



E.T. N° 21 - D.E.10

FRAGATA ESCUELA LIBERTAD

Asignatura: **ALGORITMOS Y ESTRUCTURAS DE DATOS**

Curso: **4º** División: **1ª** Ciclo: **Superior Computación** Turno: **Noche**

PROGRAMA

Unidad 0: Diagnóstico

Ejercicios sencillos de procedimientos lógicos. Uso básico de PC.

Bibliografía: Apuntes del profesor. Información web.

Unidad 1: Introducción a la programación.

Conceptos iniciales. Algoritmo. Programa. Lenguajes de programación. Codificación. Archivo fuente. Compilación. Archivo ejecutable. Diagramación y otros recursos de ayuda al programador.

Concepto de variable. Almacenamiento. Tipos de variables numéricas enteras y de coma flotante. Reglas para nombres de variables. Significancia. Nomenclaturas usuales. Declaración. Inicialización.

Procedimientos básicos. Asignación y modificación de variables. Operaciones algebraicas básicas; orden de prioridad. Ingreso de datos. Muestra de resultados. Simbología básica de diagramación. Resolución de algoritmos sencillos mediante diagramación. Pruebas de escritorio.

Codificación. Equivalencia lógica entre diagramación y codificación. Estructura general de un programa. Secuencialidad del código. Fin del programa. Ingreso de datos. Operaciones con variables. Precauciones con los tipos numéricos. Impresión por pantalla; posicionamiento y formato. Comentarios; formatos e importancia. Directivas de compilador.

Entorno de trabajo. Operaciones básicas: guardar, recuperar, nuevo, salir, imprimir, buscar, copiar y pegar. Obtención de ejecutables. Modos de compilación. Solución de errores de compilación. Ayuda incorporada. Codificación y prueba de programas.

Bibliografía: Apuntes del profesor. Información web. Libros y manuales sobre el lenguaje a usar.

Unidad 2: Estructuras de control.

Bifurcación condicional. Operadores de comparación. Diversos formatos. Bifurcaciones combinadas (casos 'Y' y 'O' inclusivo). Diagramación y codificación.

Lazos condicionales basados en contadores. Noción de contador. Alternativas de conteo. Precauciones. Diagramación y codificación.

Contadores y acumuladores. Estructura. Operaciones comunes: promedios, porcentajes, etc.

Lazos condicionales basados en condiciones lógicas. Diversos formatos. Precauciones. Diagramación y codificación.

Distribuidores. Diversos formatos. Precauciones. Diagramación y codificación.

Máximos y mínimos. Métodos de obtención de máximos y/o mínimos sin repetición.

Normas de estilo. Indentado de programas y otras normas de estilo. Documentación del código.

Resolución de problemas. Diagramación. Pruebas de escritorio; análisis de posibilidades y casos límites. Validaciones básicas. Codificación y prueba de programas.

Bibliografía: Apuntes del profesor. Información web. Libros y manuales sobre el lenguaje a usar.

Unidad 3: Variables y expresiones.

Variables numéricas. Tipos y rangos. Declaración e inicialización. Casting. Constantes. Directivas de compilador. Codificación.

Expresiones algebraicas. Módulo. Funciones numéricas de biblioteca: potencia, raíz cuadrada, parte entera, valor absoluto, trigonométricas, logarítmicas, etc.. Precedencias. Aplicaciones comunes: múltiplos, redondeos, etc. Codificación.

Variables de carácter. Cadenas de caracteres. Forma de almacenamiento. Precauciones. Acceso a caracteres individuales. Funciones de biblioteca para cadenas: longitud, copia, concatenación, comparación, etc. Codificación.

Expresiones lógicas. Operadores AND, OR y NOT. Combinaciones. Precedencia. Codificación.

Resolución de problemas. Diagramación. Pruebas de escritorio. Codificación y prueba de programas.

