

# UNIVERSIDAD MILITAR NUEVA GRANADA



<b>CONTENIDO PROGRAMÁTICO</b>	Fecha Emisión: 2018/02/09	AC-GA-F-8
	Revisión No. 3	Página 1 de 9

<b>NOMBRE DEL PROGRAMA: INGENIERÍA BIOMÉDICA</b>	
<b>NOMBRE DE LA ASIGNATURA</b>	<i>Programación de computación</i>
<b>CÓDIGO</b>	<i>172701</i>
<b>SEMESTRE</b>	<i>Segundo (2)</i>
<b>PRERREQUISITOS</b>	<i>Programación I</i>
<b>CORREQUISITOS</b>	<i>Ninguno</i>
<b>COORDINADOR Y/O JEFE DE ÁREA</b>	<i>Ing. Liliana Quintero</i>
<b>DOCENTE (S)</b>	<i>Ing. Juan Camilo Gutiérrez Pinzón</i>
<b>CRÉDITOS ACADÉMICOS</b>	<i>Tres (3)</i>
<b>FECHA DE ELABORACIÓN/ ACTUALIZACIÓN</b>	<i>Noviembre 26 de 2020</i>

## JUSTIFICACIÓN

La programación orientada a objetos se hace pertinente para ampliar la visión de los estudiantes como programadores con conceptos de mayor dimensión y abstracción, los cuales le permitan resolver problemas de un grado de complejidad más alto y cercano a la realidad.

Con esta orientación se fortalece el pensamiento lógico y creativo. Un estudiante adquiere habilidades complejas para programar y para abordar problemas de forma integral; los estudiantes producen acondicionan y optimizan el software en cualquier área, como la computación gráfica, el procesamiento de imágenes, la simulación, la realidad virtual, la animación entre otras.

## OBJETIVO GENERAL

Solucionar problemas de sistematización de información a través de la programación modular, usando estructuras de datos incursionando en aplicaciones de la Programación Orientada a Objetos para observar sus ventajas en el desarrollo de software.

## COMPETENCIA GLOBAL

Al terminar el curso el estudiante estará en la capacidad de diseñar soluciones a problemas complejos empleando los siguientes conceptos y herramientas así:

- Programación modular.
- Almacenamiento de información en Archivos.
- Creación de estructuras de datos cuyos componentes pueden ser de tipos diferentes de datos
- Creación de los programas para modelar los objetos del mundo real de un modo más eficiente, a través de la Programación Orientada a Objetos(POO).

El uso no autorizado así como la reproducción total o parcial de su contenido por cualquier persona o entidad, estará en contra de los derechos de autor.



<b>CONTENIDO PROGRAMÁTICO</b>	Fecha Emisión: 2018/02/09	AC-GA-F-8
	Revisión No. 3	Página 2 de 9

<b>COMPETENCIAS ESPECÍFICAS</b>
1. Desarrolla un aplicativo para dar solución a un problema relacionado con la vida real utilizando programación estructurada y programación orientada a objetos
2. Reconoce y relaciona información mediante el uso de tipos de datos abstractos
3. Implementa un programa aprovechando las ventajas del concepto de funciones
4. Reconoce y maneja la sintaxis de un lenguaje de programación
5. Entiende los conceptos necesarios para el manejo de datos estructurados
6. Modela soluciones mediante la aplicación de los conceptos de la programación orientada a objetos
7. Reconoce los conceptos básicos de la computación gráfica.
8. Propone alternativas de solución a un problema
9. Implementa aplicaciones como expresión de una idea
10. Indica las instrucciones básicas para la elaboración de un programa
11. Expresa inquietudes con respecto a aspectos semánticos y sintácticos del lenguaje
12. Detecta y corrige los errores de lógica y sintaxis obtenidos al depurar un programa
13. Formula, argumenta y da soluciones para la creación de un aplicativo orientado a objetos
14. Manifiesta interés en aprender de forma autónoma
15. Desarrolla habilidades para buscar, procesar y analizar información procedente de diferentes fuentes

