

UT03 Práctica 1

Dado un tipo de objeto literal para representar un libro con la siguiente estructura de propiedades.

```
let book = {
  ISBN: 978-84-9804-654-0,
  title: "El Quijote",
  author: "Miguel de Cervantes",
  publicationDate: new Date(1605, 0, 1),
  price: 20,
},
```

Deberás implementar las siguientes estructuras de datos a través de funciones.

1. Lista

Implementa un tipo de datos **lista** que permita almacenar objetos `Book` sobre un array. La lista debe tener un número máximo de elementos el cuál no se podrá rebasar. Utiliza una constante para almacenar dicho límite.

Las funciones/operaciones que debes implementar, teniendo en cuenta que `elem` estaremos haciendo referencia a un tipo de objetos `Book`, son:

Valor de retorno	función	Descripción
Array	<code>create</code>	Crea una lista con un array vacío para almacenar en su contenido los elementos de la lista.
Boolean	<code>isEmpty(list)</code>	Devuelve true o false en función de si la lista está vacía.
Boolean	<code>isFull(list)</code>	Devuelve true o false en función de si la lista está llena.
Number	<code>size(list)</code>	Devuelve el número de elementos de la lista.
Number	<code>add(list,elem)</code>	Añade un nuevo elemento al final de la lista. Devuelve el tamaño de la lista una vez añadido.
Number	<code>addAt(list,elem,index)</code>	Añade un nuevo elemento en la posición especificada en la lista. Devuelve el tamaño de la lista una vez añadido.
Book	<code>get(list,index)</code>	Devuelve el elemento de la lista de la posición indicada.
String	<code>toString(list)</code>	Devuelve la lista en formato cadena. El delimitador de elementos será "-".
Number	<code>indexOf(list,elem)</code>	Devuelve la posición del elemento indicado. Si el elemento no está en la lista devuelve -1. Realiza la comparación por ISBN.
Number	<code>lastIndexOf(list,elem)</code>	Devuelve la posición del elemento indicado comenzando por el final. Si el elemento no

		está en la lista devuelve -1. Realiza la comparación por ISBN.
Number	capacity(list)	Devuelve el máximo número de elementos que podemos tener en la lista.
-	clear(list)	Vacía la lista.
Book	firstElement(list)	Devuelve el primer elemento de la lista
Book	lastElement(list)	Devuelve el último elemento de la lista
Book	remove(list,index)	Elimina el elemento de la posición indicada. Devuelve el elemento borrado.
Boolean	removeElement(list,elem)	Elimina el elemento indicado de la lista. Devuelve true si se ha podido borrar el elemento, false en caso contrario.
Book	set(list,elem,index)	Reemplaza el elemento de la lista indicado por el índice. Devuelve el elemento que estaba anteriormente en la lista.

Tabla 1 Operaciones lista

1.1. Gestión de errores

En cuanto a la gestión de errores, las excepciones que deben lanzar cada función son las siguientes:

Función	Excepción
add(list,elem)	<ul style="list-style-type: none"> - El elemento no es un Book. Puedes comprobar si tiene la propiedad <i>ISBN</i> y <i>title</i>. - La lista está llena
addAt(list,elem,index)	<ul style="list-style-type: none"> - El elemento no es un Book - La lista está llena - El índice está fuera de los límites de la lista
get(list,index)	<ul style="list-style-type: none"> - El índice está fuera de los límites de la lista
indexOf(list,elem)	<ul style="list-style-type: none"> - El elemento no es un Book
lastIndexOf(list,elem)	<ul style="list-style-type: none"> - El elemento no es un Book
firstElement(list)	<ul style="list-style-type: none"> - La lista está vacía
lastElement(list)	<ul style="list-style-type: none"> - La lista está vacía
remove(list,index)	<ul style="list-style-type: none"> - El índice está fuera de los límites de la lista
removeElement(list,elem)	<ul style="list-style-type: none"> - El elemento no es un Book
set(list,elem,index)	<ul style="list-style-type: none"> - El elemento no es un Book - El índice está fuera de los límites de la lista

Tabla 2 Excepciones funciones de lista

2. Lista ordenada por ISBN

Implementa un tipo de datos **lista ordenada** que permita almacenar enteros sobre un array.

Las características de este tipo de datos son las mismas que para el tipo de datos lista, salvo que los elementos tienen una relación de orden entre sí, la cual será una ordenación ascendente de los mismos, en este caso por **ISBN**.

Las funciones/operaciones serán las mismas que para la lista, aunque con algunas excepciones. Las siguientes funciones no tendrían sentido dada la relación de orden ya que podríamos romper dicha relación.

- addAt
- lastIndexOf
- set

El resto de las funciones quedarían como siguen. Las funciones en negrita son las que han cambiado su funcionalidad.

