1. **Identificación de la asignatura**

**División:** Ingenierías

**Departamento:** Ingeniería de Sistemas.

**Nombre de la asignatura:** Algoritmia y Programación I

**Código de la asignatura: IST 2088**

**Nivel de la asignatura (Pregrado, Postgrado):** Pregrado.

**Requisitos (Código y nombre de la asignatura):** No tiene.

**Número de créditos de la asignatura:** 3

**No. de horas teóricas por semana:** 2

**No. de horas prácticas por semana:** 2

**No. de horas trabajo independiente por semana:** 6

**Número de semanas:** 16

**Idioma de la asignatura:** Español.

**(Español, Inglés, Alemán, francés, otros)**

**Modalidad de la asignatura:** Virtual.

**(Presencial, Virtual, Híbrido, otros)**

**Nombre del profesor:**

**NRC:**

**Contacto del profesor:**

**Horario de atención (si aplica):**

1. **Descripción de la asignatura.**

En este curso se estudian los conceptos básicos de las tecnologías de Software y Hardware situando a la máquina (el computador) como herramienta de apoyo en el proceso de solución de problemas. Al finalizar el curso, el estudiante estará en capacidad de construir algoritmos bajo el enfoque de la programación estructurada. Para lograr este objetivo se inicia con conceptos básicos dando paso a las estructuras secuenciales, y a las estructuras condicionales e iterativas, para luego pasar a la Programación Modular (Funciones), finalizando con los temas relacionados con las Estructuras de Datos.

1. **Justificación**

El significativo progreso científico y tecnológico, especialmente de la tecnología computacional, se evidencia con mayor frecuencia en la vida diaria y en el ejercicio profesional de los ciudadanos. La ingeniería, Geología y carreras afines son quizás de las áreas que registran la mayor aplicación de tecnologías computacionales para resolver los problemas fundamentales de cada una de las especialidades. Los profesionales de hoy, no importa cuál sea su área de actuación, deben ser capaces de comprender, operar, diseñar y desarrollar programas que contribuyan a la solución de estos problemas. El dominio de los conceptos, principios y técnicas computacionales ya no es una exclusividad de los profesionales de las Ciencias de la Computación o de la Ingeniería de Sistemas. Hoy en día no se concibe un profesional de la ingeniería, geología y afines que no sea capaz de dominar la tecnología computacional para crear o adaptar las soluciones que requiere para su trabajo. Por tanto, es de vital importancia para el estudiante de estas carreras desarrollar su capacidad de análisis y lógica para solucionar problemas propios de su profesión; para lograrlo es necesario el conocimiento de los conceptos fundamentales de informática, así como el dominio del proceso de la programación de computadores.

1. **Objetivo general de la asignatura.**

Este curso se orientará a: Desarrollar la lógica y el pensamiento analítico y sistémico para resolver problemas básicos de ingeniería mediante el diseño de algoritmos y su codificación en un lenguaje de programación.

**Objetivos específicos:**

OB1. Aplicar conceptos básicos de la matemática en la solución de problemas.

OB2. Aplicar la lógica algorítmica para definir una solución a un problema propuesto, en términos de entrada-proceso-salida.

OB3. Comprender e interpretar los algoritmos escritos en pseudo-código.

OB4. Diseñar algoritmos en pseudocódigo para solucionar problemas básicos.

OB5. Utilizar técnicas de pruebas de escritorio para verificar el funcionamiento de los algoritmos y corregirlos con base en los resultados de las pruebas efectuadas.

OB6. Implementar los algoritmos usando el lenguaje de programación.

OB7. Desarrollar actitudes y valores relacionados con:

* Honestidad: Actuar y exigir un comportamiento honesto.
* Respeto y Tolerancia: Cumplir las normas de conducta, respetar al otro y tolerar las diferencias.
* Responsabilidad: Realizar las actividades académicas con dedicación, cumpliendo con la calidad del trabajo propuesto dentro de los plazos estipulados.
* Perseverancia: Repasar los contenidos, realizar los ejercicios y hacer las lecturas antes de cada clase.
* Organización y planificación: Definir prioridades en la realización de tareas y elaborar un plan de acción que considere los recursos necesarios y las restricciones impuestas.

1. **Resultados de Aprendizaje.**

CO1. Evaluar expresiones aritméticas y lógicas, teniendo en cuenta la precedencia de los operadores y los tipos de variables.

* Evaluación: Un quis y 1er parcial.

CO2. Analizar el problema planteado y describirlo en términos de las entradas, el proceso y la salida.

