

ABB vs Negro & Rojo



MARCO TEÓRICO

- Un ABB tiene la característica de que todos los subelementos de un nodo a su izquierda están los menores y a su derecha los mayores almacenados. Por otro lado, la complejidad de los árboles negro y rojo es mayor, ya que es un árbol de búsqueda binaria de datos con asignación de colores.
- Poseen las siguientes características: Los nodos tienen un color propio ya sea rojo o negro. Aparte de todo lo que cumple un árbol BB se debe cumplir lo siguiente:

Nodo rojo o negro

Raíz negra

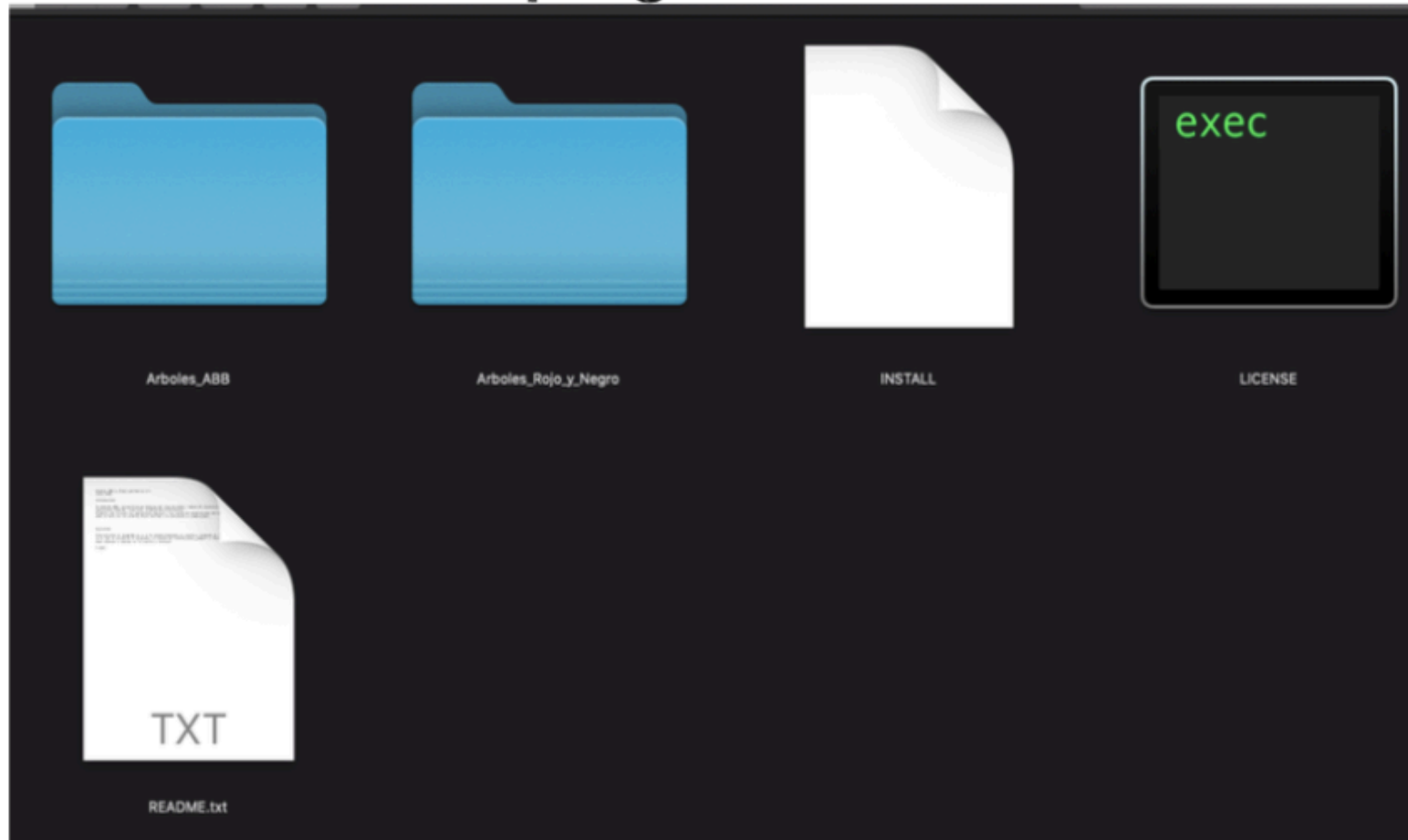
Hojas negras

Hijos de todo nodo rojo son negros

Cada camino tiene la misma cantidad de nodos negros y esto se denomina altura del árbol



