

Grupo 3 Taller WBS - Estructura De Datos



**Juliana Aguirre Ballesteros
Juan Carlos Santamaria Orjuela**

**Pontificia Universidad Javeriana
Facultad de ingeniería
Departamento de ingeniería de sistemas
Bogotá D.C.
2025**

TAD WBS

Descripción:

Modela la estructura de desglose de trabajo y permite gestionar el proyecto en términos de fases, entregables y actividades elementales.

Datos mínimos:

- `arbolGeneral`, tipo `ArbolGeneral<NodoArbol>`, será la estructura jerárquica de trabajo.

Comportamiento operaciones:

- `costoNodo()`, calcula el costo de un nodo
- `TiempoNodo()`, calcula el tiempo de un nodo
- `costoProyecto()`, calcula el costo total del proyecto
- `TiempoProyecto()`, calcula el tiempo total del proyecto
- `CargarArchivo()`, construye el WBS a partir de un archivo

TAD Arbol General

Descripción:

Modela la estructura jerárquica del proyecto en forma de árbol general. La raíz representa el proyecto, los nodos intermedios las fases y las hojas las actividades elementales.

Datos mínimos:

- `NodoGeneral<T>*` `raiz`, puntero a la raíz

Comportamiento operaciones:

- `ArbolGeneral()`, constructor por omisión
- `ArbolGeneral(T val)`, constructor con valor de la raíz
- `ArbolGeneral(NodoGeneral<T>*)`, constructor a partir de un nodo
- `ArbolGeneral(const ArbolGeneral<T> &)`, constructor copia
- `~ArbolGeneral()`, destructor
- `subArbolN()`, analiza la lista de subárboles asociados
- `esVacio()`, Analizadora que está vacía
- `obtenerRaiz()`, Analizadora para obtener Información en la raíz
- `fijarRaiz(T)`, Modificadora para cambiar la información de la raíz
- `operator=(const ArbolGeneral<T> & other)`, Sobrecarga operador '='
- `insertarNodo(const T &padre, const T &n)`, Inserta un hijo con información n en padre padre
- `eliminarNodo(const T &n)`, eliminar el nodo cuya información es n
- `altura()`, Calcular la altura del árbol
- `tamano()`, Calcular el peso y número de elementos en el árbol
- `orden()`, Calcular el orden y número máximo de subárboles presentes en un árbol

TAD NodoArbol

Descripción:

El nodo será el aquel que tendrá los datos de cada actividad, fase o entregable

Conjunto mínimo de datos :

- id,entero,identifica la actividad, fase o entregable.
- descripción,cadena de caracteres, describe la actividad, fase o entregable.
- costo,decimal, costo estimado en dinero.
- tiempo,entero, tiempo estimado de ejecución

Comportamiento operaciones:

- crear(), crea un nodo con sus datos específicos.
- obtenerId (), mostrar el dato del id.
- fijarId (), cambiar el contenido del id.
- obtenerDescripcion (), mostrar el dato del descripcion.
- fijarDescripcion (), cambiar el contenido del descripcion.
- obtenerCosto (), mostrar el dato del costo.
- fijarCosto (), cambiar el contenido del costo.
- obtenerTiempo (),mostrar el dato del tiempo.
- fijarTiempo (), cambiar el contenido del tiempo.

TAD Nodogeneral

Descripción:

Nodo de un árbol. Cada nodo almacena un dato genérico T (template) y dos enlaces: al primer hijo y al siguiente hermano.

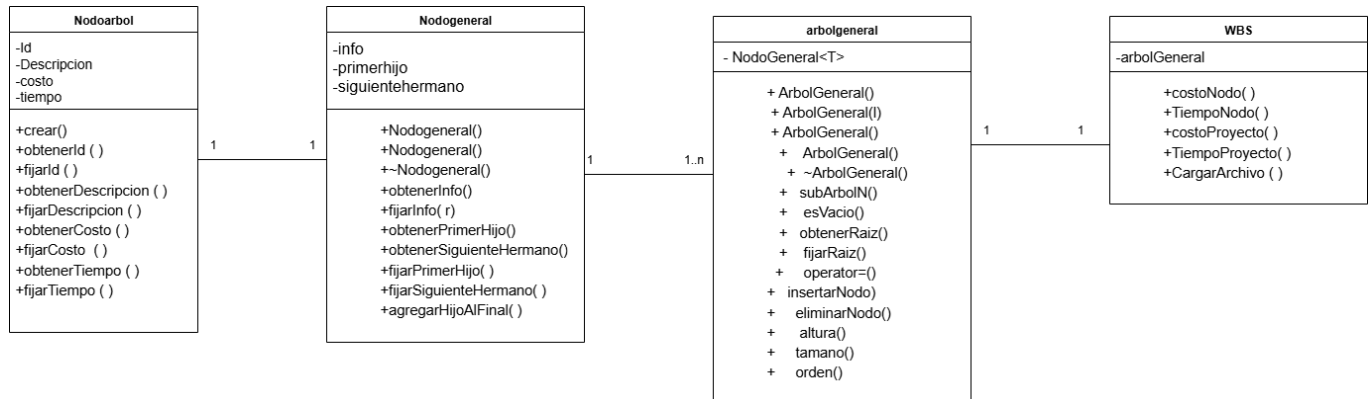
Conjunto mínimo de datos :

