|  |
| --- |
| **TAD** Queue. |
|  |
| {inv:} |
| **Operaciones primitivas:**   * Enqueue: Nodo 🡪 Queue * Dequeue: 🡪 Queue * Front: 🡪 Queue * IsEmpty: 🡪 Boolean * Size: 🡪 Entero |

**Operaciones modificadoras.**

|  |
| --- |
| **Enqueue ()**  “Agrega un nodo a la cola”  {pre: Nodo no está vacío}  {post: Queue.Size() 0} |

|  |
| --- |
| **Dequeue ()**  “Elimina y retorna el nodo que esta al principio de la cola”  {pre: Queue.Size() 0}  {post: Queue.Nodo & Queue.Size() – 1} |

**Operaciones analizadoras.**

|  |
| --- |
| **Front ()**  “Retorna el nodo que esta al principio de la cola”  {pre: Queue.Size() 0}  {post: Queue} |

|  |
| --- |
| **IsEmpty ()**  “Comprueba el estado de la cola”  {pre: TRUE}  {post: Retorna TRUE si la cola esta vacía. De lo contrario, retorna FALSE} |

|  |
| --- |
| **Size ()**  “Retorna el número de elementos almacenados en la cola”  {pre: TRUE}  {post: Retorna un Z+ *n*. De lo contrario, retorna 0} |

|  |
| --- |
| **TAD** Stack. |
|  |
| {inv:} |
| **Operaciones primitivas:**   * Push: Nodo 🡪 Stack * Pop: 🡪 Stack * Top: 🡪 Stack * IsEmpty: 🡪 Boolean * Size: 🡪 Entero |

**Operaciones modificadoras.**

|  |
| --- |
| **Push ()**  “Agrega un nodo a la pila”  {pre: Nodo no está vacío}  {post: Stack.Size() 0} |

|  |
| --- |
| **Pop ()**  “Retorna y elimina el nodo que se encuentra en la cima de la pila”  {pre: Stack.Size() 0}  {post: Stack.Nodo & Stack.Size() – 1} |

|  |
| --- |
| **Top ()**  “Retorna el nodo que está en la cima de la pila”  {pre: Stack.Size() 0}  {post: Stack.Nodo} |

**Operaciones Analizadoras.**

|  |
| --- |
| **IsEmpty ()**  “Comprueba el estado de la pila”  {pre: TRUE}  {post: Retorna TRUE si la pila esta vacía. De lo contrario, retorna FALSE} |
| **Size ()**  “Retorna el número de elementos almacenados en la pila”  {pre: TRUE}  {post: Retorna un Z+ *n*. De lo contrario, retorna 0} |