Funcionamiento del programa



ABB



ArbolABB.cpp



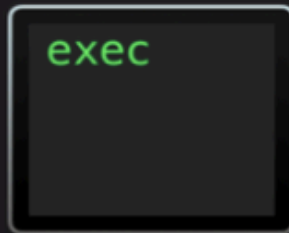
ArbolABB.h



doxygen.pdf



main.cpp



makefile



NodaArbol.cpp



NodaArbol.h



NEGRO Y ROJO



arbol_rojo_negro.h



doxygen.pdf



header.h



main.cpp



makefile



Readme

Arboles ABB vs Black and Red en C++
Junio 2019

Introducción:

En árboles ABB: características básicas del tipo de árbol, reglas de construcción, operaciones básicas: inserción, eliminación, localización. Ejemplos que aclaran las operaciones básicas y las reglas de construcción. De este mismo modo se hará con los árboles Black and Red y se analizará su complejidad.

Ejecución:

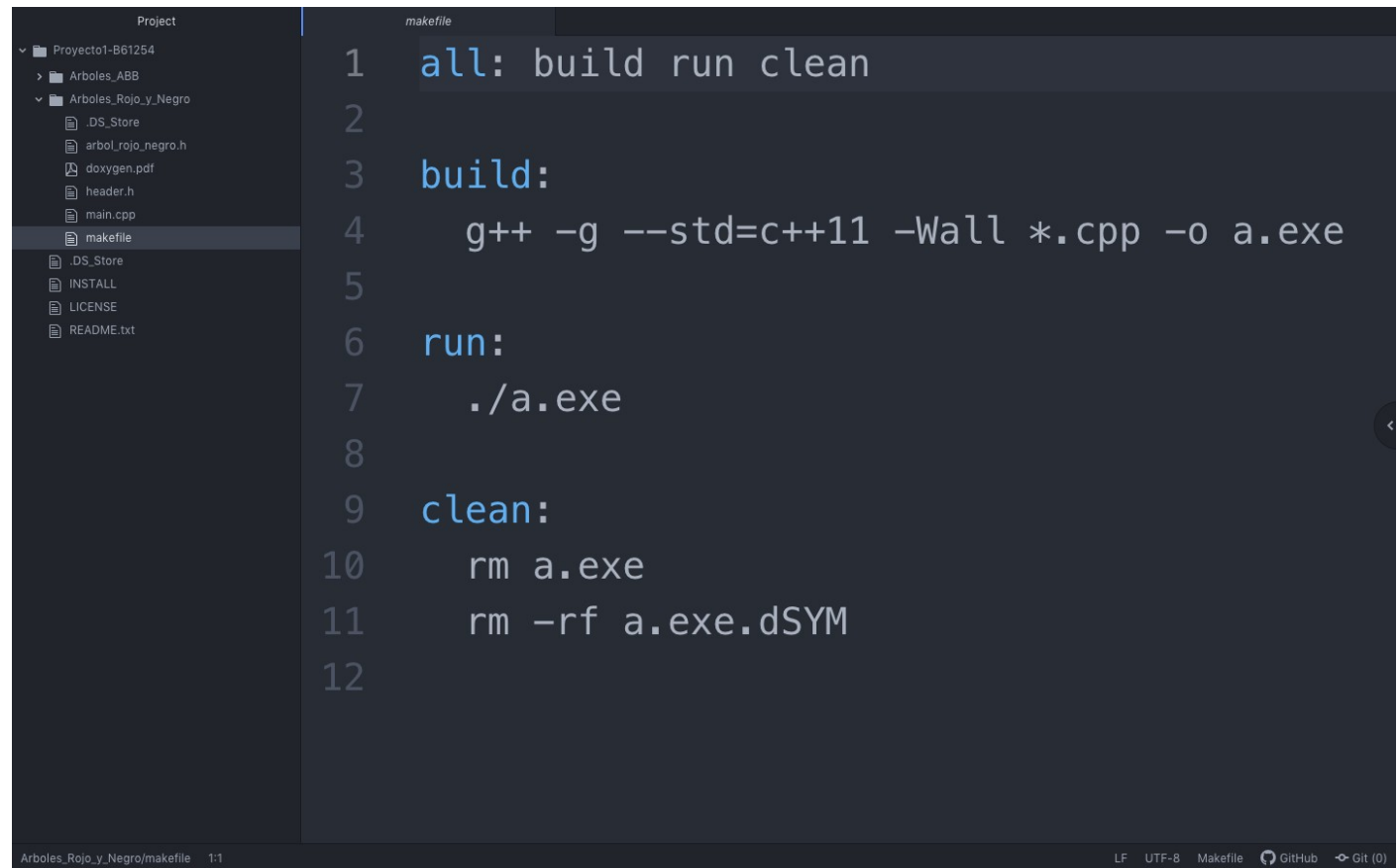
Para ejecutar el programa es ir a la carpeta mediante su consola o terminal al folder en el que se encuentra el programa, la carpeta de "Arboles_Rojo_y_Negro" o "Arboles_ABB" Debe ingresar y digitar en la consola o terminal:

```
$ make
```



