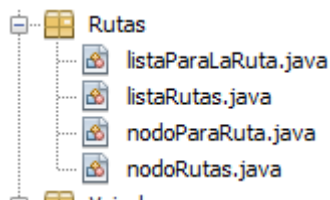


## Manual Tenido

### Método Main

```
1  package edd_proyecto2;
2
3  import Blockchain.listaBlock;
4  import Clientes.*;
5  import Conductores.*;
6  import Rutas.listaRutas;
7  import Veiculo.*;
8
9  public class EDD_Proyecto2 {
10
11      public static TablaH tablita = new TablaH();
12      public static ListaC lista = new ListaC();
13      public static ArbolB t = new ArbolB(3);
14      public static listaRutas listaRutas = new listaRutas();
15      public static listaBlock listaViajes = new listaBlock();
16
17      public static void main(String[] args) {
18          // TODO code application logic here
19
20          listaRutas.cargaMasiva();
21
22      }
23
```

Paquete Rutas (Carga de Rutas, Nodo para Grafo, Nodo para Lista simple de rutas)



Start Page X listaParaLaRuta.java X

Source History

1 package Rutas;

2

3 + import ...4 lines

7

8 public class listaParaLaRuta {

9

10 public nodoParaRuta inicioRuta;

11

12 + public listaParaLaRuta() {...3 lines }

15

16 + public void insertar(String contenido, String tiempo) {...12 lines }

28

29 + public void mostrarParaRuta(nodoParaRuta tempo) {...8 lines }

37

38 + void graficar() {...20 lines }

58

59 + public void graficarDesdeRuta(nodoParaRuta tempo) {...20 lines }

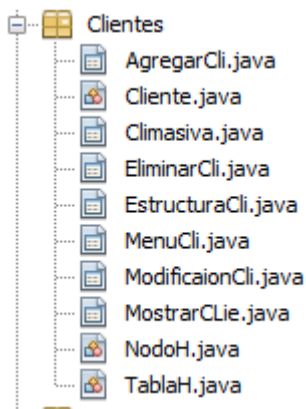
79

80 + String graficar2(nodoParaRuta tempo\_, FileWriter archivo) {...28 lines }

108

```
Start Page x listaRutas.java x
Source History
1 package Rutas;
2
3 import ...11 lines
14
15 public class listaRutas {
16
17     public nodoRutas inicio;
18     nodoRutas temporalOrigen;
19     String[][] distancias;
20     String[][] recorridos;
21
22     void insertarRuta(String origen_, String destino_, int tiempo) {...27 lines }
49
50     public listaRutas() {...3 lines }
53
54     boolean existeOrigen(String origen_, nodoRutas temporal) {...12 lines }
66
67     boolean existeDestino(String destino_, nodoRutas temporal) {...9 lines }
76
77     boolean vacia() {...3 lines }
80
81     nodoRutas insertarOrigenes(String origen_, nodoRutas temporal) {...9 lines }
90
91     nodoRutas insertarDestinos(String destino_, int tiempo_, nodoRutas temporal) {...8 lines }
99
100    public void mostrarOrigenes(nodoRutas tempo_) {...8 lines }
108
109    void mostrarDestinos(nodoRutas tempo_) {...8 lines }
117
118    public void cargarOrigenesComboBox(nodoRutas tempo_, JComboBox comboBox) {...8 lines }
126
127    public void cargaMasiva() {...26 lines }
153
154    void graficar() {...19 lines }
173
174    void graficar2(nodoRutas tempo_, FileWriter archivo) {...24 lines }
198
199    int calcularTamaño() {...9 lines }
208
209    void llenarRutas() {...51 lines }
260
261    void algoritmoFloyd() {...33 lines }
294
```

Paquete Clientes (Carga de Clientes, Tabla Hash, Agregar, eliminar, mostrar, modificar)



Start Page X Cliente.java X

Source History

1 + ...5 lines

6 package Clientes;

7

8 + /\*\*...4 lines \*/

12 public class Cliente {

13 String dpi, nombre,apellido,genero,fechnac,telefono,direccion;

14

15 + public Cliente(String dpi, String nombre, String apellido, String

24

25 + public Cliente() {...9 lines }

34

35 + public String getDpi() {...3 lines }

38

39 + public void setDpi(String dpi) {...3 lines }

42

43 + public String getNombre() {...3 lines }

46

47 + public void setNombre(String nombre) {...3 lines }

50

51 + public String getApellido() {...3 lines }

54

55 + public void setApellido(String apellido) {...3 lines }

58

59 + public String getGenero() {...3 lines }

62

63 + public void setGenero(String genero) {...3 lines }

66

67 + public String getFechnac() {...3 lines }

70

71 + public void setFechnac(String fechnac) {...3 lines }

74

75 + public String getTelefono() {...3 lines }

78

79 + public void setTelefono(String telefono) {...3 lines }

82

83 + public String getDireccion() {...3 lines }

86

87 + public void setDireccion(String direccion) {...3 lines }

90

StartPage X TablaH.java X

Source History

1 + ...5 lines

6 package Clientes;

