**TAD**

**Integrantes:**

* Cristian Perafan - A00378035
* Danna Espinosa - A00378613
* Samuel Viviescas- A00351949

| TAD Stacks |
| --- |
| Stacks = {E element} |
| { inv:〈E != int,char,double…)〉} (Los stacks no deben contener tipos primitivos.) |
| Operaciones Primitivas:   * Stack: N/A → stack * empty: N/A → boolean * peek: N/A → E * pop: N/A → E * push: E ítem → E * search Object object → int. |
|  |

Stack: Crea una pila vacía.

empty: Comprueba si esta pila está vacía.

peek: Mira el objeto en la parte superior de esta pila sin quitarlo de la pila.

pop: Quita el objeto en la parte superior de esta pila y devuelve ese objeto como el valor de esta función.

push: Empuja un elemento en la parte superior de esta pila.

search: devuelve la posición basada en donde se encuentra un objeto en esa pila.

| TAD Queue |
| --- |
| Queue= {E element} |
| { inv:〈E != int,char,double…〉} (Las Queue no deben contener tipos primitivos.) |
| Operaciones Primitivas:   * add: E item → boolean * element: N/A → E * offer: E item → boolean * peek: E ítem → E * poll: N/A → E * remove: N/A →E |

add: Inserta el elemento especificado en esta cola si es posible hacerlo inmediatamente sin infringir las restricciones de capacidad, regresando al tener éxito y lanzando un espacio si no hay espacio disponible actualmente

element: Recupera, pero no quita, el jefe de esta cola.

offer: Inserta el elemento especificado en esta cola si es posible hacerlo inmediatamente sin infringir las restricciones de capacidad.

peek: Recupera, pero no quita, el encabezado de esta cola o devuelve si esta cola está vacía:null.

poll: Recupera y quita el encabezado de esta cola, o devuelve si esta cola está vacía:null.

remove: Recupera y quita el encabezado de esta cola.

| TAD Hashtable |
| --- |
| Hashtable= {Key<E> key, V<E> value} |
| { inv:〈K<E> && V<E> != int,char,double…)〉} (Los key y value no deben ser tipos primitivos.) |
| Operaciones Primitivas:   * Hashtable N/A —> * clear: N/A → void * compute: K key, BiFunction <? super K , ? super V, ? extends V> remappingFunction → default V * computeIfAbsent: K key, Function <? super K, ? extends V> mappingFunction →default V * containsKey: E item → boolean * containsValue: E ítem → E * elements: N/A → E * entrySet: N/A →E * equals: Object o → boolean * forEach: BiConsumer <? super K, ? super V> action → default void * get: Object key → V * getOrDefault: Object K, V defaultValue → default V * hashCode: N/A →int * isEmpty: N/A → boolean * keySet: N/A → Set<K> * put: K key, V value → V * putAll: Map <? extends K,? extends V> m → void * putIfAbsent: K key, V value → default V * remove: Object key → V * remove: Object key, Object V → default boolean * replace: K key, V value → default V * replace: K key, V oldValue, V new Value → default boolean. * replaceAll: BiFunction <? super K, ? super V, ? extends V> function → default void. * size: N/A → int * values: N/A → Collection <V> |

hashtable:Construye un nuevo hashtable vacío con una capacidad inicial predeterminada (11) y un factor de carga (0,75).

clear: Borra esta tabla hash para que no contenga claves.

compute: Intenta calcular una asignación para la clave especificada y su valor asignado actual (o si no hay una asignación actual null.

computeIfAbsent: Si la clave especificada aún no está asociada a un valor (o está asignada a ), intenta calcular su valor mediante la función de asignación dada y la introduce en este mapa a menos que null.

containsKey: Comprueba si el objeto especificado es una clave en esta tabla hash.

containsValue: Devuelve true si esta tabla hash asigna una o varias claves a este valor.

elements: Devuelve una enumeración de los valores de esta tabla hash.

entrySet: Devuelve una vista Set de las asiganciones contenidas en este mapa.

equals: Compara el objeto especificado con este mapa para obtener igualdad, según la definición de la interfaz Mapa.

forEach: Realiza la acción dada para cada entrada de este mapa hasta que se hayan procesado todas las entradas o la acción arroje una excepción.

get: Devuelve el valor al que se asigna la clave especificada o si este mapa no contiene ninguna asignación para la clave devuelve null.

getOrDefault:Devuelve el valor al que se asigna la clave especificada o si este mapa no contiene ninguna asignación para la clave.defaultValue

hashCode:Devuelve el valor de código hash para este mapa según la definición de la interfaz Mapa.

isEmpty: Compueba si esta tabla hash no asigna claves a los valores.

keys:Devuelve una enumeración de las claves de esta tabla hash.

keySet:Devuelve una vista [**Set**](https://docs.oracle.com/javase/8/docs/api/java/util/Set.html) de las claves contenidas en este mapa.

put: Asigna lo especificado a lo especificado en esta tabla hash.

putAll: Copia todas las asignaciones del mapa especificado a esta tabla hash.

putIfAbsent: Si la clave especificada aún no está asociada a un valor (o está asignada a ) la asocia con el valor dado y devuelve , de lo contrario devuelve el valor actual: null.