<b>CONTENIDO</b>
<p><b>Unidad Modular 1. Funciones y Recursividad</b></p> <ul style="list-style-type: none"> <li>• <i>Funciones de paso por valor</i></li> <li>• <i>Funciones de paso por referencia</i></li> <li>• <i>Arreglos unidimensionales</i></li> <li>• <i>Arreglos bidimensionales</i></li> <li>• <i>Arreglos como parámetros de funciones.</i></li> <li>• <i>Recursividad.</i></li> </ul>
<p><b>Unidad Modular 2. Tipos de Datos Abstractos (TDA)</b></p> <ul style="list-style-type: none"> <li>• <i>Concepto de tipo abstracto</i></li> <li>• <i>Implementación de un TDA.</i></li> <li>• <i>Operaciones de inserción y extracción de información.</i></li> <li>• <i>Tipos abstractos de datos como parámetros de funciones.</i></li> <li>• <i>Uso de archivos de datos</i></li> <li>• <i>Cadenas de caracteres.</i></li> </ul>
<p><b>Unidad Modular 3. Programación Orientada a Objetos</b></p> <ul style="list-style-type: none"> <li>• <i>Definición y características de la POO</i></li> <li>• <i>Modelado e identificación de objetos</i></li> <li>• <i>Declaración de una clase</i></li> <li>• <i>Declaración de objetos de clases</i></li> </ul>

El uso no autorizado así como la reproducción total o parcial de su contenido por cualquier persona o entidad, estará en contra de los derechos de autor.



<b>CONTENIDO PROGRAMÁTICO</b>	Fecha Emisión: 2018/02/09	AC-GA-F-8
	Revisión No. 3	Página 3 de 9

- Declaración de métodos
- Constructores destructores
- Propiedades fundamentales de POO.
- Herencia
  - Simple
  - Múltiple
- Polimorfismo
- Encapsulamiento.

### UNIDAD MODULAR 3. GRÁFICAS Y PUERTOS DE COMUNICACIÓN

- Representación de imágenes como mapas de bits e imágenes vectoriales
- Proyecto de Gráficos Bidimensionales
- Asociación de eventos a la geometría
- Utilización de primitivas básicas
- Generalidades de los puertos de comunicación del computador
- Laboratorio puerto USB.

Semana	Tema o actividad presencial	Actividades de trabajo independiente
1 (27 Ene.- 29 Ene.)	Presentación. Concepto y tipos de arreglos.	Revisión de los conceptos necesarios para dar inicio al programa. Presentación por parte del docente. Lectura en el aula virtual: Manual de Arreglos. Ejercicios de aplicación.
2 (01 Feb. - 05 Feb.)	Concepto de función Concepto de arreglo y función, concepto de variable y paso por referencia	Taller de arreglos con funciones (Aula Virtual) Ejercicios de aplicación.
3 (08 Feb. - 12 Feb.)	Sintaxis, punteros y arreglos, aritmética de punteros. Concepto de recursividad	Lectura en el aula virtual: Manual sobre funciones página 20 Recursividad. Exposición del tema y ejemplos de aplicación.
4 (15 Feb. - 19 Feb.)	Cadenas. Definición y Uso de los TDA en programación	Lectura "Registros y archivos" (págs. 112-113) en el aula virtual. Guía Práctica de laboratorio No. 2. Taller Tipos de datos abstractos
5 (22 Feb. - 26 Feb.)	<b>Primer Parcial</b>	
	TDA como arreglos de una y dos dimensiones.	Lectura "Registros y archivos" (págs. 112-113) en el aula virtual. Guía Práctica de laboratorio No. 2
6 (01 Mar. - 05 Mar.)	Conceptos fundamentales de POO: concepto de clase, visibilidad de miembros, objeto, instanciamiento, encapsulamiento.	Lectura "Manual de Programación 2" (págs. 127-132) del Aula Virtual, Guía de laboratorio No. 3
7 (08 Mar. - 12 Mar.)	Creación e inserción de datos, Archivos	Lectura "Registros y archivos" (págs. 115-120) en el aula virtual. Guía Práctica de laboratorio No. 2 Lectura "Registros y archivos" (págs. 115-120) en el aula virtual.
	Modelado e identificación de objetos,	Ejercicios de aplicación

El uso no autorizado así como la reproducción total o parcial de su contenido por cualquier persona o entidad, estará en contra de los derechos de autor.

