**Pontificia Universidad Catolica Madre y Maestra**

**Campus-Santiago**

****

**Facultad de Ciencias de la Ingeniería**

**Programación II**

**Reporte**

Arboles Binarios y MESH

**Presentado Por:**

Jean Carlos Marte Gomez

Robert Saint-Hilaire

Felix Alejandro Guzman

Emmanuel Jáquez Castillo

**A:**

Ing. José Luis Alonso Ochoa

**Matricula:**

2010-2037

2010-2301

2014-0565

2007-0410

**En Fecha:**

19/09/2016

**Introducción**

En el siguiente reporte logramos hacer una investigación sobres los árboles binarios que son un *tipo de estructura de datos útil a la hora que se tengan que tomar decisiones en dos sentidos en cada punto de un proceso. Como también sobre algunas definiciones y que componen al árbol binario como* ***raíz****,****hoja*** *y* ***subárbol****, entre otras, que explicaremos más adelante.*

*Más adelante veremos las clasificaciones de los árboles binarios que son 4 tipos:* **Árbol Binario Distinto, Árbol Binario Similar,Árbol Binario Equivalente,Árbol Binario Completo.** Para finalizar el tema de arboles binarios, veremos como recorrer un árbol y cuales son los tipos de recorridos y búsquedas del mismo**.**

**Árboles Binarios**

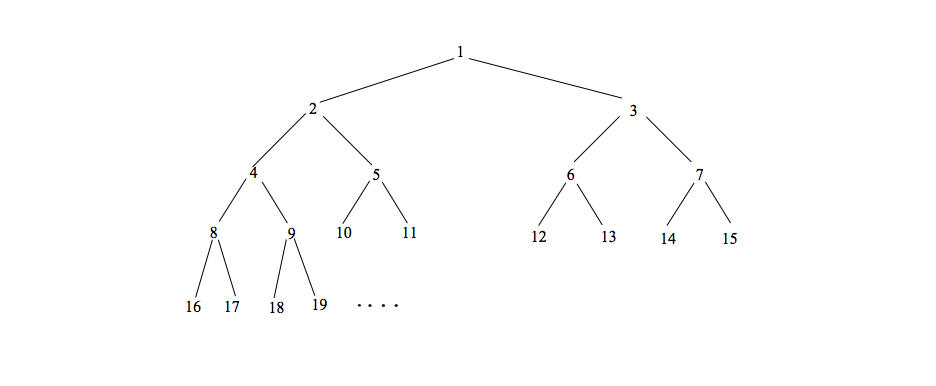
Hay un tipo especial de árbol muy usado en computación, denominado árbol binario, en el que cada nodo pueden colgar,a lo más, dos subárboles. Estos se denominan subarbol derecho y subarbol izquierdo, y tambien son árboles binarios, es decir un árbol nulo o un árbol cuyos nodos tienen a lo sumo dos hijos.

Un árbol binario es una estructura de datos útil a la hora que de se deben tomar decisiones en dos sentidos en cada punto de un proceso.

La forma usual de representar los árboles supone el uso de punteros, aunque también se puede hacer con vectores. En un árbol binario cada nodo está constituido por una parte de datos (información) y dos punteros que indican la posición de sus hijos.

**Algunas definiciones:**

* **Raíz:** es el único nodo que no tiene ancestros propios.
* **Hoja:** es el único nodo sin descendientes propios.
* **Subárbol:** un árbol es un nodo junto con todos sus descendientes.
* **Peso de un árbol:** es el número de nodos del árbol sin contar la raíz
* **Peso de un nodo:** en un árbol es la longitud del camino más largo del nodo a una hoja.
* **Padre:** X es padre de Y si y sólo si el nodo X apunta a Y. También se dice que X es antecesor de Y.
* **Hijo:** X es hijo de Y, si y sólo si el nodo X es apuntado por Y. También se dice que X es descendiente directo de Y.
* **Hermano:** Dos nodos serán hermanos si son descendientes directos de un mismo nodo.
* **Nivel:** Es el número de arcos que deben ser recorridos para llegar a un determinado nodo. Por definición la raíz tiene nivel 1.
* A**ltura**: es el máximo número de niveles de todos los árboles..
* **Profundidad de un nodo** es la longitud del camino único de la raíz al nodo.
* **Profundidad de un árbol** es la profundidad de la hoja más profunda.
* **Grado:** Es el número de descendientes directos de un determinado nodo.
* **Grado de un árbol:** Es el máximo grado de todos los nodos del árbol.
* **Longitud de camino:** Es el número de arcos que deben ser recorridos para llegar desde la raíz al nodo X. Por definición la raíz tiene longitud de camino 1, y sus descendientes directos longitud de camino 2 y así sucesivamente.