Ejecutarlo

Makefile



The image shows a code editor window with a dark theme. On the left is a file explorer showing a project structure. The main area displays a Makefile with three targets: 'all', 'build', and 'clean'. The 'all' target depends on 'build', 'run', and 'clean'. The 'build' target uses 'g++' with various flags to compile *.cpp files into a.exe. The 'run' target executes ./a.exe. The 'clean' target removes a.exe and a.exe.dSYM. The bottom status bar indicates the file is 'Arboles_Rojo_y_Negro/makefile' and shows encoding (UTF-8) and other details.

```
1 all: build run clean
2
3 build:
4     g++ -g --std=c++11 -Wall *.cpp -o a.exe
5
6 run:
7     ./a.exe
8
9 clean:
10    rm a.exe
11    rm -rf a.exe.dSYM
12
```

Arboles_Rojo_y_Negro/makefile 1:1 LF UTF-8 Makefile GitHub Git (0)

Experimentos realizados

```
*****
```

```
Tester de arboles ABB #1
```

```
*****
```

```
-----  
Agregando valores  
-----
```

```
Agregar 35  
Agregar 12  
Agregar 50  
Agregar 8  
Agregar 17  
Agregar 32  
Agregar 53  
Agregar 3  
Agregar 10  
Agregar 29  
Agregar 57  
-----
```

Prueba de espacio en los árboles

Realizando un recorrido a lo ancho del primer árbol
35 12 50 8 17 53 3 10 32 57 29

Realizando un recorrido a lo ancho del segundo árbol
53 12 57 8 50 59412480 3 10 17 32 29
Como resultado obtenemos que:
No esta lleno

Como resultado obtenemos que:
No esta completo

Eliminado hojas

Borrando hoja 29 1

Realizando un recorrido a lo ancho 35 12 50 8 17 53 3 10 32 57

Borrando nodo con 1 hijo 53 1

Realizando un recorrido a lo ancho 35 12 50 8 17 57 3 10 32

Borrando nodo con dos hijos 121

Realizando un recorrido a lo ancho 35 8 50 3 17 57 10 32

Tester de arboles ABB #2

El arbol (29,11,28,18,31): No esta lleno

El arbol (29,11,28,18,31,8): No esta lleno

El arbol (29,11,28,18,31,8) No esta completo

El arbol (29,11,28,18,31,8,9,5,10,12) Como resultado obtenemos que:

No esta completo



////////////////////////////////////

Vamos a realizar las siguientes operaciones:

////////////////////////////////////

3 10 8 17 32 35 50 57

Borrando hoja 32 1

3 10 8 17 35 50 57

Borrando hoja 29 0

3 10 8 17 35 50 57

Borrando hoja 28 0

3 10 8 17 35 50 57

Borrando hoja 7 0

3 10 8 17 35 50 57

Borrando hoja 12 0

3 10 8 17 35 50 57

Borrando hoja 90 0

3 10 8 17 35 50 57

Borrando hoja 17 1

3 10 8 35 50 57

Borrando hoja 9 0

3 10 8 35 50 57

Borrando hoja 11 0

3 10 8 35 50 57

rm a.exe

rm -rf a.exe.dSYM

Belindas-MacBook-Air:Arboles_ABB belindabrown\$ █



```
Belindas-MacBook-Air:Arboles_Rojo_y_Negro belindabrown$ make
g++ -g --std=c++11 -Wall *.cpp -o a.exe
./a.exe
```

```
*****
```

Árbol rojo y negro

```
*****
```

1. Agregar elemento al árbol
2. Buscar un elemento
3. PRE-ORDER
4. POST-ORDER
5. Eliminar un elemento del árbol
6. Salir

```
*****
```

