

INSTITUTO TECNOLÓGICO DE CHIHUAHUA II

Autor: José Sebastian López Ibarra

Docente: Rubén Alonso Hernández Chávez

07 / 10 / 2022

EDD – 09 – ISC

Índice

[Introducción 1](#_Toc115979122)

[Objetivos 2](#_Toc115979123)

[Desarrollo 3](#_Toc115979124)

[Collections 3](#_Toc115979125)

[Tipos de colecciones 3](#_Toc115979126)

[Jerarquía de las Colecciones 4](#_Toc115979127)

[Colecciones en los diferentes lenguajes de programación 5](#_Toc115979128)

[Python 5](#_Toc115979129)

[C++ 5](#_Toc115979130)

[C# 6](#_Toc115979131)

[JavaScript 6](#_Toc115979132)

[Conclusiones 7](#_Toc115979133)

[Recomendaciones 8](#_Toc115979134)

[Referencias 9](#_Toc115979135)

# Introducción

La manera en que se organiza una computadora puede parecer complicado, y aunque si lo es, una vez que entiendes su funcionamiento descubres que puede ser más sencillo de lo que parece.

Esto mismo sucede con la forma de organizar los datos dentro del mismo computador. Podríamos decir que la memoria se maneja y organiza en bloques, y a su vez, se divide en dos unidades fundamentales: el Stack y el Heap.

Existen diversas formas de guardar esa información, por ejemplo: arreglos, listas, etc. En algunos lenguajes de programación, existe una clase que abarca todas estas estructuras, llamadas collections.

# Objetivos

* Tener un conocimiento sobre las estructuras que guardan información en los lenguajes de programación.
* Ampliar nuestro conocimiento sobre la carrera con los conocimientos previos a esta.

# Desarrollo

## Collections

El marco de las colecciones es el conjunto de clases e interfaces las cuales llevan a cabo estructuras que guardan datos. Proporciona interfaces, las cuales definen las colecciones y clases que se van a utilizar.

La mayoría de las colecciones en Java derivan de la interfaz java.util.Collection. Esta interfaz define las partes básicas de todas las colecciones. Dentro de la misma interfaz podemos encontrar diferentes funciones/métodos para manipular las colecciones. De igual forma. La interfaz Collection es genérica, es decir, cualquier colección se puede escribir para almacenar cualquier clase, guardar sólo valores String, int, etc.

### Tipos de colecciones

Existen tres tipos de colecciones que contienen la clase Collection, las cuáles son:

#### Listas ordenadas.

Este tipo de colecciones permiten al programador insertar elementos teniendo un cierto orden, recuperando esos elementos en el mismo orden.

Las listas son más flexibles en el manejo de estas, en comparación con los arreglos. Como mencioné anteriormente, los elementos guardan un orden y se pueden colocar en una posición concreta.

Un ejemplo de listas ordenadas puede ser una lista de espera para comprar un boleto para un concierto. Una vez que se terminó de atender al primer cliente, continua con el siguiente, mientras más personas se van integrando a la lista.

#### Diccionarios/mapas

Al conjunto de referencias a objetos con una clave de búsqueda para acceder a los valores del objeto se le conoce como diccionarios o mapas.

Un ejemplo de una llave es una tarjeta de identificación.

#### Conjuntos

Se conoce como conjunto a las colecciones desordenadas que se pueden iterar y contienen cada elemento como máximo una vez. (Wikipedia contributors, 2021)

## Jerarquía de las Colecciones

## Colecciones en los diferentes lenguajes de programación

Las colecciones son estructuras que guardan datos, por lo que son útiles en muchos casos.

Java no es el único lenguaje de programación que implementa este tipo de estructuras. Existen distintos lenguajes que también las implementan, como lo son:

### Python

Al igual que en java, Python implementa:

* Listas. Colección ordenada y modificable
* Conjuntos. Colección que no tiene un orden.
* Diccionarios. Misma naturaleza que una lista.

Python, sin embargo, implementa otro tipo de colección de datos, la cual es la “Tupla”.

La tupla, al igual que una lista, es una colección de datos ordenada, con la única diferencia que esta es inmutable, no cambia. Un ejemplo donde se pueden implementar las tuplas es a la hora de pedir nombres de personas.