**Clasificaciones de los árboles binarios**:

Existen cuatro tipos de árbol binario:.

* Árbol Binario Distinto.
* Árbol Binario Similares.
* Árbol Binario Equivalentes.
* Árbol Binario Completos.

A continuación una breve descripción de cada uno de los tipos de árboles binarios.

**Árbol Binario Distinto:**

Se dice que dos árboles binarios son distintos cuando sus estructuras son diferentes.

**Árbol Binario Similar:**

Dos árboles binarios son similares cuando sus estructuras son idénticas, pero la información que contienen sus nodos es diferente.

**Árbol Binario Equivalente:**

Son aquellos árboles que son similares y que además los nodos contienen la misma información.

**Árbol Binario Completo:**

Son aquellos árboles en los que todos sus nodos excepto los del último nivel, tiene dos hijos; el subárbol izquierdo y el subárbol derecho.

**Recorrido de un Árbol Binario**

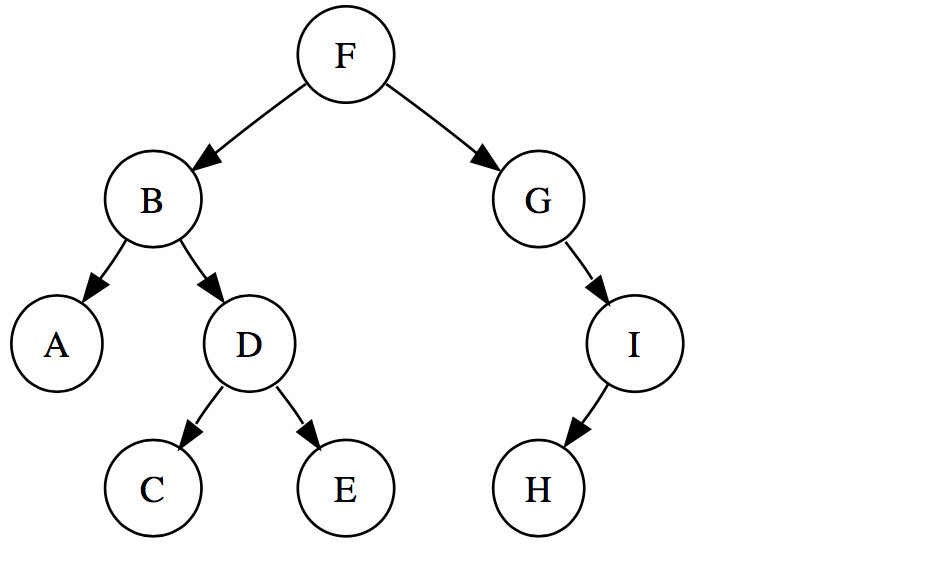
Hay dos formas básicas de recorrer un árbol:

* Recorrido en Amplitud.
* Recorrido en Profundidad.

Estos métodos nos permiten almacenar y procesar un conjunto ordenado, con bastante facilidad, sin que tengamos que proceder a largas operaciones de readaptación.

**Recorrido en Amplitud:**

Los árboles también pueden ser recorridos en orden por nivel (de nivel en nivel), donde visitamos cada nodo en un nivel antes de ir a un nivel inferior. Seria la raiz F el nivel 1, B-G nivel 2, A-D-I nivel 3 y por último C-E-H nivel 4.



