UNIVERSIDAD DEL AZUAY

FACULTAD DE CIENCIAS DE LA ADMINISTRACIÓN
ESCUELA DE INGENIERÍA EN CIENCIAS DE LA COMPUTACIÓN

1. Datos generales

Materia: FUNDAMENTOS DE PROGRAMACIÓN
Código: ICC104
Paralelo: A
Periodo : Agosto-2024 a Diciembre-2024
Profesor: ORTEGA CHASI PATRICIA MARGARITA
Correo electrónico: portega@uazuay.edu.ec

Nivel: 1

Distribución de horas.

Docencia	Práctico	Autónomo:120		Total horas
		Sistemas de tutorías	Autónomo	
48	32	16	104	200

Prerrequisitos:

Ninguno

2. Descripción y objetivos de la materia

Su objetivo es capacitar a los estudiantes en el análisis de un problema, su solución mediante la elaboración de algoritmos representados en diagramas de flujo y pseudocódigo, y su codificación en el lenguaje de programación.

Contribuye de forma transversal con asignaturas como sistemas operativos, base de datos, inteligencia artificial, entre otras.

Fundamentos de Programación es una materia de gran importancia en la carrera de Ingeniería de Ciencias de la Computación porque constituye uno de los ejes de formación profesional del futuro Ingeniero, sienta las bases para el eje de formación de lenguajes de programación.

3. Objetivos de Desarrollo Sostenible

4. Contenidos

1	Algoritmos
1.1	Introducción
1.2	Concepto y características.
1.3	Ejercicios de aplicación
1.4	Herramientas para representar algoritmos
1.4.1	Pseudo-código
1.5	Constantes y variables, Tipos de datos, Operadores y Expresiones
1.5.1	Entero, real, caracter, cadena, booleano, definidos por el usuario.
1.5.2	Asignación, aritméticos, lógicos, relacionales, prioridad de operadores

1.5.3	Entrada y salida de datos
1.6	Estructuras de control (Programas estructurados)
1.6.1	Secuencial
1.6.2	Condicionales
1.6.3	Repetitivas
2	Introducción a lenguajes de programación
2.1	Software: Conceptos, Software del sistema y software de aplicaciones. Los lenguajes de programación: concepto, lenguaje de máquina, ensamblador y lenguaje de alto nivel
2.2	Editor de programas, compiladores, traductores y depurador de programas
2.3	Entornos de programación
2.4	Sintaxis y semántica de los lenguajes de programación.
2.4.1	Estructura general de un programa
2.5	Constantes y variables, Tipos de datos, Operadores y Expresiones
2.6	Estructuras de control
2.6.1	Secuencial
2.6.2	Condicionales
2.6.3	Repetitivas
3	Funciones y procedimientos
3.1	Concepto, características y definición
3.2	Ámbito de las variables: globales y locales
3.3	Paso de parámetros por valor y referencia
4	Arreglos unidimensionales y multidimensionales.
4.1	Concepto, características y definición
4.2	Arreglos unidimensionales
4.3	Ordenamiento y búsqueda
4.4	Arreglos bidimensionales
4.5	Arreglos multidimensionales
4.6	Arreglo como parámetro
4.7	Cadenas de caracteres
4.8	Punteros.
5	Estructuras de datos
5.1	Concepto, características y definición
5.2	Arreglos de estructuras
6	Flujos y archivos.
6.1	Concepto, características y definición
6.2	Operaciones sobre archivos (lectura y escritura)

5. Sistema de Evaluación

Resultado de aprendizaje de la carrera relacionados con la materia

Resultado de aprendizaje de la materia	Evidencias
ad1. Resuelve problemas básicos de ingeniería mediante la aplicación de un lenguaje de consulta estructurado.	
-Implementa funciones que contengan estructuras de control aprendidas en	-Evaluación escrita