# UNIVERSIDAD MILITAR NUEVA GRANADA



<b>CONTENIDO PROGRAMÁTICO</b>	Fecha Emisión: 2018/02/09	AC-GA-F-8
	Revisión No. 3	Página 4 de 9

8 (15 Mar. – 19 Mar.)	declaración de clases, métodos. Conceptos de objetos, C++ Concepto de función, atributo y clase	
9 (23 Mar. – 26 Mar.)	Concepto de Constructor, Destruyores de una Clase. Encapsulamiento.	Ejercicios de aplicación
(29 Mar. – 04 Abr.)	<b>Semana Santa</b>	
10 (05 Abr. – 09 Abr.)	Herencia	Ejercicios de aplicación. Guía Práctica de laboratorio No.4. Lectura "Manual de Programación 2" (págs. 155-165) del Aula Virtual. Taller de Herencia (Aula Virtual) Ejemplos de aplicación.
11 (12 Abr. – 16 Abr.)	<b>Segundo Parcial</b>	
	Polimorfismo	Guía Práctica de laboratorio No.5. Lectura "Manual de Programación 2" (págs. 96-98) del Aula Virtual. Taller polimorfismo (Aula Virtual) Lectura "Manual de Programación 2" (págs. 160-165) del Aula Virtual.
12 (19 Abr. – 23 Abr.)	Polimorfismo	Elaboración de los talleres propuestos para el tema.
13 (26 Abr. – 30 Abr.)	Gráficos Bidimensionales	Ejercicios de aplicación
14 (03 Mayo - 07 Mayo)	Gráficos 2D y 3D	Laboratorio 3D
15 (10 Mayo - 14 Mayo)	Gráficos 2D y 3D	Ejercicios de aplicación
16 (18 Mayo - 21 Mayo)	Puertos de comunicación del computador.	Laboratorio puerto USB
(24 Mayo - 28 Mayo)	<b>Examen Final</b>	
(31 Mayo - 04 Jun.)	<b>Socialización de notas</b>	

## SISTEMA DE EVALUACIÓN

**Rubricas Generales de la Asignatura:**

	EXCELENTE	BUENO	ACEPTABLE	DEFICIENTE
<b>CORTE I</b>				
<b>Desarrolla un programa para dar</b>	Desarrolla correctamente la	Desarrolla con algunos errores la solución a	Desarrolla con muchos errores la solución a	Ningún planteamiento para

El uso no autorizado así como la reproducción total o parcial de su contenido por cualquier persona o entidad, estará en contra de los derechos de autor.

# UNIVERSIDAD MILITAR NUEVA GRANADA



<b>CONTENIDO PROGRAMÁTICO</b>	Fecha Emisión: 2018/02/09	AC-GA-F-8
	Revisión No. 3	Página 5 de 9

<i>solución a un problema utilizando estructuras de datos de una y dos dimensiones empleando funciones de usuario de paso por valor y referencia, además, utiliza los conceptos de recursividad.</i>	<i>solución a un problema aplicando estructuras de datos de una y dos dimensiones y emplea adecuadamente funciones de usuario de paso por valor y referencia en un lenguaje de programación</i>	<i>un problema aplicando estructuras de datos de una y dos dimensiones y emplea adecuadamente funciones de usuario de paso por valor y referencia en un lenguaje de programación</i>	<i>un problema aplicando estructuras de datos de una y dos dimensiones y emplea adecuadamente funciones de usuario de paso por valor y referencia en un lenguaje de programación</i>	<i>la solución del problema</i>
<b>CORTE II</b>				
<b>Desarrolla un programa en un lenguaje de programación para dar solución a un problema utilizando correctamente Tipos de Datos Abstractos (TDA) y archivos.</b>	<i>Desarrolla e implementa eficazmente el paradigma de los TDA, en todo tipo soluciones informáticas con funciones, arreglos y archivos.</i>	<i>Desarrolla e implementa algunos conceptos de los TDA, en soluciones informáticas con funciones, arreglos y archivos.</i>	<i>Desarrolla e implementa pocos conceptos de los TDA, en soluciones informáticas con funciones, arreglos y archivos.</i>	<i>Ningún planteamiento para la solución del problema.</i>
<b>CORTE III</b>				
<b>Aplica los conceptos de Programación Orientada a Objetos, empleando los conceptos de instanciamiento, encapsulamiento, constructores, clases, clase base y clase derivada, herencia y polimorfismo en diferentes contextos o problemáticas planteadas.</b>	<i>Domina y aplica los conceptos de Programación Orientada a Objetos, empleando instanciamiento, encapsulamiento, constructores, clases, clase base y clase derivada, herencia y polimorfismo en diferentes contextos o problemáticas planteadas.</i>	<i>Aplica algunos de los conceptos de Programación Orientada a Objetos, empleando instanciamiento, encapsulamiento, constructores, clases, clase base y clase derivada, herencia y polimorfismo en diferentes contextos o problemáticas planteadas.</i>	<i>Utiliza pocos de los conceptos de Programación Orientada a Objetos, empleando instanciamiento, encapsulamiento, constructores, clases, clase base y clase derivada, herencia y polimorfismo en diferentes contextos o problemáticas planteadas.</i>	<i>Ningún planteamiento para la solución del problema.</i>
<b>Aplica de manera eficiente las funciones primitivas para el trazado de gráficos</b>	<i>Domina y utiliza las funciones primitivas para el trazado de gráficos</i>	<i>Reconoce los conceptos básicos para el trazado de gráficos</i>	<i>Identifica el sistema de coordenadas cartesianas para el trazo de gráficos y configura el ambiente de trabajo gráfico</i>	<i>Ningún planteamiento para el trazado de gráficos</i>