**Recorrido en Profundidad:**

Recorrer un árbol supone acceder a sus elementos de forma sistemática lo que supone llevar a cabo tres actividades:

1. Visitar el nodo raíz
2. Recorrer el subárbol izquierdo
3. Recorrer el subárbol derecho

Estas tres acciones repartidas en diferentes órdenes proporcionan diferentes recorridos del árbol, llamados: **pre-orden, post-orden, in-orden.** Su nombre refleja el momento en que se visita el nodo raíz. En el “in-orden” el raíz está en el medio del recorrido, en el “pre-orden”, el raíz está en el primero y en el “post-orden”, el raíz está el último:

* **Preorden**: (raíz, izquierdo, derecho). Para recorrer un árbol binario no vacío en preorden, hay que realizar las siguientes operaciones recursivamente en cada nodo, comenzando con el nodo de raíz:

1. Visite la raíz
2. Atraviese el sub-árbol izquierdo
3. Atraviese el sub-árbol derecho

* **Inorden**: (izquierdo, raíz, derecho). Para recorrer un árbol binario no vacío en **inorden** (simétrico), hay que realizar las siguientes operaciones recursivamente en cada nodo:

1. Atraviese el sub-árbol izquierdo
2. Visite la raíz
3. Atraviese el sub-árbol derecho

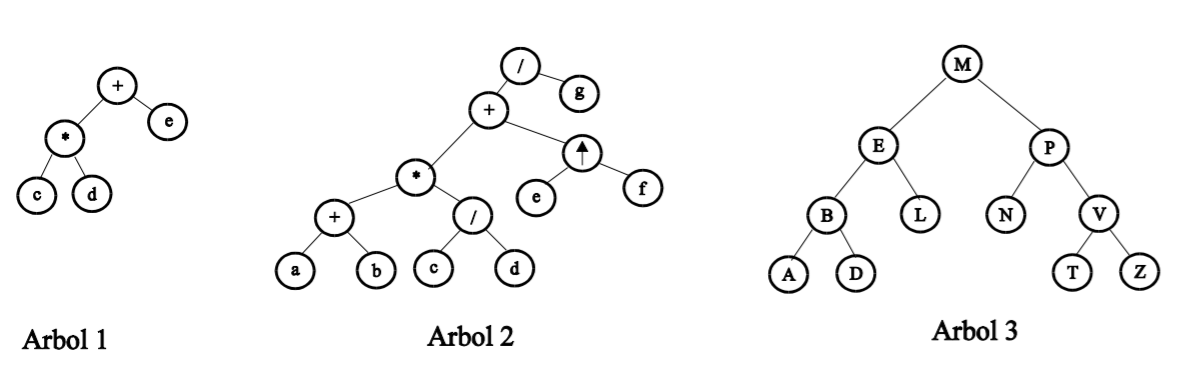
* **Postorden**: (izquierdo, derecho, raíz). Para recorrer un árbol binario no vacío en postorden, hay que realizar las siguientes operaciones recursivamente en cada nodo:

1. Atraviese el sub-árbol izquierdo
2. Atraviese el sub-árbol derecho
3. Visite la raíz

En general, la diferencia entre preorden, inorden y postorden es cuándo se recorre la raíz. En los tres, se recorre primero el subárbol izquierdo y luego el derecho.

* En preorden, la raíz se recorre antes que los recorridos de los subárboles izquierdo y derecho
* En inorden, la raíz se recorre entre los recorridos de los árboles izquierdo y derecho, y
* En postorden, la raíz se recorre después de los recorridos por el subárbol izquierdo y el derecho

Preorden (antes), inorden (en medio), postorden (después).

****

------------------------------------------------------------------------------------------------------------------------

**Árbol 1 Pre-orden +\*cde**

**In-orden c\*d+e**

**Post-orden cd\*e+**

------------------------------------------------------------------------------------------------------------------------

**Árbol 2 Pre-orden /+\*+ab/cd |efg**

**In-orden a+b\*c/d+e|f/g**

**Post-orden ab+cd/\*ef|+g**

------------------------------------------------------------------------------------------------------------------------

**Arbol 3 Pre-orden MEBADLPNVTZ In-orden ABDELMNPTVZ**

**Post-orden ADBLENTPZVPM**

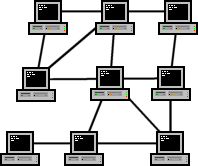
-----------------------------------------------------------------------------------------------------------------

**Árbol Binario de Búsqueda**

La búsqueda en arboles binarios es un método de búsqueda simple, dinámico y eficiente considerado como uno de los fundamentales en la ciencia de la computación. Se utilizan frecuentemente para representar conjuntos de datos cuyos elementos se identifican por una clave única. Si el árbol está organizado de tal manera que la clave de cada nodo es mayor que todas las claves su subárbol izquierdo, y menor que todas las claves del subárbol derecho se dice que este árbol es un árbol de búsqueda. Llamaremos árbol binario de búsqueda a un árbol construido de acuerdo con el procedimiento siguiente:

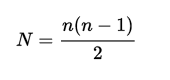
1. El primer elemento se utiliza para crear el nodo raíz.
2. Los valores del árbol deben ser tales que pueda existir un orden (entero, real, lógico o carácter e incluso definido por el usuario si se tiene un orden total.
3. En cualquier nodo, todos los valores del subárbol izquierdo del nodo son menores o iguales que el valor del nodo. De modo similar todos los valores del subárbol derecho deben de ser mayores que los valores del nodo.