- info, tipo genérico T, dato almacenado en el nodo.
- primerHijo, puntero al nodo general, puntero al primer hijohijos.
- siguienteHermano, puntero al nodo general ,puntero al siguiente hermano en la lista.

Comportamiento operaciones:

- Nodogeneral(), constructor por omisión.
- Nodogeneral(const T& valor), constructor con valor inicial.
- ~Nodogeneral(), destructor por defecto.
- obtenerInfo(), mostrar el dato info.
- fijarInfo(const T& valor), cambiar el contenido de info.
- obtenerPrimerHijo(), mostrar puntero al primer hijo.

- obtenerSiguienteHermano(), mostrar puntero al siguiente hermano.
- fijarPrimerHijo(Nodogeneral<T>* h), cambiar el puntero primerHijo.
- fijarSiguienteHermano(Nodogeneral<T>*h), cambiar el puntero siguienteHermano.
- agregarHijoAlFinal(Nodogeneral<T>* h), inserta h al final de la lista de hijos del nodo.



Plan de pruebas

comando para ejecutar el programa:

g++ -std=c++11 main.cpp NodoArbol.cpp WBS.cpp -o programa.exe
 programa.exe

Archivo wbs.txt:

Se requiere crear el WBS a partir de un archivo de tipo texto en el que cada línea contiene, separados por un espacio, el id, la descripción del proyecto fase o actividad, el costo y el tiempo para las actividades elementales. Para el nombre del proyecto el id es 0 y para las fases o entregables intermedios el costo y el tiempo estarán en 0.

wbs.txt:

```

0 -1 Proyecto 0 0
1 0 Fase1 0 0
2 0 Fase2 0 0
3 1 Actividad1 600 6
4 1 Actividad2 400 4
5 2 Actividad3 500 5
  
```

- Fases = costo 0 en los datos del archivo.
- Cálculos globales = suman los hijos y dan valores correctos.
- Mostrar camino = imprime los datos del nodo pero no la suma de sus hijos.

Con este archivo realizaremos el plan de pruebas correspondiente al taller, la siguiente imagen tiene los 3 casos de pruebas que realizamos y entraremos más en detalles de cada uno:

```
C:\Users\JULIANA\Desktop\taller3>programa.exe
==== Proyecto WBS ====
costo total del proyecto: 1500
tiempo total del proyecto: 15 dias

=== Consultas por nodo ===
Ingrese id del nodo para para calcular costo y tiempo: 1
Costo del nodo 1: 1000
Tiempo del nodo 1: 10 dias

=== Camino hasta una actividad terminal ===
Ingrese id de la actividad terminal: 3
Camino desde la raiz hasta la actividad 3:
-> Proyecto (Costo: 0)
-> Fase1 (Costo: 0)
-> Actividad1 (Costo: 600)

C:\Users\JULIANA\Desktop\taller3>programa.exe
==== Proyecto WBS ====
costo total del proyecto: 1500
tiempo total del proyecto: 15 dias

=== Consultas por nodo ===
Ingrese id del nodo para para calcular costo y tiempo: 2
Costo del nodo 2: 500
Tiempo del nodo 2: 5 dias

=== Camino hasta una actividad terminal ===
Ingrese id de la actividad terminal: 1
Camino desde la raiz hasta la actividad 1:
-> Proyecto (Costo: 0)
-> Fase1 (Costo: 0)

C:\Users\JULIANA\Desktop\taller3>programa.exe
==== Proyecto WBS ====
costo total del proyecto: 1500
tiempo total del proyecto: 15 dias

=== Consultas por nodo ===
Ingrese id del nodo para para calcular costo y tiempo: 1
Costo del nodo 1: 1000
Tiempo del nodo 1: 10 dias

=== Camino hasta una actividad terminal ===
Ingrese id de la actividad terminal: 5
Camino desde la raiz hasta la actividad 5:
-> Proyecto (Costo: 0)
-> Fase2 (Costo: 0)
-> Actividad3 (Costo: 500)
```

