TAD Queue

Queue = {First = <first>, First.Next = <next>, Last = <last>, Value = <value>}

Invariant: First != nill, Last != nill

Construction operations:
*Create : → Queue
Modifier operations:
*addElement: QueuexValue → Queue
*remove: QueuexValue → Queue
Operaciones analizadoras:
*isEmpty: Queue → booleano
*size: Queue → Integer

Create (value)
"Creates an element of the Queue with the first and last element null"

{pre: TRUE }

{post: Heap = {First = <nill>, Last = <nill>}

addElement (elementQueue)
"Inserts an element passed by parameter on the Queue structure putting it in the last position."

{pre: TRUE }

{post: Queue.Last = <elementQueue>}

remove()
"Removes the First element from the Queue, putting the First.Next as First".

{ pre: the Queue has at least one element }

{ post: First element is returned}

isEmpty(Queue):

"Informs if the Queue is empty."

{pre: TRUE}
{pre: Queue={First:<first>,...}

{post: False if the Queue.First!= nil, True otherwise}

size(Queue):

"Returns an Integer that represents the number of elements currently inserted in the Queue."

{pre: TRUE}
{pre: Queue={Queue:<first>,...}

{post: n | n ∈ Z+}