Resultado de aprendizaje de la carrera relacionados con la materia Resultado de aprendizaje de la materia	Evidencias
este curso.	
-Implementa proyectos que integren los conceptos aprendidos, expresados en un lenguaje de alto nivel con la ayuda de una herramienta de programación.	-Evaluación escrita
-Reconoce la importancia de las funciones como herramienta para simplificar la estructura de un programa.	-Evaluación escrita
-Resuelve problemas básicos de ingeniería aplicando el conocimiento y correcta utilización de estructuras de control.	-Evaluación escrita
-Utiliza archivos de texto para el almacenamiento de información.	-Evaluación escrita

Desglose de evaluación

Evidencia	Descripción	Contenidos sílabo a evaluar	Aporte	Calificación	Semana
Evaluación escrita	Pruebas y ejercicios de aplicación	Algoritmos, Introducción a lenguajes de programación	APORTE	10	Semana: 4 (16/09/2024 al 21/09/2024)
Evaluación escrita	Pruebas y ejercicios de aplicación	Arreglos unidimensionales y multidimensionales., Funciones y procedimientos	APORTE	10	Semana: 8 (14/10/2024 al 19/10/2024)
Evaluación escrita	Pruebas y ejercicios de aplicación	Estructuras de datos, Flujos y archivos.	APORTE	10	Semana: 12 (11/11/2024 al 13/11/2024)
Evaluación escrita	Examen evaluación práctico	Algoritmos, Arreglos unidimensionales y multidimensionales., Estructuras de datos, Flujos y archivos., Funciones y procedimientos, Introducción a lenguajes de programación	EXAMEN	20	Semana: 15 (02/12/2024 al 03/12/2024)
Evaluación escrita	Examen evaluación práctico	Algoritmos, Arreglos unidimensionales y multidimensionales., Estructuras de datos, Flujos y archivos., Funciones y procedimientos, Introducción a lenguajes de programación	SUPLETORIO	20	Semana: 17-18 (15-12-2024 al 21-12-2024)

Metodología

Descripción	Tipo horas
El curso se estructura mediante una metodología que combina el trabajo autónomo del estudiante con la instrucción y guía proporcionada por el docente. Esta combinación tiene como objetivo desarrollar tanto el conocimiento teórico como las habilidades prácticas necesarias para dominar los conceptos básicos de programación. Práctica Individual: Los estudiantes dedicarán tiempo a practicar la escritura de código de manera individual, utilizando ejercicios y tareas proporcionadas por el curso. Este componente se enfoca en la resolución de problemas y la implementación de algoritmos básicos, fomentando la experimentación y la corrección de errores para mejorar la comprensión y la destreza técnica. Autoestudio Teórico: Además de la práctica de programación, se espera que los estudiantes revisen materiales teóricos de manera independiente, incluyendo lecturas asignadas, videos explicativos y documentación técnica. Este estudio autónomo permitirá a los estudiantes familiarizarse con los conceptos fundamentales antes de aplicarlos en ejercicios prácticos. Proyectos y Ejercicios Autodirigidos: Los estudiantes trabajarán en proyectos pequeños y ejercicios que integren los conocimientos adquiridos, aplicando diferentes estructuras de control, funciones, y manipulación de datos. Estas actividades estarán diseñadas para consolidar el aprendizaje y preparar a los estudiantes para desafíos más complejos.	Autónomo
Clases Magistrales y Explicaciones: El docente impartirá clases magistrales que cubrirán los conceptos esenciales de la programación, como estructuras de control, tipos de datos, y funciones. Durante estas sesiones, se realizarán demostraciones en vivo de codificación, mostrando cómo aplicar las teorías a situaciones prácticas. Sesiones Prácticas Guiadas: En los talleres o laboratorios, el docente guiará a los estudiantes a través de ejercicios prácticos, proporcionando asistencia inmediata y retroalimentación en tiempo real. Estas sesiones están diseñadas para reforzar los conceptos aprendidos y ayudar a los estudiantes a superar obstáculos comunes en la programación. Evaluación Formativa y Retroalimentación: Se implementarán evaluaciones formativas, como quizzes y pequeños proyectos, para monitorear el progreso de los estudiantes. El docente proporcionará retroalimentación continua, destacando áreas de mejora y reforzando los conceptos que requieren mayor atención.	Total docencia