Acorde con la metodología y los porcentajes definidos por la Universidad para cada uno de los cortes del semestre, se realiza la evaluación del curso entre 0.0 y 5.0. La calificación del curso será la que resulte de la ponderación de cada corte y será aprobado si cumple los requisitos del reglamento estudiantil.

El uso no autorizado así como la reproducción total o parcial de su contenido por cualquier persona o entidad, estará en contra de los derechos de autor.

# UNIVERSIDAD MILITAR NUEVA GRANADA



<b>CONTENIDO PROGRAMÁTICO</b>	Fecha Emisión: 2018/02/09	AC-GA-F-8
	Revisión No. 3	Página 6 de 9

Se presentarán dos notas parciales cada una con un valor de 30% y un examen final correspondiente a un 40%. En cada corte se realizarán quices, talleres o entrega de tareas correspondientes a un 40% de la nota parcial.

Para la evaluación de la asignatura se tendrán en cuenta los criterios que se muestran en la siguiente tabla:

Actividad	Primer Corte (30%)		Segundo Corte (30%)		Tercer Corte(40%)	
	Cantidad	Valor	Cantidad	Valor	Cantidad	Valor
Quices (Individual)	2	25%	2	25%	2	10%
Talleres	2	25%	2	25%	2	10%
			Trabajo de Aplicación - Opcional Sustentación - Procedimental - Actitudinal (La sustentación de un trabajo se considera como un quiz)			30%
Examen parcial		50%		50%		50%
Total	100%		100%		100%	

\* Mínimo tres actividades por corte

## BIBLIOGRAFÍA

### Textos Guía:

- Joyanes, L. 2014, Programación en C, JAVA y UML, 2a Ed, McGraw-Hill.
- Joyanes, L. 2015, Fundamentos generales de programación, 1a Ed, McGraw-Hill.
- Deitel, H. 2009, Cómo programar en C++. 6a ed. Pearson Education. México.

### Textos Complementarios:

- Reyes J. 2018, Programación en C++. 3a ed. IT Campus Academy.
- Perez M. 2014, Programación orientada a objetos y programación estructurada, Createspace Independent Pub.

## MATERIAL COMPLEMENTARIO DE APRENDIZAJE PARA ESTUDIANTES

### 1. Glosario

Definiciones tomadas de: [https://en.wiktionary.org/wiki/Appendix:Glossary\\_of\\_computer\\_programming](https://en.wiktionary.org/wiki/Appendix:Glossary_of_computer_programming)

El uso no autorizado así como la reproducción total o parcial de su contenido por cualquier persona o entidad, estará en contra de los derechos de autor.



<b>CONTENIDO PROGRAMÁTICO</b>	Fecha Emisión: 2018/02/09	AC-GA-F-8
	Revisión No. 3	Página 7 de 9

**Clase abstracta**

Una clase que no se puede construir directamente, que se puede construir solo mediante la construcción de algunas de sus subclases.

**Aplicación**

Una aplicación que se ejecuta en un pequeño dispositivo de mano.

**Argumento**

Un valor, o referencia a un valor, pasado a una función; un argumento real.

