básicos de algoritmos	Interpretativa, Argumentativa y Propositiva	<p>Plantea del problema a resolver y conocer perfectamente el problema y los resultados a obtener.</p> <p>Construye instrucciones concretas, sin ninguna ambigüedad.</p> <p>Todos sus pasos deben ser simples y tener un orden definido.</p> <p>Maneja el diseño de algoritmos, describiendo la secuencia ordenada de pasos que conducen a la solución de un problema dado. (Análisis del problema y desarrollo del algoritmo).</p> <p>Expresa el algoritmo como un programa de lenguaje de programación adecuado. (Fase de codificación.)</p> <p>Desarrolla y propone diversos análisis y soluciones a ejercicios básicos.</p>
Capítulo 2: Conceptos básicos de Diagramas de Flujo	Interpretativa y Argumentativa y Propositiva	<p>Comprende la importancia del manejo de Diagramas de Flujo, como herramienta de ayuda alterna, a la solución de problemas.</p> <p>Comprende e Identifica los símbolos utilizados en los diagramas.</p> <p>Organiza los símbolos de tal forma que siga visualmente el flujo de arriba hacia abajo y de izquierda a derecha.</p> <p>Construye flujos de datos concretos, sin ninguna ambigüedad.</p> <p>Maneja el diseño del flujo grama, describiendo la secuencia ordenada de pasos que conducen a la solución de un problema dado.</p> <p>Desarrolla y propone diversos análisis y soluciones a ejercicios básicos.</p>
Capítulo 3: Tipos de datos en el Lenguaje C.	Interpretativa, Argumentativa	<p>Identifica los diferentes caracteres o códigos especiales en el lenguaje C.</p> <p>Sabe como definir una variable en memoria y como almacenar en ella un valor que puede o no cambiar.</p> <p>Comprende el alcance y la duración de una variable en la ejecución de un programa.</p>
Capítulo 4: Funciones estándar y Operadores	Interpretativa, argumentativa y propositiva	<p>Define los principales elementos de un código de programación</p> <p>Identifica los diferentes tipos de operadores y realiza expresiones con ellas.</p> <p>Comprende el manejo del programa, y realiza ejercicios básicos con resultados por pantalla.</p> <p>Desarrolla y propone diferentes soluciones a ejercicios planteados.</p> <p>Identifica y manipula las funciones printf, scanf, getch, gotoxy, textcolor, entre otros.</p>
Capítulo 5: Controladores de Flujo (Bifurcaciones y Bucles)	Interpretativa, argumentativa y propositiva	<p>Identifica los controladores de flujo if, else if.</p> <p>Identifica los bucles for, while, do while.</p> <p>Desarrolla y propone diversos análisis y soluciones a ejercicios básicos.</p> <p>Construye diversas soluciones aplicando la lógica al programar.</p>
Capítulo 6: Arreglos	Interpretativa, argumentativa y propositiva	<p>Comprende el concepto de matriz</p> <p>Declara una matriz unidimensional y multidimensional.</p> <p>Desarrolla y propone diferentes soluciones a ejercicios planteados.</p>
Capítulo 7: Creación y manejo de funciones.	Interpretativa, argumentativa y propositiva	<p>Tener conocimiento sobre los lenguajes de programación, para poder desarrollar una aplicación.</p> <p>Tener capacidad de implementar un programa dado un problema de la vida real.</p> <p>Tener el concepto de funciones.</p> <p>Dar solución a un ejercicio propuesto por medio de funciones.</p>

## 8. Contenido programático

	Semana/Sesión	Lineamientos	HSP	HSC	THS
Capítulo 1: Conceptos básicos de algoritmos	<u>1/1</u>	Desarrollar capacidades en los estudiantes para atacar problemas de una manera lógica y simplificada Presentación del curso Principios básicos de Algoritmos	1	1	2
	<u>1/2</u>	Pasos y tipos de algoritmos Análisis del problema	1	1	2
	<u>2/3</u>	Diseño del algoritmo. Pseudo código Ejercicios	1	1	2
	<u>2/4</u>	Pseudo código Ejercicios	1	1	2
Capítulo 2: Conceptos básicos de Diagramas de Flujo	<u>3/5</u>	Utilizar los diagramas de flujo como herramienta necesaria para formular, analizar y solucionar problemas para un determinado procedimiento Introducción Características de los Flujogramas	1	1	2
	<u>3/6</u>	Tipos de Flujogramas Simbología de los Flujogramas	1	1	2
	<u>4/7</u>	Diseño y Elaboración de Flujogramas	1	1	2
	<u>4/8</u>	Ejercicios	1	1	2
Capítulo 3: Tipos de datos en el Lenguaje C.	<u>5/9</u>	Saber como definir una variable en memoria y como almacenar en ella un valor que puede o no cambiar. Estudiar el alcance y la duración de una variable en la ejecución de un programa. Tipos de datos en C y su declaración. Constantes en el lenguaje C y asignación de valores.	1	1	2
	<u>5/10</u>	Variables en el lenguaje C y asignación de valores. Caracteres o códigos especiales en el lenguaje C.	1	1	2
	<u>6/11</u>	Ejercicios	1	1	2
Capítulo 4: Funciones estándar y Operadores	<u>6/12</u>	Estudiar como introducir valores mediante el teclado, a las variables definidas en la memoria (funciones de entrada en tiempo de ejecución). Estudiar los principales operadores del lenguaje C. aritméticos, relacionales, lógicos y de asignación. Funciones de entrada de datos: getchar, getch, getche, gets y scanf con sus caracteres de control. Funciones de salida de datos: putchar, puts, printf y fprintf con sus respectivos caracteres de control. Funciones incorporadas en el lenguaje C	1	1	2
	<u>7/13</u>	Funciones para el manejo de caracteres y cadenas. Funciones matemáticas. Funciones aleatorias	1	1	2
	<u>7/14</u>	Tabla de prioridades de los operadores. Expresiones con operadores aritméticos. Expresiones con operadores relacionales.	1	1	2
	<u>8/15</u>	Expresiones con operadores lógicos. Expresiones con operadores de asignación Ejercicios	1	1	2