7

15

16 + /\*\*...4 lines \*/

20 public class TablaH {

21

22 int m = 37;

23 NodoH[] tabla = new NodoH[m];

24 int tama = 0;

25 float pm = 0;

26

27 + public TablaH() {...5 lines }

32

33 + public BigInteger GenerarLLave(String valor) {...9 lines }

42

43 + public void Insertar(Cliente dato) {...26 lines }

69

70 + public float Porcentaje() {...9 lines }

79

80 + public void Imprimir() {...23 lines }

103

104 + public void cargarCombobox(JComboBox comboBox) {...25 lines }

129

130 + public String DatosCargados() {...20 lines }

150

151 + public void Imprimir2() {...11 lines }

162

163 + private NodoH[] Rehash() {...23 lines }

186

187 + private void ITablaR(NodoH[] n, NodoH e) {...15 lines }

202

203 + public boolean Eliminar(String el) {...43 lines }

246

247 + public NodoH Busqueda(String d) {...18 lines }

265

266 + public boolean Modificar(String d, Cliente mo) {...8 lines }

274

275 + public NodoH[] getTabla() {...3 lines }

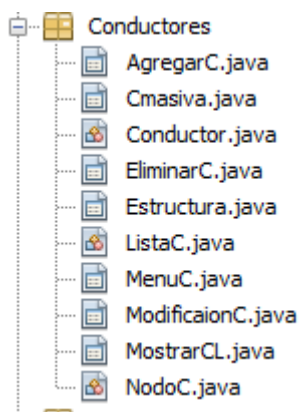
278

279 + public void setTabla(NodoH[] tabla) {...3 lines }

282

```
1  ...5 lines
6  package Clientes;
7
8  /**...4 lines */
12 public class NodoH {
13     Cliente dato;
14     NodoH derecha;
15
16     public NodoH() {...4 lines }
20
21     public NodoH(Cliente dato, NodoH derecha) {...4 lines }
25     public NodoH(Cliente dato) {...4 lines }
29
30     public Cliente getData() {...3 lines }
33
34     public void setData(Cliente dato) {...3 lines }
37
38     public NodoH getDerecha() {...3 lines }
41
42     public void setDerecha(NodoH derecha) {...3 lines }
45
```

Paquete Conductores (Carga masiva, lista circular, agregar, eliminar, modificar, mostrar estructura)



Start Page X ListaC.java X

Source History

1 + ...5 lines

6 package Conductores;

7

15

16 + /\*\*...4 lines \*/

20 public class ListaC {

21

22 int tam;

23 NodoC inicio;

24 NodoC fin;

25

26 + public ListaC() {...5 lines }

31

32 + public int getTam() {...3 lines }

35

36 + public void setTam(int tam) {...3 lines }

39

40 // inserta de ultimo en la fila :)

41 + public void InsertarUltimo(Conductor d) {...25 lines }

66 // devuelve un nodo con el dato usado

67

68 + public NodoC Buscar(String d) {...14 lines }

82

83 + public boolean BuscarB(String d) {...14 lines }

97

98 // metodo de modificacion

99 + public void ModificarD(String dato, Conductor modif) {...4 lines }

103 // elimina dato

104

105 + public void EliminarD(String dato) {...21 lines }

126 // imprime en consola la lista

127

128 + public void Imprimir() {...11 lines }

139

140 + public void cargaCombobox(JComboBox comboBox) {...12 lines }

152

153 + public String DevolverCg() {...14 lines }

167 // ordena la ista de mayor a menor

168

169 + public void ordenarListaMayorM() {...35 lines }

204 // ordena la lista de menor a mayor

205

StartPage x Conductor.java x

Source History

1 + ...5 lines

6 package Conductores;