Escoja una opción ➡➡➡➡ 1

Va a ser insertado ... 45

Se agregó el valor

➡➡➡➡Escoja una opción ➡➡➡➡ 1

Va a ser insertado ... 23

Se agregó el valor

»»»»Escoja una opción »»»» 1

Va a ser insertado ... 45 67

Se agregó el valor

»»»»Escoja una opción »»»» Favor digite una opción dentro del menú

»»»»Escoja una opción »»»» 1

Va a ser insertado ... 67

Se agregó el valor

»»»»Escoja una opción »»»» 1

Va a ser insertado ... 2

Se agregó el valor



```

****Escoja una opción **** 3
Pre-Order **** 45[Negro]**** 3[Rojo]**** 1[Negro]**** 1[Rojo]**** 2[Rojo]**** 23[Negro]**** 4[Rojo]**** 67[Rojo]**** 45[Rojo]**** 78[Rojo]

****Escoja una opción **** 4
Post-Order **** 1[Rojo]**** 2[Rojo]**** 1[Negro]**** 4[Rojo]**** 23[Negro]**** 3[Rojo]**** 45[Rojo]**** 78[Rojo]**** 67[Rojo]**** 45[Negro]

****Escoja una opción **** 2
****Escoja una opción **** 3
Pre-Order **** 45[Negro]**** 3[Rojo]**** 2[Negro]**** 1[Rojo]**** 23[Negro]**** 4[Rojo]**** 67[Rojo]**** 45[Rojo]**** 78[Rojo]

****Escoja una opción **** 4
Post-Order **** 1[Rojo]**** 2[Negro]**** 4[Rojo]**** 23[Negro]**** 3[Rojo]**** 45[Rojo]**** 78[Rojo]**** 67[Rojo]**** 45[Negro]

****Escoja una opción **** 6
Salir ****

```



CONCLUSIONES

Insertar

ABB

Parsed in 0.003 seconds at 38.20 KB/s

Statistics

Language used: cpp		
Statistic	Your source	Average for cpp
Characters in source:	110 (3307 in highlighted result)	2508.566
Total time highlighting:	0.004 seconds	0.109 seconds
Characters per second:	28720.956 chars per second	0.000/82090.532/1816989.643 chars per second

Black & Red

4.5. ,

Parsed in 0.006 seconds at 263.65 KB/s

Statistics

Language used: cpp		
Statistic	Your source	Average for cpp
Characters in source:	1726 (12515 in highlighted result)	2508.533
Total time highlighting:	0.007 seconds	0.109 seconds
Characters per second:	238364.516 chars per second	0.000/82097.107/1816989.643 chars per second

Eliminar

ABB

Parsed in 0.011 seconds at 337.18 KB/s

Statistics

Language used: cpp		
Statistic	Your source	Average for cpp
Characters in source:	3896 (28177 in highlighted result)	2508.592
Total time highlighting:	0.012 seconds	0.109 seconds
Characters per second:	316594.176 chars per second	0.000/82106.972/1816989.643 chars per second

Black & Red

Language used: cpp		
Statistic	Your source	Average for cpp
Characters in source:	6221 (37596 in highlighted result)	2508.748
Total time highlighting:	0.018 seconds	0.109 seconds
Characters per second:	347250.704 chars per second	0.000/82118.127/1816989.643 chars per second



Implementación

ABB

Parsed in 0.020 seconds at 342.60 KB/s

Statistics

Language used: cpp		
Statistic	Your source	Average for cpp
Characters in source:	6994 (48134 in highlighted result)	2508.936
Total time highlighting:	0.021 seconds	0.109 seconds
Characters per second:	327315.111 chars per second	0.000/82128.441/1816989.643 chars per second

Black & Red

Parsed in 0.005 seconds at 232.39 KB/s

Statistics

Language used: cpp		
Statistic	Your source	Average for cpp
Characters in source:	1295 (10551 in highlighted result)	2508.885
Total time highlighting:	0.006 seconds	0.109 seconds
Characters per second:	206997.854 chars per second	0.000/82133.694/1816989.643 chars per second

