|  |
| --- |
| **TAD** Stack |
| Stack = {ArrayList<T> = {stack}} |
| { inv:  stack.size() >= 0} |
| Operaciones primitivas:  ·         Stack  ·         add    Node  ·         get    Node  ·         isEmpty  ·         pop Node  ·         peek  size |

|  |
| --- |
| Stack()  **“Constructora”**  “Crea un nuevo Stack con un ArrayList vacío”    { pre: TRUE }    { post:  list = { First: null, size = 0 } } |

|  |
| --- |
| add(T values)  **“Modificadora”**  “Añade un nuevo elemento a la lista enlazada, poniéndolo en la posición necesaria para que la lista siempre esté ordenada.”    { pre:  list = { First: null, … } ⋀ element ¡= null }  { post:  list = { First: Node<element>, … size = size + 1 } ⋀ element ¡= null } |

|  |
| --- |
| get(int index)  **“Analizadora”**  “Elimina el nodo con el elemento buscado”   { pre: list = {First: Node<element>, … size = 1 } ⋀ element != null}  { post: list = {First: null, … , size  = size - 1 } ⋀ element != null} |

|  |
| --- |
| isEmpty()  **“Analizadora”**  “Devuelve el tamaño de la lista enlazada”  { pre: list = {First: Node<>, size >= 0} }  { pos: < size > } |

|  |
| --- |
| pop()  **“Modificadora”**  “Devuelve el String con la lista enlazada y sus elementos en forma de array”  { pre: list = {First: Node<>, size >= 0} }  { pos: {“Node1, Node2, Node3 … NodeN”}} |

|  |
| --- |
| peek()  **“Modificadora”**  “Hace el llamado al método nodesToString y devuelve un String de los nodos de la lista enlazada cubiertos con corchetes ‘[...]’. ”  { pre: list = {First: Node<>, size >= 0} }  { pos: {“[ Node1, Node2, Node3 … NodeN ]”}} |

|  |
| --- |
| size()  **“Analizadora”**  “Hace el llamado al método nodesToString y devuelve un String de los nodos de la lista enlazada cubiertos con corchetes ‘[...]’. ”  { pre: list = {First: Node<>, size >= 0} }  { pos: {“[ Node1, Node2, Node3 … NodeN ]”}} |