**Mesh (Malla)**

La topología de red malla es una topología de red en la que cada nodo está conectado a todos los nodos. De esta manera es posible llevar los mensajes de un nodo a otro por distintos caminos. Si la red de malla está completamente conectada, no puede existir absolutamente ninguna interrupción en las comunicaciones. Cada servidor tiene sus propias conexiones con todos los demás servidores.

Esta topología, a diferencia de otras más usuales como la topología en árbol y la topología en estrella, no requiere de un nodo central, con lo que se reduce el riesgo de fallos, y por ende el mantenimiento periódico (un error en un nodo, sea importante o no, no implica la caída de toda la red).

Muchas veces la topología malla se va a unir a otra topología para formar una topología híbrida. En las redes de malla las computadoras están conectadas entre sí, de modo que no existen un una privilegio de un computador sobres otros, refiriéndose a la concentración del tráfico de comunicación.

El número de enlaces que existen en una malla completa, i.e, una topología en malla en la que existe un enlace punto-a-punto entre todos los terminales, viene dado por la siguiente fórmula:

**Ventajas de Malla**

* Si la red de malla está completamente conectada, puede existir absolutamente ninguna interrupción en las comunicaciones.No requiere de un servidor o nodo central, con lo que se reduce el mantenimiento.
* Si un enlace falla, no inhabilita todo el sistema.
* Cuando un mensaje viaja a través de una línea dedicada, solamente lo ve el receptor adecuado.
* Las líneas de seguridad y privacidad evitan el acceso de otros usuarios.

**Desventajas de Malla**

* La desventaja física principal es que sólo funciona con una pequeña cantidad de nodos, ya que de lo contrario la cantidad de medios necesarios para los enlaces, y la cantidad de conexiones con los enlaces se torna abrumadora.
* El costo de la red puede aumentar en los casos en los que se implemente de forma alámbrica, la topología de red y las características de la misma implican el uso de más recursos.
* La instalación y configuración es más difícil, porque cada dispositivo debe estar conectado a otro.
* Baja eficiencia de las conexiones o enlaces, debido a la existencia de enlaces redundantes.

**Conclusión**

En este reporte se pudo aprender y conocer sobre los tipos de árboles binarios, cómo diferenciar los tipos de árboles binarios, sus clasificaciones y definiciones. Cada clasificación cumplen con algunas requisitos que hacen que los diferencien entre sí. Se pudo aprender sobre el recorrido de los árboles para saber su grado, su raíz y los subárbol que tienen si ese es el caso. También aprendimos sobre los tipos de recorridos de un árbol binario, como los son el recorrido en amplitud y el recorrido en profundidad que este último tiene distintos orden de recorrido: **Preorden, Inorden y Postorden**.Conocimos la definición de cada uno, usos y ejemplos de los mismos.

También se pudo aprender sobre el tipo de topología Mesh (Malla), que se caracteriza porque utiliza enlaces dedicados que garantiza que cada conexión sólo debe transportar la carga de datos propia de los dispositivos conectados. Otro aspecto importante de esta tipología es que si un enlace falla no inhabilita todo el sistema.

**Bibliografía**

[**http://delta.cs.cinvestav.mx/~adiaz/anadis/BinTree.pdf**](http://delta.cs.cinvestav.mx/~adiaz/anadis/BinTree.pdf)

[**http://www.utm.mx/~rruiz/cursos/ED/material/ABB.pdf**](http://www.utm.mx/~rruiz/cursos/ED/material/ABB.pdf)

[**http://robotica.uv.es/pub/Libro/PDFs/CAPI5.pdf**](http://robotica.uv.es/pub/Libro/PDFs/CAPI5.pdf)

<https://www.wikiwand.com/es/Red_en_malla>