|  |
| --- |
| TAD Stack |
| Stack = |
| 0≤ n ꓥ Size(Stack) = n ꓥ top = |
| * Stack - Stack * push Stack x Element Stack * pop Stack Stack * top Stack Element * isEmpty Stack Boolean |

|  |
| --- |
| **Stack**  Crea un stack vacio  {pre: -}  {post: Stack s = 0} |

|  |
| --- |
| **push**  Añade un nuevo element e al stack s  {pre: Stack s = y elemento e or s = 0 y elemento e}  {post: Stack s = or s =} |

|  |
| --- |
| **pop**  Extrae los elementos del stack s, el elemento que se haya insertado recientemente  {pre: Stack s ≠ 0 i.e. s = }  {post: Stack s = } |

|  |
| --- |
| **top**  recupera el valor del elemento que está en la parte superior del stack  {pre: Stack s ≠ 0 i.e. s= }  {post: Elemento } |

|  |
| --- |
| **isEmpty**  determina si el Stack s esta vacio o no  {pre: Stack s}  {post: True si S = 0, False si s ≠ 0} |

|  |
| --- |
| TAD Queue |
| Stack = |
| 0≤ n ꓥ Size(Queue) = n ꓥ front = |
| * Queue - Queue * enqueue Queue x Element Queue * dequeue Queue Element * front Queue Element * isEmpty Queue Boolean * ~Queue - |

|  |
| --- |
| **Queue**  Crea un Queue vacio  {pre: -}  {post: Queue q = 0} |

|  |
| --- |
| **enqueue**  Inserta un nuevo elemento e al final de la queue q  {pre: Queue q = y elemento e or q = 0 y elemento e}  {post: Queue q = or q =} |

|  |
| --- |
| **dequeue**  Extrae el elemento en que esta al principio de la queue  {pre: Queue q ≠ 0 i.e. q = }  {post: Queue q = y el elemento } |

|  |
| --- |
| **Front**  recupera el valor del elemento que está en la parte superior del Queue  {pre: Queue q ≠ 0 i.e. q = }  {post: Elemento } |

|  |
| --- |
| **isEmpty**  determina si la Queue esta vacia o no  {pre: Queue q}  {post: True si q = 0, False si q ≠ 0} |

|  |
| --- |
| TAD Hash Table |
| Table =  K1  K2  K3 |
| Invariante |
| * Table - Table * insert Table x key Table * search Table x key key * delete table x key table * isEmpty Table Boolean * tableLength table Int |

|  |
| --- |
| **Table**  Crea una table vacia |

|  |
| --- |
| **insert**  inserta un nuevo ítem en la table de acuerdo a su orden |

|  |
| --- |
| **search**  Busca un elemento en la table de acuerdo a su clave |

|  |
| --- |
| **delete**  elimina un elemento con una clave de búsqueda dada en la table |

|  |
| --- |
| **IsEmpty**  Determina si la table esta vacia o no  {pre: Table t}  {post: True si t = 0, False si t ≠ 0} |

|  |
| --- |
| **tableLength**  determina el número de elementos en la tabla  {pre: Table t}  {post: |