Plantilla de Especificación Operacional

|  |  |  |  |
| --- | --- | --- | --- |
| **Estudiante** |  | **Fecha** |  |
| **Programa** |  | **No. de Pgma** |  |
| **Instructor** |  | **Lenguaje** |  |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Número de escenario** |  | **Objetivo del usuario** |  | |
| **Objetivo del escenario** | |  | | |
| **Fuente** | **Paso #** | **Acción** | | **Comentarios** |
|  |  |  | |  |
|  |  |  | |  |
|  |  |  | |  |
|  |  |  | |  |
|  |  |  | |  |
|  |  |  | |  |
|  |  |  | |  |
|  |  |  | |  |
|  |  |  | |  |
|  |  |  | |  |
|  |  |  | |  |
|  |  |  | |  |
|  |  |  | |  |
|  |  |  | |  |
|  |  |  | |  |
|  |  |  | |  |
|  |  |  | |  |
|  |  |  | |  |
|  |  |  | |  |
|  |  |  | |  |
|  |  |  | |  |
|  |  |  | |  |
|  |  |  | |  |
|  |  |  | |  |
|  |  |  | |  |
|  |  |  | |  |

Plantilla de Especificación Funcional

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Estudiante** | |  | | | **Fecha** |  |
| **Programa** | |  | | | **# de Pgma** |  |
| **Instructor** | |  | | | **Lenguaje** |  |
|  | | | | | | |
| **Nombre de la clase** | | | NodoArbol | | | |
| **Clase(s) padre(s)** | | |  | | | |
|  | | |  | | | |
|  | | |  | | | |
|  | | |  | | | |
|  | | | | | | |
| **Atributos** | | | | | | |
|  | **Declaración** | | | **Descripción** | | |
|  | Tipo t informacion | | | Informacion a guardar en el nodo | | |
|  | apuntador a nodoarbol izq | | | Apuntador al hijo izquierdo | | |
|  | apuntador a nodoarbol der | | | Apuntador al hijo derecho | | |
|  | entero numHijos | | | Cantidad de hijos | | |
|  |  | | |  | | |
|  |  | | |  | | |
|  | | | | | | |
| **Elementos** | | | | | | |
|  | **Declaración** | | | **Descripción** | | |
|  | sin tipo NodoArbol(tipo t valor) | | | Constructor de la clase, guarda el valor e inicializa los apuntadores a los hijos en nulo y la cantidad de hijos en 0 | | |
|  | Tipo t traeInformacion() | | | Regresa la información guardada en el nodo | | |
|  | apuntador a nodoarbol traeHijoIzq() | | | Regresa el apuntador al hijo izquierdo | | |
|  | apuntador a nodoarbol traeHijoDer() | | | Regresa el apuntador al hijo derecho | | |
|  | entero traeCantHijos() | | | Regresa la cantidad de hijos | | |
|  | sin tipo guardaInformacion(tipo t valor) | | | Guarda el valor | | |
|  | sin tipo guardaHijoIzq(apuntador a nodo arbol hizq) | | | Guarda el apuntador al hijo izquierdo | | |
|  | sin tipo guardaHijoDer(apuntador a nodo arbol hder) | | | Guarda el apuntador al hijo derecho | | |
|  | sin tipo guardaCantHijos(entero cuantos) | | | Guarda la cantidad de hijos | | |
|  |  | | |  | | |
|  |  | | |  | | |
|  |  | | |  | | |

Plantilla de Especificación Funcional

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Estudiante** | |  | | | **Fecha** |  |
| **Programa** | |  | | | **# de Pgma** |  |
| **Instructor** | |  | | | **Lenguaje** |  |
|  | | | | | | |
| **Nombre de la clase** | | | ArbolBinario | | | |
| **Clase(s) padre(s)** | | |  | | | |
|  | | |  | | | |
|  | | |  | | | |
|  | | |  | | | |
|  | | | | | | |
| **Atributos** | | | | | | |
|  | **Declaración** | | | **Descripción** | | |
|  | apuntador a nodoarbol raiz | | | Apuntador a la raíz del árbol binario | | |
|  |  | | |  | | |
|  |  | | |  | | |
|  |  | | |  | | |
|  |  | | |  | | |
|  |  | | |  | | |
|  | | | | | | |
| **Elementos** | | | | | | |
|  | **Declaración** | | | **Descripción** | | |
|  | Sin tipo ArbolBinario() | | | Inicializa en nulo a la raiz | | |
|  | Sin tipo agregarNodo(Tipo t valor) | | | Agrega un nodo al árbol y le pone el valor | | |
|  | Sin tipo eliminarNodo(Tipo t valor) | | | Eliminar el primer nodo que encuentre que tenga almacenado valor | | |
|  | booleano estaVacio() | | | Regresa verdadero si el árbol está vacío y si no regresa falso | | |
|  | Sin tipo recorrePreorden() | | | Visita los nodos del árbol haciendo un recorrido en preorden | | |
|  | Sin tipo recorrePostOrden() | | | Visita los nodos del árbol haciendo un recorrido en postorden | | |
|  | Sin tipo recorreInOrden() | | | Visita los nodos del árbol haciendo un recorrido en inorden | | |
|  | Sin tipo agregarNodo(apuntador a nodoarbol padre, apuntador a nodoarbol nuevoNodo) | | | Funcion recursiva para agregar el nuevo Nodo al subárbol que empieza en padre | | |
|  | Apuntador a nodoArbol encontrarPapaNodo(Tipo t valor) | | | Función que regresa el apuntador al nodo padre del nodo cuya información es igual a valor | | |
|  | Apuntador a nodoArbol encontrarPapaNodo(apuntador a nodo árbol padre, Tipo t valor) | | | Funcion recursiva para regresar el apuntador al nodo padre del nodo cuya información es igual a valor del subárbol que empieza en padre | | |
|  | Apuntador a nodoArbol encontrarHoja(apuntador a nodo árbol padre) | | | Encuentra la primera hoja del árbol que empieza en padre | | |
|  |  | | |  | | |