**UNIVERSIDAD DISTRITAL “FRANCISCO JOSÉ DE CALDAS”**

**Facultad Tecnológica**

	Semana/Sesión	Lineamientos	HSP	HSC	THS
Capítulo 5: Controladores de Flujo (Bifurcaciones y Bucles)	8/16	Estudiar y aplicar las instrucciones que permiten decidir, entre dos o entre múltiples alternativas de ejecución de instrucciones, según cierta condición sobre variables.	2	2	4
	9/17	Estudiar y aplicar las estructuras que ofrece el lenguaje para realizar la interacción de un bloque de instrucciones. Generalidad Estructuras condicionales: if ( ), if ( )-else ( ) Ejercicios			
	9/18	Estructura de bifurcación múltiple: switch-case ( ). Break, Ejercicios			
	10/19				
	10/20	Estructuras de iteración: for ( ), while ( ), do-while ( ). Ejercicios			
	11/21				
Capítulo 6: Arreglos	11/22	Estudiar el tratamiento que da el lenguaje a una gran cantidad de datos o variables que son del mismo tipo y que matemáticamente representan un vector.	2	2	4
	12/23	Comprender como maneja el lenguaje aquellos problemas en los cuales se generan datos o variables del mismo tipo, y que forman una tabla o matriz. Concepto de arreglo y tipos de arreglos. Ejercicios			
	12/24	Declaración y definición de arreglos unidimensionales. Ejercicios			
	13/25				
	13/26	Declaración y definición de arreglos multidimensionales. Ejercicios			
	14/27				
Capítulo 7: Creación y manejo de funciones.	14/28	Diseñar y aplicar en lenguaje C, las funciones definidas por el programador.	2	2	4
	15/29	Aprender el paso entre una función y otra, de los valores o direcciones de las variables o parámetros, según sea el caso. Prototipo y definición de función. Ejercicios			
	15/30	Acceso de una función. Ejercicios			
	16/31				
	16/32	Argumentos de una función: variables y arreglos Ejercicios	1	1	2
			<b>32</b>	<b>32</b>	<b>64</b>

## 9. Estrategias de evaluación

Parciales	X	Talleres, tareas y otros	X
Prácticas de laboratorio	X	Proyectos	X

## 10. Valoración de las estrategias de evaluación

	Estrategia	Porcentaje	Temas a evaluar	Fecha
<b>1<sup>ra</sup> Nota</b>	Evaluación escrita.	5%	Capítulo 1.	-
<b>2<sup>ra</sup> Nota</b>	Trabajos en clase, consultas, participación.	10%	Capítulo 1, 2, 3.	-
<b>3<sup>ra</sup> Nota</b>	Evaluación practica individual.	20%	Capítulo 1, 2, 3.	-
<b>4<sup>ta</sup> Nota</b>	Evaluación escrita.	5%	Capítulo 4.	-
<b>5<sup>ta</sup> Nota</b>	Trabajos en clase, consultas, participación.	10%	Capítulo 4, 5, 6, 7.	-

**UNIVERSIDAD DISTRITAL “FRANCISCO JOSÉ DE CALDAS”**  
**Facultad Tecnológica**

<b>6<sup>ta</sup> Nota</b>	Evaluación practica individual.	20%	Capítulo 4, 5, 6, 7.	-
<b>7<sup>ta</sup> Nota</b>	Examen Final	30%	todos	-

## **11. Bibliografía y demás fuentes de documentación**

- Becerra S, Cesar A., Lenguaje C, el nuevo concepto, Editorial Kimpresxltda, 1987.
- Ceballos, Francisco J., Enciclopedia del Lenguaje C, Editorail Alfaomega, 1997.
- Brian M., Ritchie., El lenguaje de programación C, Prentice Hall, 1991.
- Schildt, H., C++: Guía de Autoenseñanza, McGraw-Hill, 2<sup>a</sup> edición, 1995.
- Stroustrup, B., The C++ Programming Language, Addison-Wesley, 3<sup>a</sup> edición, 1997.
- Ellis, M.A. y Stroustrup, B., Manual de Referencia C++ con Anotaciones, Addison-Wesley/Díaz de Santos, Madrid, 1994.
- Becerra Santamaría Cesar A. Una Herramienta para la Programación Orientada a Objetos. 5<sup>a</sup> Edición. 2006.
- Joyanes Aguilar, Luis., Borland C++ 4/4.5 Iniciación y Referencia, Mc Graw Hill, 1996.
- Joyanes Aguilar, L.: Programación en C++. McGraw-Hill, 2000.
- Jamsa, Kris., Aprenda C++ paso a paso, Editorial Alfaomega, 1997.
- Deitel, H.M., Como Programar C++, Editorial Prentice Hall, 1999.