|  |  |
| --- | --- |
|  |  |
| Contextualización de la materia | **1** |
|  |

**DEPARTAMENTO:** Ingeniería en Sistemas de Información

**CARRERA:** Ingeniería en Sistemas de Información

**NOMBRE DE LA ACTIVIDAD CURRICULAR:** ALGORITMOS Y ESTRUCTURA DE DATOS

Año Académico: Plan 2023

Área: Desarrollo de software

Bloque: Tecnologías Básicas

Nivel: 1º

Tipo: Obligatoria

Modalidad: Anual

RTF: 10

**Cargas horarias totales**:

|  |  |  |
| --- | --- | --- |
| *Horas reloj total* | *Horas cátedra total* | *Horas cátedra semanales* |
| 120 | 160 | 5 |

**COMPOSICIÓN DEL EQUIPO DE CÁTEDRA:**

Profesor Asociado y Coordinador: Dr.Mg. Oscar Bruno

Profesora Adjunta Mg. Yamila Zakhem

Profesor Adjunto Esp. Ing. Jose maria Sola

Profesor Adjunto: Mg. Lic. Marcelo Lipkin

Profesor Adjunto: Esp. Ing. Pablo Sznajdleder

Profesor Adjunto: Lic. Eduardo Zúñiga

Profesora Adjunta: Esp. Gabriela Sanromán

Profesora Adjunta Ing. Natalia Perez Lopez

Profesor Adjunto: Esp. Ing. Pablo Méndez

Profesora Adjunta: Ing. Roxana Leituz

Profesor Adjunto Esp. Ing. Adrian Fiore

Profesor Adjunto Ing. Diego Juan

Profesor Adjunto Ing. Federico Maidan

Profesor Adjunto Ing. Ricardo Ferre

ATP 1ra Ingeniero Mariano Bengochea

ATP 1ra Ingeniera Laura Lazzati

**COMPETENCIAS DE EGRESO GENÉRICAS A LAS QUE CONTRIBUYE:**

|  |  |  |  |
| --- | --- | --- | --- |
| Competencia | Baja | Media | Alta |
| Identificación, formulación y resolución de problemas de la disciplina. CG1 |  | x |  |
| Utilización de técnicas y herramientas de aplicación en la disciplina. CG4 |  |  | x |
| Desempeñarse de manera efectiva en equipos de trabajo CG6 |  |  | x |
| Fundamentos para el aprendizaje continuo.CG9 |  | x |  |

**OBJETIVOS (APRENDIZAJE/LOGROS A ALCANZAR)[[1]](#footnote-1)**

***Al finalizar con éxito la cursada de la asignatura “Algoritmos y estructura de datos” del DISI – UTN.BA los estudiantes serán capaces de:***

1. Desarrollar un pensamiento creativo que permita resolver situaciones problemáticas en forma apropiada y satisfactoria, con o sin el uso de computadoras.
2. Identificar la información contenida en una situación a resolver, utilizando un enfoque lógico y estructurado.
3. Diferenciar los problemas en términos de su eficiencia computacional en cuanto a recursos y tiempos de ejecución, en el paradigma imperativo.
4. Adquirir habilidades prácticas y experiencia en la escritura de programas en lenguajes formales de programación estructurada.
5. Aplicar el proceso de pensamiento sistemático, permita inferencias y toma de decisiones, para resolver problemas con la computadora.
6. Construir a partir de la propia experiencia, promoviendo la producción y la construcción del conocimiento de técnicas de resolución de problemas y programación.
7. Promover la construcción de programas en forma colaborativa en los diversos espacios de participación físicos y virtuales.
8. Predecir el comportamiento de programas utilizando el razonamiento lógico en entornos gráficos y de implementación.
9. Comprender la noción de algoritmo como un método de resolución de problemas en contextos diferentes.
10. Relacionar una representación gráfica con una secuencia ordenada de acciones para resolver problemas de información.
11. Experimentar utilizando la exploración las diferentes soluciones concretas a una situación problemática utilizando programación estructurada.
12. Diseñar, escribir y depurar programas del paradigma imperativo que cumplan objetivos específicos.
13. Analizar los procedimientos y rutinas a los efectos de ser reutilizados, por ellos o por otros, valorizando así el trabajo en equipo y la necesidad de una correcta especificación.
14. Trabajar con variables y diversas formas de entrada y salida, para incorporar el concepto de estructuras de datos persistentes o volátiles.
15. Generalizar patrones estableciendo estructuras que puedan ser reutilizadas en otras aplicaciones.
16. Comparar diferentes algoritmos para la solución de un mismo problema en los términos de refinamiento y optimización.