rehash: Aumenta la capacidad y reorganiza internamente este hashtable, con el fin de acomodar y acceder a sus entradas de manera más eficiente.

remove:Quita la clave (y su valor correspondiente) de esta tabla hash.

remove:Quita la entrada de la clave especificada sólo si está asignada actualmente al valor especificado.

replace:Reemplaza la entrada de la clave especificada solo si actualmente está asignada a algún valor.

replace:Reemplaza la entrada de la clave especificada solo si está asignada actualmente al valor especificado.

replaceAll: Reemplaza el valor de cada entrada con el resultado de invocar la función dada en esa entrada hasta que se hayan procesado todas las entradas o la función arroje una excepción.

size:Devuelve el número de claves de esta tabla hash.

toString:Devuelve una representación de cadena de este objeto Hashtable en forma de un conjunto de entradas, encerradas en llaves y separadas por los caracteres ASCII ", " (coma y espacio).

values:Devuelve una vista de [**colección**](https://docs.oracle.com/javase/8/docs/api/java/util/Collection.html) de los valores contenidos en este mapa.

| TAD ArrayList |
| --- |
| ArrayList= {E element} |
| { inv:〈E != int,char,double…)〉} (Los elementos no deben ser tipos primitivos.) |
| Operaciones Primitivas:   * ArrayList: N/A → ArrayList. * ArrayList: Collection <? extends E> c → ArrayList * ArrayList: init initialCapacity →ArrayList * add: E e → boolean * add: int index, E element → void * addAll: Collection <? extends E> c → boolean * addAll: (int index, Collection <? extends E> c →boolean * clear: N/A →boolean * clone N/A → Object * contains: Object o → boolean * ensureCapacity: int minCapacity → void * get: int index → E element * indexOf: Object o → int * isEmpty: N/A → boolean * iterator N/A → iterator <E> * lastIndexOf: Object o → int * listIterator: N/A → ListIterator<E> * listIterator: int index → ListIterator<E> * remove: int index → E element * remove Object o →boolean * removeAll: Collection <?> c →boolean * removeRange: int fromIndex, int toIndex →void * retainAll: Collection <?> c → boolean * set: int index, E element → E * size: N/A → int * subList: int fromIndex, int toIndex →List<E> * toArray: N/A → Object [] * to Array: T[] a → <T> T[] * trimToSize: N/a → void |

ArrayList:Construye una lista vacía con una capacidad inicial de diez.

ArrayList:Construye una lista que contiene los elementos de la colección especificada, en el orden en que son devueltos por el iterador de la colección.

ArrayList:Construye una lista vacía con la capacidad inicial especificada.

add: Anexa el elemento especificado al final de esta lista.

add:Inserta el elemento especificado en la posición especificada de esta lista.

addAll:Anexa todos los elementos de la colección especificada al final de esta lista, en el orden en que los devuelve el Iterador de la colección especificada.

addAll:Inserta todos los elementos de la colección especificada en esta lista, comenzando por la posición especificada

clear:Quita todos los elementos de esta lista.

clone:Devuelve una copia superficial de esta instancia de ArrayList.

contains: Devuelve true si esta lista contiene el elemento especificado.

ensureCapacity:Aumenta la capacidad de esta instancia de ArrayList, si es necesario, para garantizar que pueda contener al menos el número de elementos especificados por el argumento de capacidad mínima.

get:Devuelve el elemento en la posición especificada en esta lista.

indexOf:Devuelve el índice de la primera aparición del elemento especificado en esta lista, o -1 si esta lista no contiene el elemento.

isEmpty:Devuelve true si esta lista no contiene elementos.

iterator:Devuelve un iterador sobre los elementos de esta lista en la secuencia adecuada.

lastIndexOf:Devuelve el índice de la última aparición del elemento especificado en esta lista, o -1 si esta lista no contiene el elemento

listIterator:Devuelve un iterador de lista sobre los elementos de esta lista (en la secuencia adecuada).

listIterator:Devuelve un iterador de lista sobre los elementos de esta lista (en la secuencia adecuada), comenzando por la posición especificada en la lista.

remove:Quita el elemento en la posición especificada en esta lista.

remove:Quita la primera aparición del elemento especificado de esta lista, si está presente.

removeAll:Quita de esta lista todos sus elementos contenidos en la colección especificada.

removeRange:Elimina de esta lista todos los elementos cuyo índice está entre , inclusive y , exclusivo

retainAll:Conserva sólo los elementos de esta lista contenidos en la colección especificada.

set:Reemplaza el elemento en la posición especificada en esta lista por el elemento especificado.

size:Devuelve el número de elementos de esta lista.

subList:Devuelve una vista de la parte de esta lista entre la especificada , inclusive y , exclusive

toArray:Devuelve una matriz que contiene todos los elementos de esta lista en la secuencia adecuada (del primer al último elemento).

toArray:Devuelve una matriz que contiene todos los elementos de esta lista en la secuencia adecuada (del primer al último elemento); el tipo de tiempo de ejecución de la matriz devuelta es el de la matriz especificada.

trimToSize:Recorta la capacidad de esta instancia de ArrayList para que sea el tamaño actual de la lista.