**Booleano**

Un tipo de datos para sí o no, valores verdaderos o falsos.

**Clase**

Un conjunto de objetos que tienen el mismo comportamiento (pero generalmente difieren en el estado) o una plantilla que define dicho conjunto.

**Código fuente**

Instrucciones legibles por humanos en un lenguaje de programación, para ser transformadas en instrucciones de máquina por un compilador, intérprete, ensamblador u otro sistema similar.

**Compilador**

Un programa de computadora que transforma el código fuente en código objeto.

**Constructor**

Un método de clase (en programación orientada a objetos) que crea e inicializa cada instancia de un objeto.

**Punto flotante**

Un método para representar números reales como un par de enteros (la mantisa y característica)

**Función**

Una rutina que recibe cero o más argumentos y puede devolver un resultado.

**Método**

En lenguajes orientados a objetos, una subrutina o función que pertenece a una clase u objeto.

**Módulo**

Un programa que está vinculado con otros para formar una aplicación en funcionamiento; un método para implementar una subrutina

**Objeto**

Una instancia de una clase.

**Orientado a objetos**

Usar entidades llamadas objetos que pueden procesar datos e intercambiar mensajes con otros objetos.

**Paradigma**

Un estilo fundamental de programación de computadoras al que el diseño de un lenguaje de programación típicamente tiene que atender.

**Parámetro**

Un nombre en una definición de función o subrutina que se reemplaza por, o se vincula con, el argumento real correspondiente cuando se llama a la función o subrutina.

**Procedimiento**

Una subrutina o función codificada para realizar una tarea específica.



<b>CONTENIDO PROGRAMÁTICO</b>	Fecha Emisión: 2018/02/09	AC-GA-F-8
	Revisión No. 3	Página 8 de 9

<p><b>Tipo abstracto</b> Un tipo en un sistema de tipo nominativo que no puede ser instanciado.</p>
<p><b>2. Preguntas de repaso</b></p> <p>¿Cuáles son los tipos de parámetros que se pueden utilizar en una función?          ¿Qué es un arreglo?          ¿Qué es recursividad?          ¿Qué son los tipos abstractos de datos?          ¿Cómo se ingresan o extraen datos de un archivo?          ¿Qué es la Programación Orientada a Objetos?          ¿Qué es una clase y un objeto?          ¿A qué se refiere el concepto de herencia en programación orientada a objetos?          ¿Qué es polimorfismo?          ¿Cómo se pueden realizar gráficos en el lenguaje de programación?          ¿Cómo se puede implementar el uso de los puertos de comunicaciones en la programación?</p>
<p><b>3. Material Multimedia</b></p> <p><a href="http://www.youtube.com/watch?v=ofemmXa_M5o&amp;feature=relmfu">http://www.youtube.com/watch?v=ofemmXa_M5o&amp;feature=relmfu</a></p>
<p><b>4. Enlaces en la Red:</b></p> <p><a href="http://www.abcdatos.com/tutoriales/tutorial/1257.html">www.abcdatos.com/tutoriales/tutorial/1257.html</a>  <a href="http://www.lawebdelprogramador.com">http://www.lawebdelprogramador.com</a>  <a href="http://www.abcdatos.com/tutoriales/tutorial/1257.html">http://www.abcdatos.com/tutoriales/tutorial/1257.html</a>  <a href="http://es.wikipedia.org/wiki/C%2B%2B">http://es.wikipedia.org/wiki/C%2B%2B</a>  <a href="http://www.monografias.com/trabajos34/programas-visual-c/programas-visual-c.shtml">http://www.monografias.com/trabajos34/programas-visual-c/programas-visual-c.shtml</a></p>
<p><b>5. Curso Virtual:</b>  <a href="http://virtual2.umng.edu.co/">http://virtual2.umng.edu.co/</a></p>

## COMPETENCIA DEL DOCENTE

*El perfil del docente de cátedra u ocasional encargado de orientar la asignatura debe tener una formación en pregrado o posgrado en Ingeniería de Sistemas, con experiencia mínimo de un (1) año como docente de tiempo completo en instituciones de educación superior o su equivalente en medio tiempo u hora de cátedra y tener experiencia en la enseñanza de programación orientada a objetos y de los temas anteriormente mencionados, apoyándose en las TIC como herramientas mediadoras del proceso de enseñanza-aprendizaje.*