Plantilla de Especificación Lógica

|  |  |  |  |
| --- | --- | --- | --- |
| **Estudiante** |  | **Fecha** |  |
| **Programa** |  | **# de Pgma** |  |
| **Instructor** |  | **Lenguaje** |  |

|  |  |
| --- | --- |
| **Referencias** | Constructor de la clase ArbolBinario |
| **De Diseño** |  |
|  |  |
|  |  |
|  |  |

|  |  |
| --- | --- |
| **Parámetros** | Ninguno |
|  |  |
|  |  |
|  |  |
|  |  |

|  |
| --- |
| Asignar el valor de nulo a raiz |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |

Plantilla de Especificación Lógica

|  |  |  |  |
| --- | --- | --- | --- |
| **Estudiante** |  | **Fecha** |  |
| **Programa** |  | **# de Pgma** |  |
| **Instructor** |  | **Lenguaje** |  |

|  |  |
| --- | --- |
| **Referencias** | Método agregarNodo |
| **De Diseño** |  |
|  |  |
|  |  |
|  |  |

|  |  |
| --- | --- |
| **Parámetros** | Tipo t valor |
|  |  |
|  |  |
|  |  |
|  |  |

|  |
| --- |
| Declarar nuevoNodo como apuntador a NodoArbol |
| Pedir al sistema operativo el espacio para nuevoNodo asignándole valor |
| Si raíz es nulo |
| Asignar nuevoNodo a raiz |
| Sino |
| Llamar a función agregarNodoRecursivo pasándole raíz y nuevoNodo |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |

Plantilla de Especificación Lógica

|  |  |  |  |
| --- | --- | --- | --- |
| **Estudiante** |  | **Fecha** |  |
| **Programa** |  | **# de Pgma** |  |
| **Instructor** |  | **Lenguaje** |  |

|  |  |
| --- | --- |
| **Referencias** | Método agregarNodo |
| **De Diseño** |  |
|  |  |
|  |  |
|  |  |

|  |  |
| --- | --- |
| **Parámetros** | apuntador a NodoArbol padre, apuntador a NodoArbol nuevoNodo |
|  |  |
|  |  |
|  |  |
|  |  |

|  |
| --- |
| Declarar cantidadHijos como entero |
| Declarar auxiliar como apuntador a NodoArbol |
| Asignar a cantidadHijos el resultado de llamar a la función traeCantHijos() de padre |
| Si cantidadHijos es igual a cero |
| Asignar nuevoNodo al hijo izquierdo de padre (llamando al método guardaHijoIzq) |
| Actualizar a 1 la cantidad de hijos de padre (llamando al método guardaCantHijos(1)) |
| FinSi |
| Si cantidadHijos es igual a uno |
| Asignar a auxiliar el hijoDerecho de padre (usando la función traeHijoDer) |
| Si auxiliar es nulo |
| Asignar nuevoNodo al hijo derecho de padre(llamando al método guardaHijoDer) |
| Sino |
| Asignar nuevoNodo al hijo izquierdo de padre (llamando al método guardaHijoIzq) |
| FinSi |
| Actualizar a 2 la cantidad de hijos de padre (llamando al método guardaCantHijos(2)) |
| FinSi |
| Si cantidadHijos es igual a dos |
| Asignar a auxiliar el hijoDerecho de padre (usando la función traeHijoDer) |
| Llamar a la función agregarNodo pasándole auxiliar y nuevoNodo |
| FinSi |