7

8 + /\*\*...4 lines \*/

12 public class Conductor {

13 private String dpi, nombre, apellido, TdeL, genero, Fnac, telefono, dir

14

15 + public Conductor(String dpi, String nombre, String apellido, String Tde

25

26

27

28 + public Conductor() {...10 lines }

38

39 + public String getGenero() {...3 lines }

42

43 + public void setGenero(String genero) {...3 lines }

46

47 + public String getDpi() {...3 lines }

50

51 + public void setDpi(String dpi) {...3 lines }

54

55 + public String getNombre() {...3 lines }

58

59 + public void setNombre(String nombre) {...3 lines }

62

63 + public String getApellido() {...3 lines }

66

67 + public void setApellido(String apellido) {...3 lines }

70

71 + public String getTdeL() {...3 lines }

74

75 + public void setTdeL(String TdeL) {...3 lines }

78

79 + public String getFnac() {...3 lines }

82

83 + public void setFnac(String Fnac) {...3 lines }

86

87 + public String getTelefono() {...3 lines }

90

91 + public void setTelefono(String telefono) {...3 lines }

94

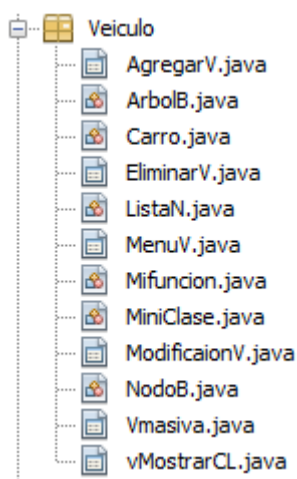
95 + public String getDireccion() {...3 lines }

98



```
1  ...5 lines
6  package Conductores;
7
8  /**...4 lines */
12 public class NodoC {
13     Conductor dato;
14     NodoC siguiente;
15     NodoC atras;
16
17     public NodoC() {...5 lines }
22
23     public NodoC(Conductor dato) {...5 lines }
28
29     public Conductor getDato() {...3 lines }
32
33     public void setDato(Conductor dato) {...3 lines }
36
37     public NodoC getSiguiente() {...3 lines }
40
41     public void setSiguiente(NodoC siguiente) {...3 lines }
44
45     public NodoC getAtras() {...3 lines }
48
49     public void setAtras(NodoC atras) {...3 lines }
52
```

Paquete Vehículo (Carga masiva, agregar, eliminar, Árbol B, mostrar, modificar)



Start Page X ArbolB.java X

Source History

1 + ...5 lines

6 package Veiculo;

7

8 + import ...5 lines

13

14 + /\*\*...4 lines \*/

18 public class ArbolB {

19

20     NodoB raiz;

21     int gradoM;

22     int tam;

23

24 + public ArbolB(int gradoM\_) {...5 lines }

29

30 + public void Ge() {...54 lines }

84

85 + public void ImprimirB() {...5 lines }

90

91 + public void cargaVehiculos1(JComboBox combo) {...5 lines }

96

97 + public String Datos() {...6 lines }

103

104 + public Carro BusquedaB(String n) {...6 lines }

110

111 + public int Indi(String n) {...6 lines }

117

118 + public void InsertarB(Carro n) {...24 lines }

142

143 + public boolean EliminarB(String n) {...16 lines }

159

160 + public int ValuString(String n) {...12 lines }

172

173 + public boolean Modificar(String n, Carro s) {...8 lines }

Start Page X Nodob.java X

Source History

1 + ...5 lines

6 package Veiculo;

7

8 + import ...6 lines

14

15 + /\*\*...4 lines \*/

19 public class Nodob {

20

21 Carro LLave[];

22 int grado;

23 Nodob segun[];

24 int nA;

25 boolean hoja;

26 String co;

27 String map[] = new String[5];

28

29 + Nodob(int gradol, boolean hojal) {...10 lines }

39

40 + public Carro[] getLLave() {...3 lines }

43

44 + public void setLLave(Carro[] LLave) {...3 lines }

47

48 + public int getGrado() {...3 lines }

51

52 + public void setGrado(int grado) {...3 lines }

55

56 + public Nodob[] getSegun() {...3 lines }

59

60 + public void setSegun(Nodob[] segun) {...3 lines }

63

64 + public boolean isHoja() {...3 lines }

67

68 + public void setHoja(boolean hoja) {...3 lines }

71

72 + public int getnA() {...3 lines }

75

76 + public void setnA(int nA) {...3 lines }

79

80 + public void Imprimir() {...16 lines }

96

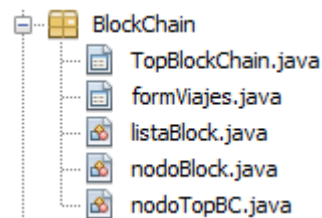
97 + public void cargaVehiculos2(JComboBox combo) {...16 lines }