* Evaluación: 1er Parcial

CO3. Diseñar, probar y corregir algoritmos que resuelvan problemas básicos

* Evaluación: 1° Parcial, 2do Parcial y Examen Final

CO4. Utilizar un lenguaje de programación para implementar los algoritmos.

Evaluación: Laboratorios

1. **Temas de la asignatura.**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Temas** | **Subtemas** | **No. de Horas a cargo del profesor** | **Semana** | **Trabajo independiente (describir las actividades)** |
| Presentación del curso | Análisis de las características del curso y de los principales elementos de la parcelación. Diagnóstico (opcional) | 2 | 1 | Analizar la parcelación del curso. |
| Programación Estructurada I (Estructuras Algorítmicas no repetitivas) | Introducción a la teoría de algoritmos:  -Conceptos básicos (variables, tipos de datos, operadores)  - Pasos para la elaboración de un algoritmo.  - Algoritmos de la Vida cotidiana | 4 | 1 y 2 | Hacer los ejercicios básicos  Analizar los algoritmos de la vida cotidiana |
| Estructura algoritmos secuenciales  Asignar, leer, escribir  Pruebas de escritorio  Ejercicios:   * Cálculo de la definitiva de una asignatura. * Conversiones * Áreas y perímetros de figuras geométricas * Descuentos en porcentajes * Aumentos en porcentajes   Implementar en el lenguaje de  Programación:  Estructuras secuenciales  Compilación  Ejecución  Corrección de errores en el lenguaje | 4 | 2 y 3 | Preparar el tema antes de la clase  Revisar los ejercicios realizados en clases  Hacer los ejercicios propuestos  Practicar en JAVA |
| Condicionales no repetitivos.  Si - Sino - Fin-si.  Si - Fin-si.  Dependiendo De.  Diferencias en el uso de las tres estructuras anteriores  Condicionales anidados  Ejercicios:   * Verificar si un número es par o impar * Aprobación de una asignatura * Menús * Ordenar tres números * Ubicar un punto del plano cartesiano   Implementación de Condicionales no repetitivos en el lenguaje de Programación. | 8 | 3, 4 y 5 | -Preparar el tema antes de la clase  -Revisar los ejercicios realizados en clases  -Hacer los ejercicios propuestos  -Practicar en JAVA |
| Programación Estructurada I (Estructuras Algorítmicas repetitivas) | Bloques condicionales repetitivos.  Mientras Que (MQ-Fin).  Ejercicios   * Descomponer un número en sus cifras. * Juegos (dados, cartas, otros) * Seleccionar el mayor y el menor) * Números perfectos, primos, amigos   Implementación en el lenguaje del ciclo MQ | 10 | 5, 6 y 7 | -Preparar el tema antes de la clase  -Revisar los ejercicios realizados en clases  -Hacer los ejercicios propuestos  -Practicar en JAVA |
| -Bloques condicionales repetitivos.  Haga-Hasta-Que – Fin.  Ejercicios   * Verificar si un número es primo   Implementación en el lenguaje de programación del ciclo Hacer Hasta | 4 | 8 |
| -Bloques condicionales repetitivos.  Para - Fin.  Ejercicios   * Verificar si un número es perfecto   Implementación en el lenguaje del ciclo Para | 4 | 9 |
| Series | 4 | 10 |
| Funciones | Definición  Sintaxis  Llamado  Ejercicios:   * Combinatoria * .   Implementación de funciones en el lenguaje de programación | 6 | 11 y 12 | Preparar el tema antes de la clase  -Revisar los ejercicios realizados en clases  -Hacer los ejercicios propuestos  -Practicar en JAVA |
| Estructuras de Datos | Arreglos unidimensionales  Ejercicios   * Leer un vector * Crear un vector con algunas especificaciones * Mayor y menor elemento de un vector. * Búsqueda de un elemento * Sumar dos vectores * Aplicaciones de problemas en ingeniería, geología y Ciencias de Datos.   Implementación de vectores en el lenguaje de programación | 10 | 12, 13 y 14 | Preparar el tema antes de la clase  -Revisar los ejercicios realizados en clases  -Hacer los ejercicios propuestos  -Practicar en JAVA |
| Arreglos bidimensionales  Ejercicios   * Leer un matriz * Crear una matriz con algunas especificaciones * Búsqueda de un elemento * Sumar dos matrices * Notas de estudiantes y promedios utilizando vectores para guardar código. * Aplicaciones de problemas en ingeniería, geología y Ciencias de Datos.   Implementación de matrices en el lenguaje de programación | 8 | 15 y 16 | Preparar el tema antes de la clase  -Revisar los ejercicios realizados en clases  -Hacer los ejercicios propuestos  -Practicar en JAVA |