Python 3 tiene varias estructuras de datos integradas, incluidas tuplas, diccionarios y listas. Las estructuras de datos nos brindan una forma de organizar y almacenar datos. El módulo de colecciones nos ayuda a poblar y manipular estructuras de datos de manera eficiente. (Colecciones en Python, s. f.)

### C++

Windows::Foundation::Collections::IObservableVector y Windows::Foundation::Collections::IObservableMap definen eventos que se desencadenan cuando la colección cambia de distintas maneras. Al implementar estas interfaces, Platform::Collections::Map y Platform::Collections::Vector permiten enlaces de datos con colecciones de XAML.

Las colecciones se dividen en cuatro categorías: versiones modificables y versiones de solo lectura de colecciones de secuencia y asociativas.

Los elementos que se encuentran dentro de una colección modificable pueden cambiar, sin embargo, los elementos de una colección de solo lectura, que se denomina vista, solo se pueden leer. Se puede acceder a los elementos de Platform::Collections::Vector o Platform::Collections::VectorView con un iterador o Vector::GetAt de la colección, y un índice. (Colecciones (C++/CX), 2022)

### C#

Las clases del espacio de nombres System.Collections no almacenan los elementos como objetos de tipo específico, sino como objetos del tipo Object.

Las colecciones utilizadas en este lenguaje de programación son:

* ArrayList. Lista que guarda datos y que aumenta dinámicamente cuando se requiere.
* HashTable.
* Queue. Colección de objetos con metodología FIFO (First In, First Out). Primero que entra, primero que sale.
* Stack. Colección de objetos con metodología LIFO (Last In, First Out). Último que entra, primero que sale. (Colecciones (C#), 2022)

### JavaScript

Estas estructuras de datos, introducidas en ECMAScript Edition 6, tienen como referencias a objetos como claves. Set y WeakSet representan un conjunto de objetos, mientras que Map y WeakMap se asocian un valor a un objeto. Éstos últimos serían similares a los diccionarios en Java.

Las implementaciones nativas de ellos (incluidos los WeakMaps) pueden tener un rendimiento de búsqueda que es aproximadamente logarítmico al tiempo constante.

Por lo general, para vincular datos a un nodo DOM, se pueden establecer propiedades directamente en el objeto o usar atributos data-\*. Tiene como desventaja que los datos están disponibles para cualquier script que se ejecute en el mismo contexto. Los Maps y WeakMaps facilitan la vinculación privada de datos a un objeto. (Tipos de datos y estructuras en JavaScript - JavaScript | MDN, 2022)

# Conclusiones

La manera en que los tipos de datos se almacenan hoy en día han sido de gran utilidad en la programación, nos permiten contener información de una manera mucha más práctica

Implementar métodos que faciliten un proceso es lo que un programador busca, facilitar la manera en que guarda información.

# Recomendaciones

* En el Instituto existen algunas áreas para descansar, sin embargo, son muy pocas las que cuentan con un espacio de sombra. Siento que, al implementar nuevas zonas, el alumnado disfrutaría el poder ir a su institución sabiendo que tendrá un lugar donde reposar.
* El nivel de los docentes es bueno, de hecho, he aprendido y me he llenado de mucho conocimiento. Sin embargo, muchos profesores tienen una manera de enseñar que no es adecuada para el aprendizaje, no porque no enseñen, más bien es la forma en que lo hacen, confunde mucho a los estudiantes.

# Referencias

* Wikipedia contributors. (2021, 31 octubre). *Java collections framework*. Wikipedia. Recuperado 5 de octubre de 2022, de <https://en.wikipedia.org/wiki/Java_collections_framework>
* *Colecciones en Python*. (s. f.). Linuxteaching. Recuperado 6 de octubre de 2022, de <https://www.linuxteaching.com/article/collections_in_python>
* *Colecciones (C++/CX)*. (2022, 26 septiembre). Microsoft Learn. Recuperado 6 de octubre de 2022, de <https://learn.microsoft.com/es-es/cpp/cppcx/collections-c-cx?view=msvc-170>
* *Colecciones (C#)*. (2022, 5 octubre). Microsoft Learn. Recuperado 6 de octubre de 2022, de <https://learn.microsoft.com/es-es/dotnet/csharp/programming-guide/concepts/collections>
* *Tipos de datos y estructuras en JavaScript - JavaScript | MDN*. (2022, 4 octubre). Recuperado 6 de octubre de 2022, de <https://developer.mozilla.org/es/docs/Web/JavaScript/Data_structures>