|  |
| --- |
| **TAD** **Stack** |
|  |
| { *inv: Los procedimientos de acceso permiten inserciones y eliminación de elementos solo en un extremo de la secuencia a excepción del ítem 1(top – last in first out)* } |
| Operaciones Primitivas:   * Push: *Stack x item* -> Stack * Pop: *Stack* -> Item * isEmpty: *Stack* -> Booleano * length: *Stack* -> Entero * top: *Stack* -> Item |

|  |
| --- |
| **Push(**Stack, item**)**  “Agrega un nuevo elemento al top de la pila”  { *pre:* TRUE }  { *post: Stack.top = item* } |

|  |
| --- |
| **Pop(**Stack**)**  “Devuelve el item en el top y al mismo tiempo lo elimina. Por lo tanto, modifica el Stack”  { *pre:* TRUE }  { *post: Si Stack.isEmpty != false se lanza la excepción de que no es posible eliminar el objeto,*  *Si Stack.lenght > 1 o Stack.isEmpty = false el item en el top es eliminado de la pila* } |

|  |
| --- |
| **IsEmpty (**Stack**)**  “Informa si la pila está vacía.”  { *pre:* TRUE }  { *post: Si la Stack.lenght = 1 entonces Stack.isEmpty = true,*  *Si la Stack.lenght > 1 entonces Stack.isEmpty = false* } |

|  |
| --- |
| **Length(**Stack**)**  “Devuelve el tamaño de la pila o stack”  { *pre:* TRUE }  { *post: Determina el número de elementos que contiene el Stack* } |

|  |
| --- |
| **Top(**Stack**)**  “Devuelve el item en el top del Stack”  { *pre:* TRUE }  { *post: Si Stack.isEmpty != true el item en el top es retornado sin generar cambios en la pila,*  *Si Stack.isEmpty = true se lanza una excepción informando que la pila esta vacía* } |