1. **Evaluación**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Evaluación** | **%** | **Descripción** | **Fecha** | **Temas** (Todas las evaluaciones son acumulativas) |
| Primer Corte | 25\* | 15 % Examen Parcial (\*\*)  10% Quices, tareas, talleres | Semana 6  Del 7 al 12 de marzo | Todos los temas hasta condicionales (incluyendo Dependiendo De) |
| Prácticas de Laboratorio Corte\_1 | 15\* | 5 % Practica en Java Condicionales  10% Practica en Java Ciclos | Hasta Semana 10 | Todos los temas hasta ciclos. |
| Segundo Corte | 25 | 20 % Examen Parcial (\*\*)  5% Quices, tareas, talleres | Semana 12  Del 25 al 30 de abril | Todos los temas hasta ciclos (incluyendo sumatorias y productorias). |
| Prácticas de Laboratorio Corte\_2 | 10 | Prácticas de Arreglos de Datos y funciones en Java | Desde semana 11 hasta semana 16 | Arreglos de Datos y funciones. |
| Corte Final | 25 | 20 % Examen Final  5% Quices, tareas, talleres | Estipulada por Registro | Todos los temas incluidos en el contenido. |

**(\*) Componentes que corresponden al 40%.**

(\*\*) Los parciales se realizarán en la semana 6 (Primer Parcial: desde introducción a la teoría de algoritmos hasta estructuras condicionales incluyendo Dependiendo De) y semana 12 (Segundo Parcial: Ciclos simples, ciclos anidados, series (sumatorias y productorias)).

No se tolerará el **plagio** o la **copia**. Sin excepción, en caso de darse uno de estos casos, a los estudiantes involucrados se les iniciará proceso disciplinario, y se actuará conforme al Reglamento de Estudiantes de la Universidad del Norte. El plagio incluye usar contenidos sin la debida referencia, de manera literal o con mínimos cambios que no alteren el espíritu del texto. Adicionalmente, el estudiante debe tener presente que todos los exámenes, informes, trabajos, presentaciones, laboratorios y cualquier otro producto a evaluar debe:

* Ser realizado de forma individual, excepto, en aquellos casos en que el profesor lo autorice por escrito y sin ninguna clase de participación de terceros.
* Debe referenciar en el código fuente las librerías o algoritmos de terceros que hayan sido utilizados. Dar crédito a los autores de las ideas, conceptos teorías, datos en los que se basó el estudiante para realizar su trabajo.
* Debe enviarse por el medio definido por el docente. Si el trabajo debe ser enviado por catálogo web, sólo se recibirá por ese medio, y no por medios alternativos. En caso de no utilizar los medios establecidos para el desarrollo del curso, se asumirá que el trabajo no fue enviado.

1. **Bibliografía de la asignatura**

Bibliografía Básica

1. R. Capacho, R. Ebratt y A. Mancilla. *Diseño y construcción de algoritmos*. Barranquilla: Ediciones Uninorte, 2015. **TEXTO GUÍA.**
2. Capacho, G. León, *Estrategias para el diseño y construcción de algoritmos*. Editorial Académica Española – EAE, 2014.

[https://www.eae-publishing.com/catalog/details//store/es/book/978-3-659-04037-5](https://www.eae-publishing.com/catalog/details/store/es/book/978-3-659-04037-5)

1. L. Joyanes, Fundamentos de Programación: Algoritmos, Estructura de datos y objetos. Madrid: McGraw-Hill, 2003.

Bibliografía complementaria

1. C. Becerra, *Algoritmos: Conceptos básicos,* 7a ed. Bogotá: Kimpres, 2011.
2. R. Capacho, G. León, Estrategias para el diseño y construcción de algoritmos. Ciudad: Editorial Académica Española – EAE, 2014.
3. Code.org, *Introducción a la Informática*. Disponible en <http://learn.code.org/>
4. H. Deitel, *Cómo programar en JAVA:**Introducción a la programación Web con CGI y diseño orientado con UML a objetos*. 4 ed. México: Pearson Educación, 2003.
5. E. Oviedo, *Lógica de Programación*. Ciudad: Ecoe Ediciones. 2006.
6. PSeInt, Manuales y Documentación. 25,07, 2014. Disponible en: <http://pseint.sourceforge.net/>
7. A. Salazar, *Canal de videos para apoyar el desarrollo de la lógica algorítmica: Ejercicios en PSeINT* (Videos). Disponible en: <https://www.youtube.com/channel/UCK--whlbxyQok1u5XBf6Sew/featured>

Actualizada Julio 19 de 2018, Dpto. de Sistemas; 11:00 a.m.