**Contenidos mínimos**

Programación Imperativa y Concepto de algoritmo.

Concepto de Dato.

Tipos de Datos Simples.

Tipo Abstracto de datos.

Estructuras de Control Básicas.

Estrategias de Resolución de problemas.

Estructuras de Datos.

Abstracciones con procedimientos y funciones.

Estructuras de Datos lineales y no lineales.

Algoritmos de Carga, Búsqueda, Recorrido y Ordenamiento.

Archivos de Acceso Secuencial y Aleatorio.

Recursividad.

Punteros y asignación dinámica en memoria

Estructuras enlazadas: Lineales y arboreas

**Computación – Informática - Sistemas**

**Metáfora: Luthier – Musico – director de orquesta**

Sistemas

Ingeniero en sistemas

UTN

Informática

Ingeniero informático

UBA

Computación

Lic, Cs. Computación

UBA

## ISI 🡪DISI🡪UTN.BA

## Ingenieria en Sistemas de Informacion

* DISI
  + Area ciencias básicas 574 hs.
  + Area física 320 hs.
  + Area Sistemas 960 hs.
    - Gestión 234 hs.
    - Modelos 908 hs.
  + Computación 512 hs.
  + Programación 640 hs

## El área de programacion

* **Logica y estructuras discretas 96 hs. 🡪 Logica y modelos**
* **Algoritmos y estructura de datos 160 hs. 🡪 Resolucion de problemas**
* **Sintaxis y semántica de los lenguajes 128 hs. 🡪 Correcta escritura y significado**
* **Paradigmas de programación 128 hs. 🡪 Distintos enfoques**
* **Gestión de datos 128 hs. 🡪 Volumen de datos**

|  |  |  |
| --- | --- | --- |
| **LOGICA Y EST. DISCRETAS 96** | **ALGORITMOS Y ESTRUCTURA DE DATOS 160** | |
| **SINTAXIS Y S EMANTICA 128** | **PARADIGMAS 128** | **GESTION DE DATOS 128** |

## Algoritmos y estructura de datos

El propósito de algoritmos y estructura de datos es la resolución de problemas RP computables de información por lo que requiere la conjunción de:

1. Datos: debe definir sus tipos
2. Con una secuencia lógica y precisa de acciones: definiendo el algoritmo
3. implementar la solución: mediante en LP, en este caso C++

## Paradigmas de programación

## Matriz Transversal de Contenidos propuesta:

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | Generales | | | Tipos de datos | | | | | | | | | |
| LI | RP | TD | DS | ER | AT | AB | EI | ED | EA | EG | CC | Bib |
| LyED | ALTA | BAJA | ALTA | BAJA | BAJA | BAJA | BAJA | BAJA | BAJA | BAJA | BAJA | NO | NO |
| **AyE** | **ALTA** | **ALTA** | **ALTA** | **ALTA** | **ALTA** | **ALTA** | **ALTA** | **ALTA** | **ALTA** | **MED.** | **BAJA** | **ALTA** | **MED.** |
| SSL | ALTA | MED. | ALTA | BAJA | MED. | MED. | MED. | MED. | BAJA | MED. | ALTA | MED. | ALTA |
| PP | ALTA | MED. | ALTA | BAJA | BAJA | MED. | MED. | MED. | BAJA | MED. | BAJA | BAJA | ALTA |

## Temas a desarrollar

El contenido total del documento abarca:

1. Introducción a la algoritmia
2. Introducción a C, C++
3. Tipos de datos
   1. Datos simples
   2. Estructuras de datos
4. Asignación, análisis de caso repeticiones
5. Patrones algorítmicos simples
6. Struct
7. Array
8. Flujos
9. Estructuras enlazadas
10. Combinaciones de estructuras
11. Patrones de carga, recorrido, búsqueda y ordenamiento
12. Criterios de selección de estructuras de datos y de control

|  |
| --- |
| Introduccion |
| [Introducción a la algoritmia](https://drive.google.com/file/d/1XaFTxPZdnCCSbB8aKlgk0uOP5s8pxaQ7/view?usp=sharing) |
| [Entorno de desarrollo C-C++](https://drive.google.com/file/d/1XWei9Rra_sYrNGs-MgfyuPAM0nAY8Q65/view?usp=sharing) |
| [Tipos de datos](https://drive.google.com/file/d/1cy1iuULUc3Dz3QmC64tSaANyUgbhdkif/view?usp=sharing) |
| Estructuras de control |
| Asignación  Análisis de caso  Ciclos |
| Patrones algorítmicos simples |
| Intercambios |
| Máximos y mínimos: distintos lotes, distintos criterios |
| Seguidillas |
| Ejercicios integradores |
| Práctica |

**Funciones**

|  |
| --- |
| Tema |
| Definiciones y declaraciones |
| Declaración, definición, prototipos, invocación, bibliotecas |
| Intercambio de información |
| Argumentos, parámetros: por valor, por referencia |
| Reusabilidad – Generalidad |
| Concepto de reusabilidad |
| Concepto de generalidad |
| Ejercicios integradores |
| Práctica |

**Struct**

|  |
| --- |
| Tema |
| Definiciones y declaraciones |
| Combinación de estructuras: struct con un campo struct |
| Asignación interna y externa |

**Array**

|  |
| --- |
| Tema |
| Necesidad de su uso. Definiciones y declaraciones |
| Vectores y matices |
| Datos simples, estructuras |
| Combinación de estructuras: array de struct, struct con array |
| Recorridos |
| Secuencial |
| En un rango |
| Con Corte de control (criterio único/criterio múltiple) |
| Apareo (criterio único/criterio múltiple) |
| Búsqueda |
| Directa |
| Secuencial |
| Dicotómica (criterio único/criterio múltiple) |
| Carga |
| Directa |
| Secuencial |
| Ordenada |
| Ordenamiento |
| Con Posición Única Predecible |
| Método de ordenamiento |
| Ejercitación |
| Ejercicios integradores |
| Práctica |

**Contenidos analíticos**

**Parte I: Del Universo real al universo computacional**

**Unidad 1: Definición de términos y frases**

Definición de paradigmas de programación. Profundización del paradigma imperativo, modular y estructurado. Concepto de Computación, Informática, Sistemas. Tipos de problemas: problemas computacionales. Definición de asignación, sentencia, declaración, dato, información, conocimiento, lenguajes de programación. Partes de un programa, Abstracción, Modularización, Especificacion

**Unidad 2: Conceptos Fundamentales de Algoritmia**

Del problema real a su implementación en computadoras. Análisis y comprensión de problemas de información planteados en lenguajes natural. Representaciones graficas para la formalización y resolución estratégica de problemas computables de información. Concepto de algoritmo. Refinamientos sucesivos. Lexico y algoritmo. Características, propiedades y eficiencia de los algoritmos. Estructura de los algoritmos. Proceso computacional.

**Unidad 3: Tipos de datos primitivos y operadores**

Definición de tipo de dato. Tipos de datos primitivos (definición de los conjuntos de valores y operaciones aplicadas a los mismos. Definición y alcance de tipos de datos simples, ordinales y no ordinales, introducción al concepto de cadenas de caracteres. Operadores lógicos, aritméticos de relación de asignación, utilización y combinaciones de los mismos. Patrones algorítmicos con datos simples: secuencias, seguidillas, selecciones.

**Logros Pedagógicos:** Comprender los problemas de información con la utilización de datos de tipo simple y utilizar el concepto de refinamientos sucesivos para la eficaz resolución de un problema computacional Mediante el desarrollo conceptual y/o grafico del algoritmo mas adecuado para la resolución.

**Parte II: Constructos del Lenguaje: Expresiones, Declaraciones, Sentencias y Estructura de Programa**

**Unidad 4 Tipos, Operadores, y Expresiones**

Tipo, precedencia, asociatividad, orden de evaluación. Declaraciones. Definiciones, Alcance. Identificadores, Constantes, Variables, valorL

**Unidad 5 El lenguaje de programación C++:**

Estructura general de un programa. Concepto de bibliotecas, introducción al conceptode la descomposición: prototipos, firmas y desarrollo de funciones. Concepto de espacios de nombres.

Expresiones y sentencias: sentencias simples estructuradas, compuestas. Análisis de caso simples, múltiples, completos, incompletos. Composiciones iterativas: la necesidad de iterar. Composiciones exactas, no exactas, precondicionales, poscondicionales. Iteracion versus recursividad. Composiciones recursivas.

**Unidad 6 Descomposiciones y Funciones**

La necesidad de la descomposición. Declaración y definición de funciones: argumentosy parámetros; invocación y declaración. La propiedad de la Reusabilidad. Ventajas de la generalidad de tipo y de criterio de selección

Logros Pedagógicos: Aplicar las abstracciones de los lenguajes de programación para la implementación de los algoritmos mediante la descomposición a través de funciones.

**Parte III Estructura de datos estáticos y almacenamiento logico**

**Unidad 7 Concepto de tipos de datos derivados y abstractos**

Definición de tipo de dato derivado y estructura de dato. Operadores de acceso a cada miembro de la estructura. Estructuras de datos con almacenamiento lógico: Estructuras de posiciones contiguas de memoria de datos no homogéneos y estructuras de datos homogéneos. Combinaciones. Concepto de estructuras estáticas y estructuras con asignación dinámica en memoria. Concepto de estructuras con almacenamiento físico. Profundización de estudio de estructuras de posiciones contiguas no homogéneas en memoria. Introducción al concepto de abstracción de datos

**Unidad 8 Estructuras indexadas**

Definición y declaración de array. Array de tipo de dato simple. Array de tipos de datos derivados. Tamaño lógico y tamaño físico. Estructura unidimensional o multidimensional. Array de caracteres. Concepto de puntero constante, desplazamiento.

**Unidad 9 Patrones algorítmicos**

1. Patrones de carga: en la definición. Secuencial, directa, ordenada
2. Patrones de recorrido: total, parcial, en ambas direcciones. Con corte de control, apareando dos o mas estructuras
3. Patrones de búsqueda: directa, binaria, lineal. Análisis de eficiencia
4. Patrones de ordenamiento: Con posición Única predecible, mediante métodos de ordenamiento. Análisis de eficiencia

Logros Pedagógicos: Implementar patrones algorítmicos de carga, recorrido y búsqueda en estructuras con posiciones contiguas de memoria, homogéneas, no homogéneas y combinadas

**Parte IV Almacenamiento Físico - Estructuras con asignación dinámica**

**Unidad 10 Estructuras de Almacenamiento físico**

Concepto de almacenamiento fisico. Archivos de texto y archivos binarios. Archivos con acceso directo. Analisis de bibliotecas para el manejo de archivo. Nombre lógico y fisico.

Patrones algoritmicos: comparación con patrones de otras estructuras de manejos de conjunto de datos del mismo tipo, conservando la lógica y modificando el modo de acceso a cada uno de los datos particulares

**Unidad 11 Punteros, asignación dinámica en memoria y estructuras enlazadas lineales**

Definición de punteros. La necesidad de la asignación dinámica en memoria. Constructores y destructores. Concepto de instancia y operadores de dirección y desreferenciacion Punteros a datos y punteros a código. Estructuras enlazadas con asignación dinámica en memoria:

Estructuras lineales, ventajas y debilidades. Definiciones y declaraciones

Estructura de tipo pila: Patrones de carga y eliminación de nodos

Estructura de tipo cola: Patrones de carga y eliminación de nodos.

Estructura de tipo lista: listas simplemente enlazadas, listas circulares, listas doblemente

enlazadas y listas circulares doblemente enlazadas.

Patrones de carga sin restricciones, de carga sin repetir del dato, de búsqueda, de recorrido (completo, parcial, con corte de control o apareo) de eliminación de nodos

**Unidad 12: Estructuras arbóreas**

Definición de árbol, árbol binario de búsqueda, árbol balanceado inserción y recorrido

recursivo e iterativo: análisis de eficiencia.

**Unidad 13 Combinaciones complejas de estructuras**

Resolución de problemas complejos combinando las estructuras analizadas y justificando la elección de las mismas.

Logros Pedagógicos: Aplicar la especificación precisa del algoritmo de resolución de un problema complejo de información mediante la utilización de las estructuras mas idóneas para la solución.

1. [↑](#footnote-ref-1)