ABB

```
793.     cout<<"Como resultado obtenemos que:\nSi son semejantes\n"<<endl;
794.     if (elIsoformo->compara_cant_elem(elSemejante) == false)
795.         cout<<"Como resultado obtenemos que:\nNo son semejantes\n"<<endl;
796.
797.     cout<<"\n////////////////////////////////////////\n"<<endl;
798.     cout<<"Vamos a realizar las siguientes operaciones:\n ";
799.     cout<<"\n////////////////////////////////////////\n"<<endl;
800.
801.     cout<<elArbol->ver_arbol(1)<<endl;
802.     cout<<"Borrando hoja 32 \n"<<elArbol->eliminar_nodo(32)<<endl;
803.     cout<<elArbol->ver_arbol(1)<<endl;
804.     cout<<"Borrando hoja 29 \n"<<elArbol->eliminar_nodo(29)<<endl;
805.     cout<<elArbol->ver_arbol(1)<<endl;
806.     cout<<"Borrando hoja 28 \n"<<elArbol->eliminar_nodo(28)<<endl;
807.     cout<<elArbol->ver_arbol(1)<<endl;
808.     cout<<"Borrando hoja 7 \n"<<elArbol->eliminar_nodo(7)<<endl;
809.     cout<<elArbol->ver_arbol(1)<<endl;
810.     cout<<"Borrando hoja 12 \n"<<elArbol->eliminar_nodo(12)<<endl;
811.     cout<<elArbol->ver_arbol(1)<<endl;
812.     cout<<"Borrando hoja 90 \n"<<elArbol->eliminar_nodo(90)<<endl;
813.     cout<<elArbol->ver_arbol(1)<<endl;
814.     cout<<"Borrando hoja 17 \n"<<elArbol->eliminar_nodo(17)<<endl;
815.     cout<<elArbol->ver_arbol(1)<<endl;
816.     cout<<"Borrando hoja 9 \n"<<elArbol->eliminar_nodo(9)<<endl;
817.     cout<<elArbol->ver_arbol(1)<<endl;
818.     cout<<"Borrando hoja 11 \n"<<elArbol->eliminar_nodo(11)<<endl;
819.     cout<<elArbol->ver_arbol(1)<<endl;
820.
821.
822.     return 0;
823. }
824.
```

Parsed in 0.063 seconds at 377.45 KB/s

Statistics

Language used: cpp		
Statistic	Your source	Average for cpp
Characters in source:	24446 (169090 in highlighted result)	2508.667
Total time highlighting:	0.066 seconds	0.109 seconds
Characters per second:	368101.568 chars per second	0.000/82092.778/1816989.643 chars per second

[Back to enter source](#)

NEGRO Y ROJO

```
360. while(info != 0){
361.     switch (info){
362.         case 1: cout << "\nVa a ser insertado ... ";
363.                 cin >> input; demo.Agregar_nodo_arbol(input);
364.         break;
365.         case 2: cout << "\nVa a ser buscado >>>>>>";
366.                 cin >> input;
367.         if(demo.TreeSearch(input)) { cout << "Se encontró el valor >>>>>>\n"; }
368.         else { cout << "No se encontró el valor >>>>>>\n"; }
369.         break;
370.         case 3: cout << "Pre-Order >>>>>> ";
371.                 demo.Pre_Ordenamiento_forma_lateral(demo.Obtener_Base());
372.                 cout << endl;
373.         break;
374.         case 4: cout << "Post-Order >>>>>> ";
375.                 demo.Post_Ordenamiento_forma_lateral(demo.Obtener_Base());
376.                 cout << endl;
377.         break;
378.         case 5: cout << "\n >>>>>>Eliminar el valor >>>>>> ";
379.                 cin >> input;
380.                 demo.eliminar(input);
381.         break;
382.     default: cout << "Favor digite una opción dentro del menú \n";
383.             }
384.     cout << "\n >>>>>>Escoja una opción >>>>>> ";
385.     cin >> info;
386. }
387. cout << "\nSalir >>>>>> \n ";
388. return 0;
389. }
390.
391.
392.
```

Parsed in 0.040 seconds at 387.89 KB/s

Statistics

Language used: cpp		
Statistic	Your source	Average for cpp
Characters in source:	15691 (93655 in highlighted result)	2507.744
Total time highlighting:	0.042 seconds	0.109 seconds
Characters per second:	376219.469 chars per second	0.000/82080.744/1816989.643 chars per second

[Back to enter source](#)

Referencias bibliográficas

- [1] Kroah-Hartman G Corbet. J, Rubini. A. Linux Coding. O'Reilly books, 1998.
- [2] Computer Science Labs. Tecnology- commands. O'Reilly books, 2018.
- [3] Mark Summerfield. Programming in C++: A Complete Introduction to the C++ Language. Anaya Multimedia, 2009.
- [4] A. M. Turing. On computable numbers with an application to the Entscheidungs problem. Proceedings of the london mathematical society, 1997.

