





TAD	Stack		
Grafico			
Invariantes	<ul style="list-style-type: none"><li>• Sólo se puede agregar un nuevo elemento al final de la pila</li><li>• Sólo se puede eliminar un elemento al final de la pila</li></ul>		
Operaciones	 push	S x Book	-> S
	 pop	S	-> Book
	 peek	S	-> Book
	 size	S	-> S

TAD	HashTable			
Grafico	<p>Diagram illustrating a HashTable structure. The table has 7 slots (0 to 6). Slots 1, 3, and 6 contain a black dot and point to external nodes. Slot 1 points to a node with <math>K_1</math> and <math>E_1</math>. Slot 3 points to a node with <math>K_2</math> and <math>E_2</math>. Slot 6 points to a node with <math>K_3</math> and <math>E_3</math>. Slots 0, 2, 4, and 5 are empty.</p>			
Invariantes	<ul style="list-style-type: none"> <li>Insertar un elemento en la tabla debe costar en complejidad temporal <math>O(1)</math></li> <li>Buscar un elemento en la tabla debe costar en complejidad temporal <math>O(1)</math></li> <li>Borrar un elemento en la tabla debe costar en complejidad temporal <math>O(1)</math></li> </ul>			
Operaciones	insert	$T \times K$	->	Ninguna
	search	$K$	->	Elemento
	delete	$K$	->	Elemento
	reHashing	Ninguna	->	Ninguna
	calculateChargeFactor	Ninguna	->	Ninguna
	funtionHash	$K \times i$	->	Double
	codeHash	$K$	->	Integer
	funtionOne	Integer	->	Integer
	funtionTwo	Integer	->	Integer

TAD	Queue		
Grafico	<p>Q=</p>		
Invariantes	<ul style="list-style-type: none"> <li>El primer elemento en entrar es el primero en salir</li> <li>Solo se pueden ingresar elementos en la ultima posicion</li> </ul>		
Operaciones	Enqueue	Q x Book	-> Q
	Back	Q	-> Book
	Dequeue	Q	-> Book
	Front	Q	-> Book

TAD	Priority Queue Min Heap		
Grafico	<p>P=</p>		
Invariantes	<ul style="list-style-type: none"> <li>El elemento que está de padre debe ser menor que sus hijos.</li> <li>El árbol debe de ser semicompleto.</li> </ul>		
Operaciones	ExtractMin	P	-> P
	insert	P X E	-> P
	isEmpty	P	-> Boolean
	Minimum	P	-> E