Valor de retorno	función	Descripción
Array	create	Crea una lista con un array vacío para almacenar en su contenido los elementos de la lista.
Boolean	isEmpty(list)	Devuelve true o false en función de si la lista está vacía.
Boolean	isFull(list)	Devuelve true o false en función de si la lista está llena.
Number	size(list)	Devuelve el número de elementos de la lista.
Number	add(list,elem)	Añade un nuevo elemento a la lista manteniendo la relación de orden. Devuelve el tamaño de la lista una vez añadido.
Book	get(list,index)	Devuelve el elemento de la lista de la posición indicada.
String	toString(list)	Devuelve la lista en formato cadena. El delimitador de elementos será "-".
Number	indexOf(list,elem)	Devuelve la posición del elemento indicado. Si el elemento no está en la lista devuelve -1. La comparación la hacemos por ISBN.
Number	lastIndexOf(list,elem)	Devuelve la posición del elemento indicado comenzando por el final. Si el elemento no está en la lista devuelve -1. La comparación la hacemos por ISBN.
Number	capacity(list)	Devuelve el máximo número de elementos que podemos tener en la lista.
-	clear(list)	Vacía la lista.
Book	firstElement(list)	Devuelve el primer elemento de la lista
Book	lastElement(list)	Devuelve el último elemento de la lista
Book	remove(list,index)	Elimina el elemento de la posición indicada. Devuelve el elemento borrado.
Boolean	removeElement(list,elem)	Elimina el elemento indicado de la lista. Devuelve true si se ha podido borrar el elemento, false en caso contrario.

Tabla 3 Operaciones lista ordenada

2.1. Gestión de errores

En cuanto a la gestión de errores quedaría de la siguiente manera.

Función	Excepción
add(list,elem)	<ul style="list-style-type: none"> - El elemento no es un Book - La lista está llena
get(list,index)	<ul style="list-style-type: none"> - El índice está fuera de los límites de la lista

indexOf(list,elem)	- El elemento no es un Book
firstElement(list)	- La lista está vacía
lastElement(list)	- La lista está vacía
remove(list,index)	- El índice está fuera de los límites de la lista
removeElement(list,elem)	- El elemento no es un Number

Tabla 4 Excepciones funciones de lista ordenada

3. Conjunto

Implementa un tipo de datos **conjunto** que permita almacenar objetos **Book** **sobre un array**. El conjunto podrá tener un número ilimitado de elementos. Como indica el enunciado, solo podrás utilizar un array aunque sea mucho más fácil implementar este ejercicio con un objeto **Map**.

Las funciones/operaciones que debes implementar, teniendo en cuenta que **elem** estaremos haciendo referencia a un tipo de objetos **Book**, son:

Valor de retorno	función	Descripción
Array	create	Crea un conjunto con un array vacío para almacenar en su contenido los elementos de la lista.
Boolean	isEmpty(set)	Devuelve true o false en función de si la lista está vacía.
Number	size(set)	Devuelve el número de elementos de la lista.
Number	add(set,elem)	Añade un nuevo elemento al conjunto. Devuelve el tamaño del conjunto una vez añadido. El elemento no puede estar incluido en el conjunto, verificando que el ISBN no exista previamente.
Boolean	has(set, elem)	Indica si el elemento está incluido en el conjunto comparando con la propiedad ISBN.
String	toString(set)	Devuelve el conjunto en formato cadena. El delimitador de elementos será "-".
-	clear(set)	Vacía el conjunto.
Boolean	remove(set, elem)	Elimina el elemento del conjunto utilizando el ISBN. Devuelve true si se ha eliminado del conjunto.

Tabla 5 Operaciones conjunto

3.1. Gestión de errores

En cuanto a la gestión de errores, las excepciones que deben lanzar cada función son las siguientes:

Función	Excepción
add(set,elem)	- El elemento no es un Book. Puedes comprobar si tiene la propiedad <i>ISBN</i> y <i>title</i> . - El ISBN ya está incluido en el conjunto
has(set, elem)	- El elemento no es un Book
remove(list,index)	- El elemento no es un Book

Tabla 6 Excepciones funciones del conjunto

4. Expresión Regular ISBN

Para cada uno de los ejercicios anteriores deberás comprobar que el objeto `Book` introducido tiene un formato correcto de ISBN. Deberás emplear una Expresión Regular para realizar la comprobación. En caso de que la propiedad ISBN no cumpla con el formato deberás lanzar una excepción indicando la circunstancia. Esta validación deberás integrarla en cada punto donde trabajes con un objeto `Book`.

5. Rúbrica y recursos solicitados

Por cada ejercicio deberás entregar un fichero HTML etiquetado con el nombre del ejercicio, y el fichero JavaScript que realice la implementación. No será necesario implementar interfaz gráfico para cada ejercicio, ya que esto lo trabajaremos más adelante.

Por cada ejercicio será necesario implementar una función de testeo donde verifiques el funcionamiento de cada una de las funciones que has implementado, en un fichero `.js` independiente. En esta función de testeo mostrará a través de la consola la traza de ejecución de operaciones. Tienes como ejemplo las implementaciones de Pila y Cola. **Si no se implementa esta función de testeo, la nota del ejercicio será un 0, independientemente de que las funciones estén bien implementadas. Recuerda que la función de testeo debe ser lo suficientemente rica en ejemplos que visualice el comportamiento de tu implementación.**

Ejercicio	Nota
Ejercicio 1: Lista de libros	4 puntos
Ejercicio 2: Lista ordenadas por ISBN	1,5 puntos
Ejercicio 3: Conjunto de libros	2,5 puntos
Ejercicio 4: Expresión Regular ISBN	1 punto
Código comentado	1 punto

Tabla 7 Rúbrica