**Módulo - 15] Colecciones:**

**(083) -** ¿Cuál es la diferencia entre las **colecciones** y las **matrices**?

Las colecciones proporcionan una manera más flexible de trabajar con grupos de objetos. A diferencia de las matrices, el grupo de objetos con el que trabaja puede aumentar y reducirse de manera dinámica a medida que cambian las necesidades de la aplicación. Para algunas colecciones, puede asignar una clave a cualquier objeto que incluya en la colección para, de este modo, recuperar rápidamente el objeto con la clave.

**(084) -** ¿Cuál es la diferencia entre las **colecciones genéricas** y las **no genéricas**?

Si la colección contiene elementos de un solo tipo de datos, puede usar una de las clases del espacio de nombres [System.Collections.Generic](https://docs.microsoft.com/es-es/dotnet/api/system.collections.generic). Una colección genérica cumple la seguridad de tipos para que ningún otro tipo de datos se pueda agregar a ella.  Por ej  [List<T>](https://docs.microsoft.com/es-es/dotnet/api/system.collections.generic.list-1).

**(085) -** ¿Es necesario determinar el tipo de dato o realizar una conversión al recuperar un objeto de una colección genérica?

Cuando recupera un elemento de una colección genérica, no tiene que determinar su tipo de datos ni convertirlo.

**(086) -** Describa los siguientes tipos de colecciones genéricas: **Dictionary**, **List**, **SortedList**.

[List<T>](https://docs.microsoft.com/es-es/dotnet/api/system.collections.generic.list-1)) Representa una lista de objetos a los que puede tener acceso el índice. Proporciona métodos para buscar, ordenar y modificar listas.

[Dictionary<TKey,TValue>](https://docs.microsoft.com/es-es/dotnet/api/system.collections.generic.dictionary-2)) Representa una colección de pares de clave y valor que se organizan según la clave.

[Queue<T>](https://docs.microsoft.com/es-es/dotnet/api/system.collections.generic.queue-1)-) Representa una colección de objetos de primeras entradas, primeras salidas (FIFO).

[SortedList<TKey,TValue>](https://docs.microsoft.com/es-es/dotnet/api/system.collections.generic.sortedlist-2)) Representa una colección de pares clave-valor que se ordenan por claves según la implementación de [IComparer<T>](https://docs.microsoft.com/es-es/dotnet/api/system.collections.generic.icomparer-1) asociada.

[Stack<T>](https://docs.microsoft.com/es-es/dotnet/api/system.collections.generic.stack-1)) Representa una colección de objetos de últimas entradas, primeras salidas (LIFO).

**(087) -** ¿Qué son y cuál es la diferencia entre una **cola (queue)** y una **pila (stack)**? Asocie con los conceptos “**FIFO**” y “**LIFO**”.

Cola->Primero que entra primero que sale

Pila->Ultimo que entra primero que sale

**(088) -** Describa los siguientes tipos de colecciones no genéricas: **ArrayList**, **Hashtable**.

[ArrayList](https://docs.microsoft.com/es-es/dotnet/api/system.collections.arraylist) -> Representa una matriz cuyo tamaño aumenta dinámicamente cuando es necesario.

[Hashtable](https://docs.microsoft.com/es-es/dotnet/api/system.collections.hashtable)-> Representa una matriz cuyo tamaño aumenta dinámicamente cuando es necesario.

**(089) -** ¿Se pueden ordenar directamente las colas y las pilas? ¿Por qué (piense en la función de dichas colecciones)? ¿Cuál es la alternativa?

Las colas, las pilas, y los diccionarios , por si solos no se ordenan , porque perdería el sentido del uso del orden en que se agregan.

La alternativa es Usar ToList.//tambien hay una biblioteca que se llama link.

**(090) -** ¿Cuál es la diferencia entre las colas y pilas genéricas y las colas y pilas no genéricas?

En las NO genéricas podes meter de todo. En las genéricas esta tipado.

**(091) -** ¿Qué muestra el siguiente código?



a. 1, 2, 3, 4, 5,

b. 5, 4, 3, 2, 1,

c. 8, 6, 4, 2, 0, ESTA , CREO

d. 0, 2, 4, 6, 8,

e. 1, 3, 5, 7, 9,

f. 9, 7, 5, 3, 1,

g. Error en tiempo de ejecución. Marcar error.

h. Error en tiempo de diseño. Marcar error.

**[Módulo - 16] Propiedades:**

**(092) -** ¿Qué es y para qué sirve una **propiedad**?

Una propiedad es un miembro que proporciona un mecanismo flexible para leer, escribir o calcular el valor de un campo.

•Las propiedades se pueden usar como si fueran miembros de datos públicos, pero en realidad son métodos especiales denominados descriptores de acceso.

•Las propiedades permiten que una clase exponga una manera pública de obtener y establecer valores, a la vez que se oculta el código de implementación o validación.

**(093) -** ¿Para qué sirve el **descriptor de acceso “get”**?

Para devolver el valor de la propiedad se usa un descriptor de acceso de propiedad get

**(094) -** ¿Para qué sirve el **descriptor de acceso “set”**? ¿Cuál es el papel de la **palabra clave “value”**?

Para asignar un nuevo valor se emplea un descriptor de acceso de propiedad set.La palabra clave value se usa para definir el valor que va a asignar el descriptor de acceso set.

**(095) -** ¿Cómo declaro una propiedad de **sólo lectura**?

Solo usando el get,

**(096) -** ¿Cómo declaro una propiedad de **sólo escritura**?

Solo usando el set.

**[Módulo - 17] Enumerados:**

**(097) -** ¿Qué es un **enumerado**? ¿Cuál es su función?

Son un conjunto propio de constantes con nombre.

Estos tipos de datos permiten declarar un conjunto de nombres u otros valores literales que definen todos los valores posibles que se pueden asignar a una variable.

•Por dentro, estas constantes están asociadas con el tipo de dato int.

**(098) -** ¿Un enumerado sólo puede estar anidado dentro de una clase?

Normalmente es mejor definir un enum directamente dentro de un espacio de nombres para que todas las clases del espacio de nombres puedan acceder a él con igual comodidad. Sin embargo, un enum también se puede anidar dentro de una clase o struct.

**(099) -** ¿Cuál es el primer valor numérico de un enumerado por defecto? ¿Se pueden sobrescribir los valores por defecto?

El valor por defecto del primer elemento es 0,

A un enumerado se le podrá asignar un valor entero a cualquiera de sus partes, cambiando la numeración por defecto.

**(100) -** Indique los valores asociados a cada constante:

enum Day {Sat, Sun, Mon=15, Tue, Wed, Thu=2, Fri};

0,1,15,16,17,2,3.