*El docente público de carrera debe tener formación profesional en Ingeniería de Sistemas, Ingeniería en Computación, Ingeniería Informático o Ingeniería de Software y/o con estudios de maestría o doctorado en Sistemas, TIC, Software, Informática o Educación. Debe tener experiencia docente mínimo de un (1) año como docente de tiempo completo en instituciones de educación superior o su equivalente en medio tiempo u hora de cátedra. Además, tener experiencia en la enseñanza de programación orientada a objetos y de los temas anteriormente mencionados, apoyándose en las TIC como herramientas mediadoras del proceso de enseñanza-aprendizaje.*

*Nota. Para los docentes Públicos de Carrera, el perfil se encuentra determinado en las convocatorias de las Facultades.*

# UNIVERSIDAD MILITAR NUEVA GRANADA



<b>CONTENIDO PROGRAMÁTICO</b>	Fecha Emisión: 2018/02/09	AC-GA-F-8
	Revisión No. 3	Página 9 de 9

## CONTROL DE CAMBIOS

CAMBIO REALIZADO	JUSTIFICACIÓN DEL CAMBIO	ACTA DE APROBACIÓN
<i>Actualización del Contenido</i>	<i>Por cambio de formato</i>	
<i>Actualización</i>	<i>Inclusión de dos líneas en el cronograma (Tercera Evaluación Parcial y Socialización) y se incluyen nuevos programas académicos</i>	<i>Acta N°06 del Comité de Currículo y Autoevaluación de la FCCBA. Julio 31 de 2018.</i>
<i>Actualización competencias</i>	<i>Requerimientos Institucionales</i>	<i>Acta # 11 del Claustro Docente de Diciembre 13 de 2018</i>
<i>Actualización Contenido programático.</i>	<i>Inicio periodo académico 2019-1</i>	<i>Acta 9 de diciembre 19 de 2018 del Comité de Currículo y Autoevaluación de la Facultad.</i>
<i>Inclusión de rúbricas</i>	<i>Requerimientos Institucionales</i>	<i>Acta 04 de abril 8 de 2019 del Comité de Currículo y Autoevaluación de la Facultad.</i>
<i>Actualización del Contenido</i>	<i>Incluir en el contenido antes del cronograma el contenido programático de la asignatura.</i>	<i>Acta 05 de mayo 27 de 2019 del Comité de Currículo y Autoevaluación de la Facultad.</i>
<i>Actualización del Contenido</i>	<i>Modificación en el cronograma de las fechas para el semestre 2019-2</i>	<i>Acta # 8 del Comité Asesor y Curricular del DEPTEC. Junio 10 del 2019.</i>
<i>Actualización de fechas y modificación de porcentajes de evaluación</i>	<i>Modificación en el cronograma de las fechas para el semestre 2020-1, ajuste de porcentaje en las actividades.</i>	<i>Acta # 22 del Comité Asesor y Curricular del DEPTEC. Diciembre 2 de 2019.</i>
<i>Actualización Contenido programático.</i>	<i>Inicio periodo académico 2020-1</i>	<i>Acta No. 11 de diciembre 18 de 2019 Comité de Currículo y Autoevaluación de la Facultad.</i>
<i>Actualización de fechas y Docentes que imparten la asignatura</i>	<i>Modificación en el cronograma de las fechas y docente para el semestre 2020-2.</i>	<i>Acta #8 de reunión Virtual del Comité Asesor y Curricular del DEPTEC. Junio 8 de 2020</i>
<i>Actualización Contenido programático.</i>	<i>Inicio periodo académico 2020-2</i>	<i>Acta No. 06 de junio 18 de 2020 Comité de Currículo y Autoevaluación de la Facultad.</i>
<i>Actualización de fechas y Docentes que imparten la asignatura</i>	<i>Modificación en el cronograma de las fechas y docente para el semestre 2021-1.</i>	<i>Acta #19 de reunión Virtual del Comité Asesor y Curricular del DEPTEC. Noviembre 26 de 2020</i>
<i>Actualización de fechas para el periodo 2021-1</i>	<i>Modificación de fechas para el primer semestre</i>	<i>Acta 11 de Comité de Currículo de Facultad, diciembre 15 de 2020</i>

El uso no autorizado así como la reproducción total o parcial de su contenido por cualquier persona o entidad, estará en contra de los derechos de autor.