Criterios de evaluación

Descripción	Tipo horas
La evaluación del componente autónomo se enfocará en la calidad y precisión del código escrito por los estudiantes, así como en su progreso y consistencia en la práctica individual. Se valorará su capacidad para implementar algoritmos correctamente, resolver problemas programáticos y aplicar los conceptos teóricos en contextos prácticos. Además, se considerará la mejora continua y el esfuerzo demostrado en completar las tareas asignadas a lo largo del curso.	Autónomo
La evaluación del componente docente se basará en la participación activa del estudiante durante las clases y talleres, y su desempeño en las evaluaciones formativas y sumativas. Se valorará la contribución del estudiante en discusiones, su disposición para aplicar la retroalimentación recibida, y su capacidad para demostrar un entendimiento sólido de los conceptos a través de actividades supervisadas y examenes realizados en clase. Política de Uso de Inteligencia Artificial (IA) En el curso de Fundamentos de Programación, se permite el uso de herramientas de Inteligencia Artificial (IA) como apoyo para el aprendizaje y la resolución de problemas, siempre y cuando su uso sea responsable y no comprometa la integridad del proceso de aprendizaje. A continuación, se detallan las directrices para el uso de IA en este curso: Uso Auxiliar y Complementario: Las herramientas de IA pueden ser utilizadas para complementar el aprendizaje, por ejemplo, en la revisión de código, la corrección de errores sintácticos o la generación de ideas para resolver problemas. Sin embargo, el trabajo principal, como la escritura y depuración de código, debe ser realizado por el estudiante de manera autónoma. Transparencia en el Uso: Los estudiantes deben ser transparentes acerca del uso de IA en sus trabajos. Si una herramienta de IA se utiliza para asistir en la resolución de un problema o en la escritura de código, esto debe ser indicado claramente en la presentación o entrega del ejercicio. La transparencia es clave para mantener la honestidad académica y la credibilidad del trabajo presentado. Ética y Originalidad: El uso de IA no debe sustituir el esfuerzo personal ni la creatividad del estudiante. Se prohíbe el uso de IA para generar soluciones completas o para automatizar la totalidad de las tareas programáticas. El objetivo del curso es que los estudiantes desarrollen sus propias habilidades de programación, y por lo tanto, cualquier dependencia excesiva de la IA será penalizada. Responsabilidad del Estudiante: El estudiante es responsable del contenido y la calidad de todo el trabajo presentado, independientemente del uso de herramientas de IA. Esto implica la obligación de revisar, entender y corregir el código creado con el soporte de IA, asegurando que sea preciso, funcional y adecuado para la tarea asignada. El incumplimiento de esta política puede resultar en una evaluación negativa del trabajo presentado y, en casos más graves, en medidas disciplinarias conforme a las normas académicas de la institución.	Total docencia

6. Referencias

Bibliografía base

Libros

Autor	Editorial	Título	Año	ISBN
JOYANES AGUILAR, LUIS	McGraw-Hill	FUNDAMENTOS DE PROGRAMACIÓN: Algoritmos, estructura de datos y objetos	2008	978-84-481-6111-8
Dale, Nell; Weems, Chip	McGraw-Hill	Programación y resolución de problemas con C++	2007	978-970-10-6110-7
Thomas H. Cormen, Charles E. Leiserson, Ronald L. Rivest, and Clifford Stein	MIT Press	Introduction to Algorithms	2009	9780262270830

Web

Autor	Título	Url
cplusplus	cplusplus	http://www.cplusplus.com/

Software

Autor	Título	Url	Versión
SourceForge	Dev-C++	https://sourceforge.net/projects/orwelldevcpp/	
Codeblocks	Codeblocks	http://www.codeblocks.org/downloads	
Dev-C++	Dev-C++	https://sourceforge.net/projects/orwelldevcpp/	

Revista

Bibliografía de apoyo

Libros

Web

Software

Revista

Docente

Director/Junta

Fecha aprobación: 19/08/2024

Estado: Aprobado