Bibliografía: Apuntes del profesor. Información web. Libros y manuales sobre el lenguaje a usar.

Unidad 4: Funciones.

Funciones y procedimientos. Necesidad. Ventajas. Características. Parámetros. Definición. Llamado. Prototipos. Alcance de variables. Variables globales. Transferencia de argumentos. Paso por valor y por referencia. Retorno de valores. Diagramación y codificación. Directivas de compilador.

Programación modular. Análisis descendente. Características. Ventajas. Limitaciones. Diagramación por bloques. Documentación del programa.

Resolución de problemas. Diagramación. Pruebas de escritorio. Codificación y prueba de programas.

Bibliografía: Apuntes del profesor. Información web. Libros y manuales sobre el lenguaje a usar.

Unidad 5: Estructuras de datos.

Vectores. Concepto. Elementos. Ventajas. Dimensionamiento. Carga, modificación y muestra. Precauciones en el uso. Cadena como vector. Codificación.

Operatoria con vectores. Métodos de ordenamiento. Búsquedas secuencial, ordenada y binaria. Agregado, eliminación e inserción de elementos. Máximos y mínimos con repetición.

Matrices. Concepto y aplicaciones. Ordenamiento. Búsquedas. Operaciones simples. Codificación.

Estructuras. Necesidad. Ventajas. Características. Carga, modificación y muestra. Codificación.

Nociones sobre punteros. Definición. Relación con los vectores. Codificación.

Resolución de problemas. Diagramación. Pruebas de escritorio. Codificación y prueba de programas.

Bibliografía: Apuntes del profesor. Información web. Libros y manuales sobre el lenguaje a usar.

Unidad 6: Archivos.

Archivo. Concepto. Utilidad. Tipos; archivos secuenciales, aleatorios y binarios. Características.

Archivos secuenciales. Nociones de campo y registro. Apertura y cierre. Escritura. Lectura. Fin de archivo. Búsquedas. Diagramación y codificación.

Archivos binarios. Apertura y cierre. Escritura. Lectura. Fin de archivo. Posicionamiento. Funciones para archivos. Aplicaciones. Diagramación y codificación.



E.T. N° 21 - D.E.10

FRAGATA ESCUELA LIBERTAD

Asignatura: **ALGORITMOS Y ESTRUCTURAS DE DATOS**

Curso: **4º** División: **1ª** Ciclo: **Superior Computación** Turno: **Noche**

Resolución de problemas. Diagramación. Pruebas de escritorio. Codificación y prueba de programas.

Bibliografía: Apuntes del profesor. Información web. Libros y manuales sobre el lenguaje a usar.

Objetivos generales de la asignatura:

Que los estudiantes sean capaces de:

- Conocer los conceptos de estructura de datos.
- Utilizar estrategias de análisis descendente para resolver problemas de programación.
- Diseñar algoritmos eficientes para la solución de problemas computacionales.
- Implementar los algoritmos diseñados en un lenguaje de programación acorde al paradigma.
- Utilizar las reglas de buenas prácticas de programación.
- Diseñar las estrategias de prueba para validación de algoritmos y de programas.
- Interpretar manuales técnicos del lenguaje de programación utilizado.
- Elaborar la documentación técnica del desarrollo del programa.

Criterios de evaluación:

El alumno será evaluado en todo su desempeño áulico. Si bien es cierto se realizaran evaluaciones en donde se objetivara la capacidad de comprensión del contenido sino también su participación en el aula. Las evaluaciones serán realizadas en un orden de complejidad cronológica según se avance en el transcurso del ciclo lectivo.

Material didáctico:

- Textos y/o apuntes de estudio.
- Guías de ejercitación.
- Pizarrón, marcadores, borrador.
- Elementos de dibujo para diagramación.
- Equipos de computación.
- Software de programación.

Criterios de acreditación:

Diseñar, confeccionar, implementar, probar y evaluar aplicaciones relacionadas con los temas de la materia.