



Manual de usuario: Proyecto Algoritmos

Integrantes: Manuel Cruz Garrote y Mariana Sandoval Garzón

Enlace Repositorio:

[https://github.com/ManuelCrzUR/Proyecto Algoritmos y Estructura de Datos.git](https://github.com/ManuelCrzUR/Proyecto_Algoritmos_y_Estructura_de_Datos.git)

Este documento tiene como objetivo ser una guía de uso para el usuario del API de Árboles Binarios, Mapas y Tablas Hash. De acuerdo con esto, se explicará la sintaxis, la navegación y acceso a las librerías.

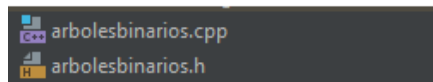
Tabla de contenidos:

1. Breve introducción a los archivos
2. Uso de la librería de Árboles Binarios
3. Uso de la librería de Mapas
4. Uso de la Librería de Tablas Hash

1. Breve introducción a los archivos

Para cada librería se tienen dos archivos uno de extensión “.cpp” y otro de extensión “.h”. Ambos contienen las mismas clases y funciones, sin embargo, el “.h” es el archivo base para la librería. El archivo de “.cpp” sirve como guía de estudio para evaluar a detalle cada función y modificarlas al gusto. En la *Imagen 1* se muestra la diferenciación de estos archivos. Adicionalmente, se incluye un archivo main.cpp donde se importarán las librerías y se ejecutaran las funciones, en las siguientes secciones se adentrará en este proceso.

Imagen 1. Ejemplo de la diferenciación de los archivos



2. Uso de la librería de Árboles Binarios

Para incluir y usar esta librería, se usa escribe lo siguiente al inicio del main.cpp. Luego, se irán implementando las funciones en la función int main().

```
#include "arbolesbinarios.h"
```

Para declarar un apuntador para un NodoAB que incluya la raíz, la sintaxis es:

```
NodoAB *nombredelaraiz
```

La siguiente tabla, muestra la sintaxis de las demás funciones:

Funciones	Sintaxis
Función: NodoAB *insertar	nombredelaraiz=insertar(nombredelaraiz, val)
Función: void PreOrden	PreOrden(nombredelaraiz)
Función: void InOrden	InOrden(nombredelaraiz)
Función: void PosOrden	PosOrden(nombredelaraiz)
Función: int buscar	buscar(nombredelaraiz, num)
Función: int Sumar	Sumar(nombredelaraiz)
Función: int contar	contar(nombredelaraiz, num)
Función: NodoAB *encontrar_nodo	encontrar_nodo(nombredelaraiz, val)
Función: int camino	camino(nombredelaraiz, num1, num2)
Función: int peso	peso(nombredelaraiz)
Función: int hojas	hojas(nombredelaraiz)

3. Uso de la librería de Mapas

Para incluir y usar esta librería, se usa escribe lo siguiente al inicio del main.cpp. Luego, se irán implementando las funciones en la función int main().

```
#include "mapas.h"
```

Para declarar un objeto Mapa, la sintaxis es:

```
//Mapa nombre= Mapa()
```

La siguiente tabla, muestra la sintaxis de las demás funciones:

Funciones	Descripción Detallada
Función: void Insertar	nombre.Insertar("nombreclave", valor);
Función: int ObtenerValor	nombre.ObtenerValor("nombreclave");

Función: bool ExisteClave	<code>nombre.ExisteClave("nombreclave");</code>
Función: void EliminarElemento	<code>nombre.EliminarElemento("sanny");</code> .
Función: bool EstaVacio	<code>nombre.EstaVacio();</code>
Función: int ObtenerTamaño	<code>nombre.ObtenerTamaño();</code>
Función: void MostrarElemento	<code>nombre.MostrarElementos();</code>

4. Uso de la librería de Tablas Hash

Para incluir y usar esta librería, se usa escribe lo siguiente al inicio del main.cpp. Luego, se irán implementando las funciones en la función int main().

```
#include "hash.h"
```

Para declarar un objeto HashMap, la sintaxis es:

```
HashMap nombre;
```

La siguiente tabla, muestra la sintaxis de las demás funciones:

Funciones	Descripción Detallada
Función: void insertar	<code>nombre.insertar(valorllave, "nombrevalor");</code>
Función: void remover	<code>nombre.remover(valorllave)</code>
Función: string obtenerValor	<code>nombre.obtenerValor(valorllave);</code>
-Función: int obtenerTamaño	<code>nombre.estaVacio(valorllave)</code>
Función: bool estaVacio	<code>nombre.obtenerTamaño()</code>
Función: void mostrarMapaHash()	<code>nombre.mostrarMapaHash();</code>