

## Clase 1 - Especificaciones

---

Para cada especificación de TDA, codificar la interface.

### Ejercicio 1

Definición del TDA Pila de valores enteros

Pila es una estructura que permite almacenar una colección de valores enteros, con la particularidad de que el último elemento en entrar es el primero en salir. También se la conoce como una estructura LIFO.

**Operaciones básicas:**

**InicializarPila:** inicializa la estructura pila.

**Apilar:** agrega un elemento. La estructura debe estar inicializada.

**Desapilar:** elimina el último elemento agregado. La estructura no debe estar vacía.

**Topo:** obtiene el primer elemento a eliminar. La estructura no debe estar vacía.

**PilaVacía:** indica si la pila contiene elementos o no. La estructura debe estar inicializada

### Ejercicio 2

Si los valores a almacenar en el TDA pila en lugar de ser un tipo de dato entero es char. ¿Qué se debe tener en cuenta para generar un nuevo TDA PilaChar?

### Ejercicio 3

Definición del TDA Cola

**Cola** es una estructura que permite almacenar una colección de valores enteros, con la particularidad de que el elemento que el primer elemento en ingresar es el primer elemento en salir. También se la conoce como una estructura FIFO.

**Operaciones básicas:**

**InicializarCola:** inicializa la estructura cola.

**Acolar:** agrega un elemento. La estructura debe estar inicializada.

**Desacolar:** elimina el elemento más antiguo agregado. La estructura no debe estar vacía.

**Primero:** obtiene el primer elemento a eliminar. La estructura no debe estar vacía.

**ColaVacía:** indica si la cola contiene elementos o no. La estructura debe estar inicializada.

### Ejercicio 4

Definición del TDA Cola con prioridad

**Cola con prioridad** es una estructura que permite almacenar una colección de valores enteros, con la particularidad de que el elemento que el primer elemento en ingresar es el primer elemento en salir.

**Operaciones básicas:**

**InicializarCola:** inicializa la estructura cola.

**AcolarPrioridad:** agrega un elemento x con prioridad p suministrados. La estructura debe estar inicializada.

**Desacolar:** elimina el elemento con mayor prioridad. La estructura no debe estar vacía.

**Primero:** obtiene el valor del dato de mayor prioridad. La estructura no debe estar vacía.

**Prioridad:** obtiene la prioridad del valor del dato de mayor prioridad. La estructura no debe estar vacía.

**ColaVacía:** indica si la cola contiene elementos o no. La estructura debe estar inicializada.

## Ejercicio 5

Definición del TDA Conjunto de valores enteros

**Conjunto** es una estructura que permite almacenar una colección de valores enteros, no repetidos y no necesariamente ordenado.

**Operaciones básicas:**

**InicializarConjunto:** inicializa el conjunto.

**Agregar:** agrega un elemento x suministrado. El conjunto debe estar inicializado y el elemento no debe existir.

**Sacar:** elimina un elemento suministrado x. EL elemento debe pertenecer al conjunto.

**Obtener:** devuelve un valor cualquiera del conjunto. El conjunto no debe estar vacío.

**ConjuntoVacío:** devuelve verdadero si el conjunto tiene elementos. El conjunto debe estar inicializado.

**Pertenece:** devuelve verdadero si el valor x recibido como parámetro pertenece al conjunto. El conjunto debe estar inicializado.

## Ejercicio 6

Definición del TDA Diccionario simple, las claves y valores almacenados son enteros

Diccionario simple es una colección de pares asociados (clave, valor), las claves son únicas. No puede existir clave sin valor asociado.

**Operaciones básicas:**

**InicializarDiccionarioSimple:** inicializa el diccionario.

**Agregar:** agrega un elemento x a una clave suministrados. El diccionario debe estar inicializado y no debe existir.

**Eliminar:** elimina una clave suministrada. La clave debe existir.

**Obtener:** devuelve el valor asociada a una clave suministrada. La clave debe existir.

**Claves:** devuelve el conjunto de claves del diccionario. El diccionario debe estar inicializado.

## Ejercicio 7

Definición del TDA Diccionario múltiple, las claves y valores almacenados son enteros

Diccionario múltiple es una colección de pares asociados (clave, valores), las claves son únicas. No puede existir clave sin valor asociado. Los valores son únicos para la misma clave.

**Operaciones básicas:**

**InicializarDiccionarioMultiple:** inicializa el diccionario.

**Agregar:** agrega un elemento x a una clave, ambos datos son suministrados. El diccionario debe estar inicializado y el valor no debe existir.

**Eliminar:** elimina una clave suministrada. La clave debe existir.

**EliminarValor:** elimina un valor dado x asociado a una clave, ambos datos son suministrados. Tanto la clave y como el valor deben existir.

**Obtener:** devuelve el conjunto de valores asociada a una clave suministrada. La clave debe existir.

**Claves:** devuelve el conjunto de claves del diccionario. El diccionario debe estar inicializado.

## Ejercicio 8

Definición del TDA Árbol binario de búsqueda (ABB). Los valores almacenados son enteros

**Árbol de búsqueda binaria** es una colección de elementos ordenados en forma jerárquica. Entre dos elementos se define dos roles: padre e hijo. Cada elemento tiene sólo un padre, excepto el elemento raíz del árbol que no tiene ningún padre.

**Operaciones básicas:**

**InicializarABB:** inicializa el árbol.

**Raíz:** devuelve el valor de la raíz. El árbol no puede estar vacío.

**HijoIzq:** devuelve la referencia al subárbol izquierdo. El árbol no puede estar vacío.

**HijoDer:** devuelve la referencia al subárbol derecho. El árbol no puede estar vacío.

**Agregar:** agrega un valor x suministrado. El árbol debe estar inicializado y no debe existir el valor.

**Eliminar:** elimina un valor suministrado. El valor debe existir.

**ÁrbolVacio:** indica si el árbol contiene elementos o no. La estructura debe estar inicializada.

**Pertenece:** devuelve verdadero si el valor pertenece al árbol. EL árbol debe estar inicializado.

## Ejercicio 9

Definición del TDA Grafo, donde los vértices y el peso de la arista son valores enteros

**Grafo** es una colección de vértices y aristas tipo dirigido, sin aristas paralelas ni bucles, donde los vértices los pesos son enteros.

### Operaciones básicas:

**InicializarGrafo:** inicializa el grafo.

**AgregarVertice:** agrega un vértice. El grafo debe estar inicializado y el vértice no debe existir.

**EliminarVertice:** elimina un vértice suministrada. El vértice debe existir.

**AgregarArista:** agrega una arista suministrando vértices de origen y destino y el peso. Los vértices deben existir, pero no debe existir la arista.

**EliminarArista:** elimina la arista, donde se suministró los vértices de origen y destino. La arista debe existir.

**Vertices** devuelve el conjunto de vértices de. El diccionario debe estar inicializado

**Peso:** devuelve el peso de la arista, identificada por los vértices de origen y destino. La arista debe existir

**ExisteArista:** devuelve verdadero si existe la arista, identificada por los vértices de origen y destino. El grafo debe estar inicializado.