



**EDUCACIÓN**  
SECRETARÍA DE EDUCACIÓN PÚBLICA



TECNOLÓGICO  
NACIONAL DE MÉXICO®



# TECNOLÓGICO NACIONAL DE MÉXICO INSTITUTO TECNOLÓGICO DE TIJUANA

## ESTRUCTURA DE DATOS

### UNIDAD 3

## List<T> en C#

*C. Abraham Jhared Flores Azcona*  
*19211640*

Profesora:  
M.C. Claudia Negrete Sanchez

26 de octubre del 2020

# Índice

<b>1</b>	<b>Introducción</b>	<b>1</b>
<b>2</b>	<b>Concepto</b>	<b>1</b>
2.1	Lista enlazada (concepto general) . . . . .	1
2.2	Lista enlazada (documentación en C#) . . . . .	1
<b>3</b>	<b>Métodos usados</b>	<b>1</b>
3.1	Constructores . . . . .	1
3.2	Propiedades . . . . .	2
3.3	Métodos . . . . .	2
<b>4</b>	<b>Conclusión</b>	<b>2</b>
	<b>Referencias</b>	<b>3</b>

# Índice de figuras

1	Modelo de las listas enlazadas. . . . .	1
---	---	---

## 1. Introducción

En esta investigación breve se explica el concepto, métodos y demás detalles relevante a la clase List<T> la cual simplifica el desarrollo de programas que emplean las listas enlazadas.

## 2. Concepto

### 2.1. Lista enlazada (concepto general)

Es una estructura de datos recursiva la cual o está vacía o con un nodo teniendo un elemento genérico y una referencia a la lista enlazada. También se considera como una estructura lineal en la cual sus elementos no están almacenados en lugares contiguos de memoria. Estos usan punteros.

### 2.2. Lista enlazada (documentación en C#)

Representa una lista fuertemente tipificada que pueden ser accedidos por un índice. Esta provee métodos para la búsqueda, ordenamiento, y manipulación de listas.

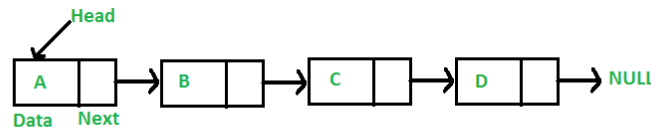


Figura 1: Modelo de las listas enlazadas.

## 3. Métodos usados

Como cualquier clase, estos contienen distintos métodos, propiedades y constructores, de los cuales los mas relevantes son los siguientes:

### 3.1. Constructores

- `List<T>()`: Inicializa una nueva instancia de la clase que está vacía y tiene la capacidad inicial por defecto.
- `List<T>(IEnumerable<T>)`: Inicializa una nueva instancia de la clase que contiene elementos copiados de una colección especificada y ue tiene la capacidad suficiente para acomodar el número de elementos copiados.
- `List<T>(Int32)`: Inicializa una nueva instancia de la clase la cual está vacía y tiene la capacidad inicial especificada.

### 3.2. Propiedades

- *Capacity*: obtiene o establece el numero total de elementos que la estructura interna puede mantener sin un cambio de tamaño.
- *Count*: obtiene el numero de elementos contenidos en *List<T>*.
- *Item[Int32]*: obtiene o establece el elemento del índice especificado.

### 3.3. Métodos

- *Add(T)*: añade un objeto al final de la lista.
- *Clear()*: remueve todos los elementos de la lista.
- *Contains(T)*: determina si un elemento dado está dentro de la lista.
- *IndexOf(T, Int32)*: busca el índice de la primer ocurrencia entre el rango de elementos desde dicho indice hasta el final de la lista.
- *Remove(T)*: remueve la primer ocurrencia del objeto especificado en la lista.

## 4. Conclusión

El uso de esta clase permite el desarrollo de programas más sencillos de desarrollar y por ende mas sencillos de comprender para el aprendizaje de los mismos.

---

## Referencias

- [1] *Linked List Data Structure - GeeksforGeeks*. URL: <https://www.geeksforgeeks.org/data-structures/linked-list/>.
- [2] *List<T> Class (System.Collections.Generic)*. URL: <https://docs.microsoft.com/en-us/dotnet/api/system.collections.generic.list-1?view=netcore-3.1>.