Plantilla de Especificación Lógica

|  |  |  |  |
| --- | --- | --- | --- |
| **Estudiante** |  | **Fecha** |  |
| **Programa** |  | **# de Pgma** |  |
| **Instructor** |  | **Lenguaje** |  |

|  |  |
| --- | --- |
| **Referencias** | Apuntador a nodoArbol encontrarPapaNodo |
| **De Diseño** |  |
|  |  |
|  |  |
|  |  |

|  |  |
| --- | --- |
| **Parámetros** | Tipo t valor |
|  |  |
|  |  |
|  |  |
|  |  |

|  |
| --- |
| Declarar auxiliar como Apuntador a nodoArbol, declarar cantHijos como entero |
| Declarar hijo como Apuntador a nodoArbol |
| Inicializar auxiliar en nulo |
| Inicializar cantHijos en cero |
| Si raíz no es nulo |
| Asignar a cantHijos el resultado de llamar a la función traeCantHijos de raiz |
| Si cantHijos no es cero |
| Asignar a auxiliar el resultado de llamar a encontrarPapaNodo(raiz,valor) |
| FinSi |
| Finsi |
| Regresar auxiliar |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |

Plantilla de Especificación Lógica

|  |  |  |  |
| --- | --- | --- | --- |
| **Estudiante** |  | **Fecha** |  |
| **Programa** |  | **# de Pgma** |  |
| **Instructor** |  | **Lenguaje** |  |

|  |  |
| --- | --- |
| **Referencias** | Apuntador a nodoArbol encontrarPapaNodo |
| **De Diseño** |  |
|  |  |
|  |  |
|  |  |

|  |  |
| --- | --- |
| **Parámetros** | apuntador a nodo árbol padre, Tipo t valor |
|  |  |
|  |  |
|  |  |
|  |  |

|  |
| --- |
| Declarar auxiliar como apuntador a nodo árbol |
| Declarar cantHijos como entero |
| Declarar hijo como apuntador a nodo árbol |
| Asignar nulo a auxiliar |
| Asignar a cantHijos el resultado de llamar a la función traeCantHijos de padre |
| Si cantHijos es igual a uno |
| Asignar a hijo el resultado de llamar a la función traeHijoDer de padre |
| Si hijo es nulo |
| Asignar a hijo el resultado de llamar a la función traeHijoIzq de padre |
| FinSi |
| Si el resultado de llamar a la función traeInformacion de hijo es igual a valor |
| Asignar padre a auxiliar |
| Si no |
| Asignar a auxiliar el resultado de llamar la función encontrarPapaNodo(hijo, valor) |
| FinSi |
| FinSi |
| Si cantHijos es igual a dos |
| Asignar a hijo el resultado de llamar a la función traeHijoDer de padre |
| Si el resultado de llamar a la función traeInformacion de hijo es igual a valor |
| Asignar padre a auxiliar |
| Si no |
| Asignar a hijo el resultado de llamar a la función traeHijoIzq de padre |
| Si el resultado de llamar a la función traeInformacion de hijo es igual a valor |
| Asignar padre a auxiliar |
| Si no |
| Asignar a auxiliar el resultado de llamar a la función encontrarPapaNodo(hijo, valor) |
| Si auxiliar es nulo |
| Asignar a hijo el resultado de llamar a la función traeHijoDer de padre |
| Asignar a auxiliar el resultado de llamar a la función encontrarPapaNodoR(hijo, valor) |
| FinSi |
| FinSi |
| FinSi |
| FinSi |
| Regresar auxiliar |
|  |

Plantilla de Especificación Lógica

|  |  |  |  |
| --- | --- | --- | --- |
| **Estudiante** |  | **Fecha** |  |
| **Programa** |  | **# de Pgma** |  |
| **Instructor** |  | **Lenguaje** |  |

|  |  |
| --- | --- |
| **Referencias** | Sin tipo eliminarNodo |
| **De Diseño** |  |
|  |  |
|  |  |
|  |  |

|  |  |
| --- | --- |
| **Parámetros** | Tipo t valor |
|  |  |
|  |  |
|  |  |
|  |  |

