TAD lista enlazada ordenada

listaEnlazadaOrd = {primerNodo= <k>, siguienteNodo= <k>, donde cada elemento es Comparable tal que ki, es menor que ki +1, && listEnlazadaOrd ={k1,k2,...,kn } donde cada elemento tiene una relacion con el sigueinte elemento en la lista}

crearLista:	>		listaEnlazadaOrd	Operaciones de construcción.
anadirElem:	listaEnlazadaOrd x valor	>	listaEnlazadaOrd	Operaciones de modificación.
eliminarElem:	listaEnlazadaOrd x valor	>	listaEnlazadaOrd	
estaVacio:	listaEnlazadaOrd	>	valor de verdad	Operaciones analizadoras.
tamaño:	listaEnlazadaOrd	>	entero	
buscar:	listaEnlazadaOrd x valor>		valor	
1				

crearLista()

"crea una nueva lista con los nodos apuntando a null" {pre: VERDAD}

{post: listaEnlazadaOrd = { primerNodo = <null>, siguienteNodo = <null>}}

anadirElem(listaEnlazadaOrd, e)

"agrega ordenadamente el elemento dado a la posción correcta en la lista" {pre: listaEnlazadaOrd = {primerNodo k1, sigNodo = k2}, e }

{post: listaEnlazadaOrd= { primerNodo != <null>, sigNodo != <null>}}

eliminarElem(listaEnlazadaOrd, e)

"Elimina el elemento dado de la posicion en la que se encontraba en el arreglo, si el elmento no existe en la lista no se realizan cambios" {pre: listaEnlazadaOrd = {primerNodo = k, nextNode = k}, e }

{post: si el elemento existia en la lista : listaEnlazadaOrd.tamaño = listaEnlazada.tamaño - 1. de lo contrario no se realizan cambios a la lista}

estaVacia(listaEnlazadaOrd)

"retorna el valor de verdad que refleja si la lista tiene o no al menos un elemento" {pre: listaEnlazadaOrd = {primerNodo = k, sigNode = k} }

{post: VERDADERO si lista.primerNodo== null, de otra manera será FALSO}

tamano(listaEnlazadaOrd)

"retorna un entero que representa la cantidad de nodos o elementos presentes en la lista"

{pre: listaEnlazadaOrd = {primerNodo = k, sigNodo = k} }

{post: enteroRetornado >= 0 }

buscar(listaEnlazadaOrd, entero)

"retorna el valor o elemento del nodo ubicado en la posicion correspondiente al numero entregado como parametro"

{pre: listaEnlazadaOrd = {primerNodo = k, nextNode = k} && number >= 0 }

{post: si el elemento existe retorna el valor o elemento del nodo ubicado en el numero correspondiente al numero entregado como parametro, de lo contrario retorna FALSO}