Caso de Prueba 1

Entrada: idActividad = 3

Descripción esperada del camino:

Proyecto -> Fase1 ->Actividad1 (Costo: 600)

Salida obtenida en consola:

```
C:\Users\JULIANA\Desktop\taller3>programa.exe
==== Proyecto WBS ====
costo total del proyecto: 1500
tiempo total del proyecto: 15 dias

=== Consultas por nodo ===
Ingrese id del nodo para calcular costo y tiempo: 1
Costo del nodo 1: 1000
Tiempo del nodo 1: 10 dias

=== Camino hasta una actividad terminal ===
Ingrese id de la actividad terminal: 3
Camino desde la raíz hasta la actividad 3:
-> Proyecto (Costo: 0)
-> Fase1 (Costo: 0)
-> Actividad1 (Costo: 600)
```

Camino desde la raíz hasta la actividad 3:

- > Proyecto (Costo: 0)
- > Fase1 (Costo: 0)
- > Actividad1 (Costo: 600)

Caso de Prueba 2

Entrada: idActividad = 1

Descripción esperada del camino:

Proyecto -> Fase1 (nodo intermedio, no actividad terminal)

Salida obtenida en consola:

```
C:\Users\JULIANA\Desktop\taller3>programa.exe
==== Proyecto WBS ====
costo total del proyecto: 1500
tiempo total del proyecto: 15 dias

=== Consultas por nodo ===
Ingrese id del nodo para calcular costo y tiempo: 2
Costo del nodo 2: 500
Tiempo del nodo 2: 5 dias

=== Camino hasta una actividad terminal ===
Ingrese id de la actividad terminal: 1
Camino desde la raiz hasta la actividad 1:
-> Proyecto (Costo: 0)
-> Fase1 (Costo: 0)
```

Camino desde la raiz hasta la actividad 1:

-> Proyecto (Costo: 0)

-> Fase1 (Costo: 0)

Caso de Prueba 3

Entrada: idActividad = 5

Descripción esperada del camino:

Proyecto → Fase2 → Actividad3 (Costo: 500)

Salida obtenida en consola:

```
C:\Users\JULIANA\Desktop\taller3>programa.exe
==== Proyecto WBS ====
costo total del proyecto: 1500
tiempo total del proyecto: 15 dias

=== Consultas por nodo ===
Ingrese id del nodo para calcular costo y tiempo: 1
Costo del nodo 1: 1000
Tiempo del nodo 1: 10 dias

=== Camino hasta una actividad terminal ===
Ingrese id de la actividad terminal: 5
Camino desde la raiz hasta la actividad 5:
-> Proyecto (Costo: 0)
-> Fase2 (Costo: 0)
-> Actividad3 (Costo: 500)
```

Camino desde la raiz hasta la actividad 5:

-> Proyecto (Costo: 0)

-> Fase2 (Costo: 0)

-> Actividad3 (Costo: 500)

Conclusiones de las pruebas realizadas:

Jerarquía mediante el WBS

- El WBS se modela como un Árbol General de NodoArbol, cumpliendo con su propósito: organizar el taller en fases y actividades elementales.
- La función CargarArchivo() construye la estructura a partir del archivo wbs.txt.
- costoNodo() y tiempoNodo() permiten calcular los valores de un nodo así acumulando los valores de sus hijos cuando es intermedio.

Organización de datos con ArbolGeneral

- El árbol tiene una raíz , nodos intermedios que son las fases y hojas son las actividades.
- Se valida los métodos de inserción y búsqueda permiten navegar la jerarquía y ubicar cualquier nodo mediante su id.
- La operación insertarNodo() funciona asociando correctamente a cada hijo con su padre en el árbol.

Encapsulación de información con NodoArbol

- Cada nodo almacena sus atributos: id, descripción, costo, tiempo.
- Para el caso de fases, el costo y tiempo se inicializan en 0 según el taller.
- Gracias a los métodos obtener() y fijar(), se accede y se modifica la información, cumpliendo con el comportamiento esperado.

Enlaces con Nodogeneral

- El árbol se construye gracias a la relación primer hijo -> siguiente hermano que permite manejar varios hijos de un mismo nodo.
- Se comprobó que la función agregarHijoAlFinal() conserva el orden de inserción y mantiene la estructura jerárquica.

Funcionalidad probada: el camino desde la raíz

- La función mostrarCamino(id) recorre los enlaces de Nodogeneral y acumula en un vector los nodos visitados así mostrando la secuencia desde.
- En las pruebas que se realizaron los caminos que se muestran en consola corresponden a la jerarquía en el archivo wbs.txt.
- Las fases muestran costo 0, la estructura del árbol permite que los cálculos acumulados muestran los valores correctos.