|  |
| --- |
| Declarar padre como apuntador a NodoArbol |
| Declarar hoja como apuntador a NodoArbol |
| Declarar hijo como apuntador a NodoArbol |
| Declarar nietoIzq como apuntador a NodoArbol |
| Declarar nietoDer como apuntador a NodoArbol |
| Declarar cantHijos, cantNietos y valorAux como enteros |
| Declarar auxiliar como apuntador a NodoArbol |
| Declarar izquierdo como booleano e inicializarlo en verdadero |
| Si raíz no es nulo |
| Asignar a padre el resultado de llamar a la función encontrarpapaNodo(valor) |
| Si padre es nulo |
| Si el resultado de traerInformacion de raíz es igual a valor |
| Asignar a cantHijos el resultado de llamar a traeCantHijos de raíz |
| Si cantHijos es 0 |
| Borrar la raiz |
| Asignar nulo a raiz |
| FinSi de cantHijos es 0 |
| Si cantHijos es 1 |
| Asignar a auxiliar el resultado de llamar a traeHijoDer de raiz |
| Si auxiliar es nulo |
| Asignar a auxiliar el resultado de llamar a traeHijoIzq de raiz |
| FinSi |
| Borrar la raiz |
| Asignar auxiliar a la raiz |
| FinSi de cantHijos es 1 |
| Si cantHijos es 2 |
| Asignar a hoja el resultado de llamar a encontrarHoja(raíz) |
| Asignar a valorAux el resultado de llamar a traeInformacion de hoja |
| Cambiar la información de la raíz por valorAux (llamando a guardaInformacion de raíz) |
| Asignar a padre el resultado de llamar a encontrarpapaNodo(valorAux) |
| Si el resultado de traeHijoIzq de padre es igual a hoja |
| Guardar nulo en el izq de padre |
| Sino |
| Guardar nulo en el der de padre |
| FinSi |
| Borrar hoja |
| FinSi de cantHijos es 2 |
| FinSi de info de raíz es igual a valor |
| Sino (del si de padre es nulo) |
| Asignar a cantHijos el resultado de llamar a traeCantHijos de padre |
| Asignar a hijo el resultado de llamar a traeHijoIzq de padre |
| Si hijo es nulo o hijo traeInformacion no es valor |
| Asignar a hijo el resultado de llamar a traeHijoDer de padre |
| Asignar falso a izquierdo |
| FinSi (de hijo es nulo) |
| Asignar a cantNietos el resultado de llamar a traeCantHijos de hijo |
| Si cantNietos es cero |
| Si izquierdo es verdadero |
| Asignar nulo a izq de padre |
| Si no |
| Asignar nulo a der de padre |
| Fin si (izquierdo es verdadero) |
| Borrar hijo |
| Poner en cero cantidad de hijos de padre |
| FinSi (cantNietos es cero) |
| Si cantNietos es uno |
| Asignar a nietoIzq el resultado de llamar a traeHijoIzq de hijo |
| Asignar a nietoDer el resultado de llamar a traeHijoDer de hijo |
| Si nietoIzq es nulo |
| Si izquierdo es verdadero |
| Asignar nietoDer a izq de padre |
| Sino |
| Asignar nietoDer a der de padre |
| FInsi (izquierdo es verdadero) |
| Sino (de nietoIzq nulo) |
| Si izquierdo es verdadero |
| Asignar nietoIzq a izq de padre |
| Sino |
| Asignar nietoIzq a der de padre |
| FInsi (izquierdo es verdadero) |
| FinSi de nietoIzq nulo |
| Borrar a hijo |
| FinSi de cantNietos es uno |
| Si cantNietos es dos |
| Asignar a hoja el resultado de llamar a encuentraHoja(hijo) |
| Asignar a valorAuxiliar la información de hoja |
| Asignar valorAuxiliar a la información de hijo |
| Llamar a eliminarNodo con valorAuxiliar |
| FinSi cantNietos es dos |
| Finsi de padre es nulo |
| Finsi de raíz no es nulo |

Plantilla de Especificación Lógica

|  |  |  |  |
| --- | --- | --- | --- |
| **Estudiante** |  | **Fecha** |  |
| **Programa** |  | **# de Pgma** |  |
| **Instructor** |  | **Lenguaje** |  |

|  |  |
| --- | --- |
| **Referencias** |  |
| **De Diseño** |  |
|  |  |
|  |  |
|  |  |

|  |  |
| --- | --- |
| **Parámetros** |  |
|  |  |
|  |  |
|  |  |
|  |  |

|  |
| --- |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |

Plantilla de Especificación Lógica

|  |  |  |  |
| --- | --- | --- | --- |
| **Estudiante** |  | **Fecha** |  |
| **Programa** |  | **# de Pgma** |  |
| **Instructor** |  | **Lenguaje** |  |

|  |  |
| --- | --- |
| **Referencias** |  |
| **De Diseño** |  |
|  |  |
|  |  |
|  |  |

|  |  |
| --- | --- |
| **Parámetros** |  |
|  |  |
|  |  |
|  |  |
|  |  |

|  |
| --- |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |

Plantilla de Especificación Lógica

|  |  |  |  |
| --- | --- | --- | --- |
| **Estudiante** |  | **Fecha** |  |
| **Programa** |  | **# de Pgma** |  |
| **Instructor** |  | **Lenguaje** |  |

|  |  |
| --- | --- |
| **Referencias** |  |
| **De Diseño** |  |
|  |  |
|  |  |
|  |  |

|  |  |
| --- | --- |
| **Parámetros** |  |
|  |  |
|  |  |
|  |  |
|  |  |

|  |
| --- |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |