

Estructuras de Datos

Profesor

Sergio Gonzalez

Unidad 2: Concepto de Tipo de Dato Abstracto (TDA)

Profesor

Sergio Gonzalez

Y las estructuras de datos????

Redefinición de Programa

=

Estructura de datos + Algoritmo

Programa

=

Estructura de datos + Algoritmo

- Estructura: Forma de organizar y representar los datos (modelo)
- Algoritmo: Pasos para resolución de problemas

Estructura de datos

- La información no aparece aislada en datos simples
- Proceso de abstracción – Identificación de características y modelado

Estructura de datos

- Variables y términos mas parecidos a los usados por los humanos (facilita manejo)
- Agrupamiento de datos básicos con estructura y organización interna
- Se ven como una unidad en conjunto

Una estructura de datos adecuada simplifica el diseño del algoritmo

Clasificación de Estructuras de datos

- Por su naturaleza:
 - Homogénea: Todos datos básicos del mismo tipo (vectores, tablas, matrices, etc.)
 - Heterogénea: Datos de diferentes tipos (registros, etc.)

Clasificación de Estructuras de datos

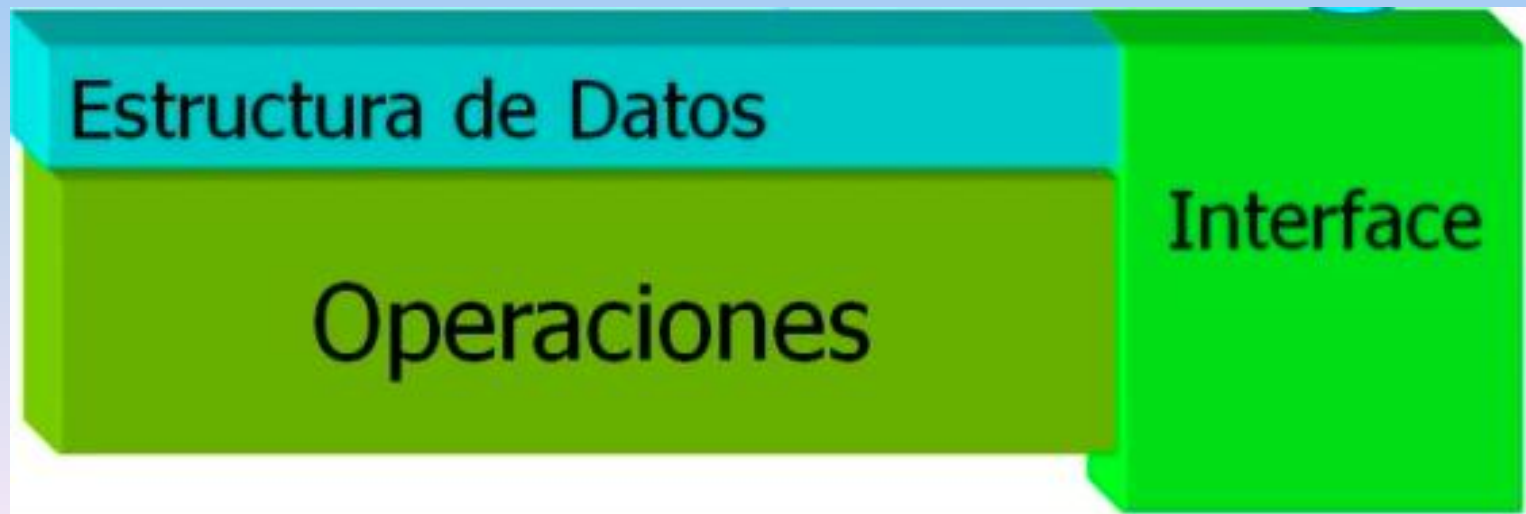
- Por su ubicación en memoria:
 - Contiguas: Componentes ubicados en áreas adyacentes de memoria (Ej. vectores)
 - Enlazadas: No tienen porque estar en zonas contiguas de memoria, uso de direcciones (Ej. listas enlazadas)

Clasificación de Estructuras de datos

- Por su forma de almacenamiento:
 - Estáticas: Tamaño conocido a priori
 - Dinámicas: El tamaño varia durante la ejecución del programa (Ej: listas, arboles, grafos)
 - Externas: Archivos y bases de datos

Tipos de datos abstractos (TDA)

Estructura de datos + Operaciones



Tipos de datos abstractos (TDA)

- Lado del usuario: El uso de un TDA solo depende de su especificación, no de la implementación
- Lado del desarrollador: Dada una especificación hay muchas implementaciones validas

Tipos de datos abstractos (TDA)

- Abstracción (Flexibilidad):
 - Ocultamiento: Usuario sabe **que** puede hacer, pero no **como** lo hace
 - Encapsulamiento: Usuario no puede modificarlo ni acceder a su interior
 - Cambios en la implementación son transparentes al usuario

Tipos de datos abstractos (TDA)

- Especificación:
 - Estructura de datos
 - Operaciones necesarias (interface con usuario)
 - Precisa, legible y no ambigua

Tipos de datos abstractos (TDA)

- Implementación interna:
 - Representación de estructura de datos
 - Operaciones
 - Estructurada, legible y eficiente
 - Transparente para el usuario (Abstracción)

Ejemplo: TDA Empleado

- Especificación:
 - Que información de un empleado necesitamos almacenar? (Estructura)
 - Que operaciones necesitamos sobre el TDA? (Interface)

Ejemplo: TDA Empleado

- Información a almacenar (Estructura):
 - Nombre
 - Apellido
 - Fecha de nacimiento
 - DNI
 - Puesto
 - Lugar de trabajo

Ejemplo: TDA Empleado

- Operaciones (Interface):
 - Crear un empleado nuevo
 - Modificar datos de un empleado existente
 - Borrar un empleado
 - Obtener valores de un empleado (Ej. Nombre)
 -

Ejemplo: TDA Empleado

- Implementación:
 - Que tipos de datos básicos utilizar para almacenar cada valor? (Estructura)
 - Algoritmos para implementar operaciones (Interface)

Ejemplo: TDA Empleado

- Tipos de datos para información a almacenar:
 - Nombre -> Cadena de caracteres
 - Apellido -> Cadena de caracteres
 - Fecha de nacimiento -> Otro TDA???
 - DNI -> Entero
 - Puesto -> Cadena de caracteres
 - Lugar de trabajo -> Cadena de caracteres

Ejemplo: TDA Empleado

- Programar operaciones necesarias:
 - Crear un empleado nuevo -> crearEmpleado()
 - Modificar datos de un empleado existente -> modificarEmpleado()
 - Borrar un empleado -> eliminarEmpleado()
 - Obtener valores de un empleado (Ej. Nombre) -> obtenerNombre()
 -

Como implementamos estructuras
de datos en Python?????