



USAC
TRICENTENARIA
Universidad de San Carlos de Guatemala

Estructuras de datos

Listas y Generics

Facultad de Ingeniería
Escuela de Ciencias y Sistemas
Estructuras de Datos
Ing. Edgar René Ornélyz
Tutor Esvin González

**¿Qué me hace ser un buen
estudiante?**

¿Qué es una estructura de datos?



Estructuras de datos

¿Qué es una estructura de datos?

Recibe este nombre todo aquel conjunto de datos relacionados entre sí que existen en la memoria primaria de un computador, su persistencia está ligada a un ámbito y a la ejecución de determinado programa.

**¿Qué estructuras de datos
existen?**

—



Estructuras de datos más comunes

Existen múltiples tipos de estructuras de datos, entre las más comunes están:

- Listas
- Iteradores
- Arreglos
- Matrices
- Tablas de dispersión
- Conjuntos
- Textos
- Árboles
- Grafos



Listas enlazadas

Son estructuras de datos unidimensionales cuyos elementos son insertados de manera lineal.

- Listas simples
- Listas dobles
- Listas circulares
- Skip lists
- Listas con políticas de acceso
 - Pila
 - Cola
 - Montículo

Lista simple

Los elementos se insertan de manera lineal. Su contenido se recorre de elemento en elemento en un único sentido, sin posibilidad de establecer una jerarquía entre sus nodos.



Eficiencia

- Los tiempos de búsqueda y eliminación son de orden n
- El tiempo de inserción depende de la implementación de la lista
- El tiempo de ordenamiento depende del algoritmo a implementar



¿En qué casos?

- No se necesita una estructura jerárquica
 - No se necesitan realizar búsquedas o tener datos ordenados
 - No se realizarán eliminaciones o modificaciones de manera recurrente
 - Los recorridos son lineales y en uno (o a lo sumo dos) sentidos
 - Los accesos son siempre sobre la totalidad de elementos, no sobre un elemento en específico
-

Implementación de una lista simple en cinco minutos

Lista doble

Los elementos se insertan de manera lineal. Su contenido se recorre de elemento en elemento, pero con la diferencia de que el mismo puede realizarse en dos sentidos, especialmente útil para “navegar” de mejor manera entre los elementos que contiene.



Generics

En Java y otros lenguajes de programación como C# se pueden declarar tipos dinámicos, esto permite diseñar clases y métodos que aplazan la especificación de uno o más tipos hasta que el código de cliente declara y crea una instancia de la clase o del método.



Comodín Tarea 1

Conceptos clave necesarios para ser un mejor programador.

- Instancia
- Referencia
- Static
- Generic
- Interface
- Modificadores de acceso

Tarea 1

\(\cup \cup \) /

—



Tarea 1: Modificar la lista simple de '002 Lists'

A) Hacer que la lista simple (**MySimpleList**) ahora sea **circular** y agregar una **bandera** para que al llamar al método **tour** recorra la lista **dos veces** utilizando el **enlace que la hará circular**. En el método **main** insertar (usando el método **addAtEnd**) las palabras:

Abeja Elefante Iguana Oso Uva

Al finalizar de insertar los datos llamar al método **tour()**. Tomar un pantallazo (no una foto) de la salida del programa y adjuntar el código del método **main** y del método **tour**.



Tarea 1: Modificar la lista doble de '002 Lists'

B) Hacer que la lista doble (**MyDoubleList**) ahora funcione con una clase genérica llamada **Pet** que tenga como atributos:

String : especie

String : name

int : age

String : toString() => "Mi " especie " se llama " name " y tiene " age " años"

En el método main insertar 3 mascotas y recorrer la lista usando:
getCurrent() -> getPrev() -> getNext() -> getNext() -> getNext()



Tarea 1: Entrega

Hacer un pdf que tenga un encabezado sencillo y el contenido de su tarea:

Nombre: Fulano de Tal (obviamente aquí va su nombre)

Carnet: y aquí su carnet

TAREA 1 (A)

El pantallazo del inciso (A)

El código de los método main y tour

TAREA 1 (B)

El pantallazo de la salida del inciso (B)

El archivo debe llamarse **T1_<su_carnet>.pdf** y deben subirlo a <https://goo.gl/4S3pSX> antes de la medianoche del domingo 30 de julio.

Resumen

- La memoria dinámica es el fundamento de las estructuras de datos
- Las listas simples son útiles para hacer recorridos lineales
- Las listas dobles son útiles para “navegar en ambos sentidos”
- Las listas son buenas para insertar elementos, más no para buscar, modificar o eliminar.



Gracias por su atención



Referencias

- Genéricos (Guía de programación de C#) | Microsoft Docs
 - Disponible en: <https://goo.gl/WiPGZu>
- Repositorio en github del contenido del laboratorio
 - Disponible en: <https://github.com/Esvux/LabEDD>
- Material adicional
 - Listas enlazadas (1ero de 3 tutoriales) <https://youtu.be/9q758AJ1nck>