113

114 + public String Texto() {...16 lines }

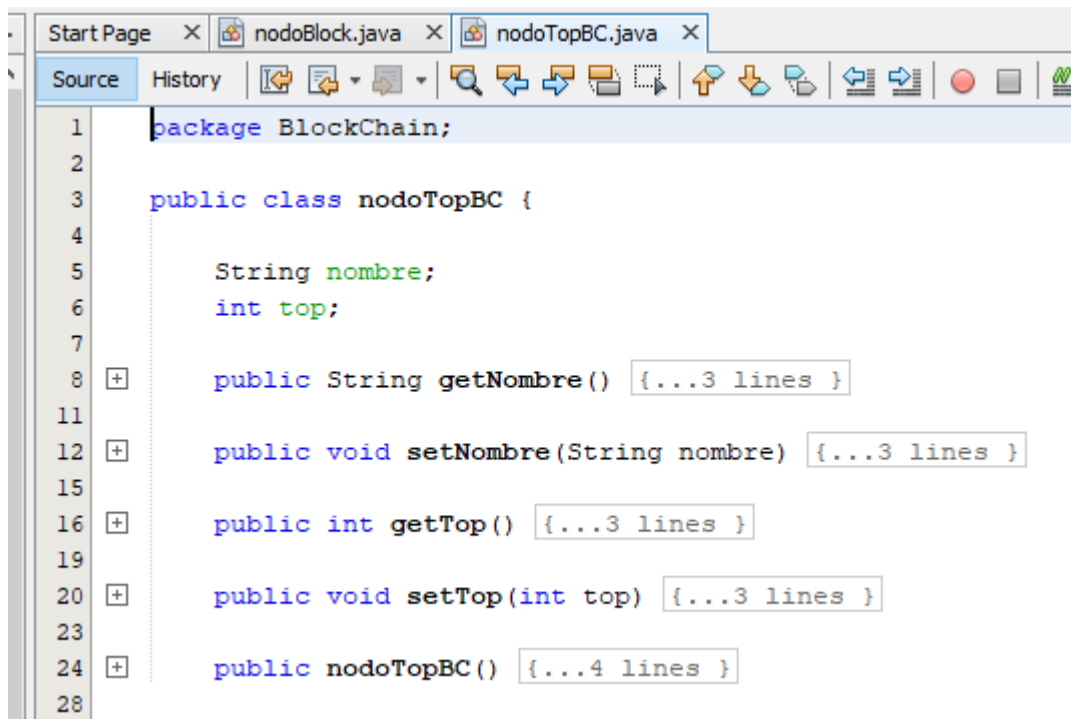
```
1  ...5 lines
6  package Veiculo;
7  /**...4 lines */
11 public class Carro {
12     String plasca, marca, modelo, ano, color, precio, transmision;
13
14     public Carro() {...9 lines }
23
24     public Carro(String plasca, String marca, String modelo, String ano, Stri
33
34     public String getPlasca() {...3 lines }
37
38     public void setPlasca(String plasca) {...3 lines }
41
42     public String getMarca() {...3 lines }
45
46     public void setMarca(String marca) {...3 lines }
49
50     public String getModelo() {...3 lines }
53
54     public void setModelo(String modelo) {...3 lines }
57
58     public String getAno() {...3 lines }
61
62     public void setAno(String ano) {...3 lines }
65
66     public String getColor() {...3 lines }
69
70     public void setColor(String color) {...3 lines }
73
74     public String getPrecio() {...3 lines }
77
78     public void setPrecio(String precio) {...3 lines }
81
82     public String getTransmision() {...3 lines }
85
86     public void setTransmision(String transmision) {...3 lines }
89
```

Paquete Block Chain (Agregar viajes, mostrar estructuras, Tops, reportes)



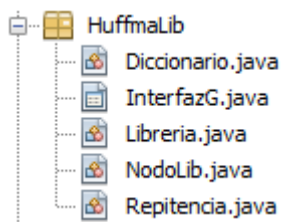
```
Start Page x listaBlock.java x
Source History
1 package Blockchain;
2
3 + import ...14 lines
17
18 public class listaBlock {
19
20     nodoBlock inicio;
21     nodoTopBC[] topActual;
22
23 + public listaBlock() {...3 lines }
26
27 + public void insertarViaje(String origen_, String destino_, String cliente, String
50
51 + String generarFechaHoraLlave() {...6 lines }
57
58 + String generarFechaHoraNodo() {...6 lines }
64
65 + String encriptarMD5(String llave_) {...16 lines }
81
82 + public void graficar() {...19 lines }
101
102 + void graficar2(nodoBlock tempo_, FileWriter archivo) {...24 lines }
126
127 + public void topClientes(JTextArea area) {...17 lines }
144
145 + public void topConductores(JTextArea area) {...17 lines }
162
163 + public void topVehiculos(JTextArea area) {...17 lines }
180
181 + public void topViajes(JTextArea area) {...17 lines }
198
199 + void buscarEnTopViajes(String contenido, nodoParaRuta tempo) {...19 lines }
218
219 + void buscarEnTop(String contenido) {...19 lines }
238
239 + void ordenarTop() {...19 lines }
258
259 + public void rutaPorLlave(String llave_, JTextArea area) {...20 lines }
```

```
Start Page X nodoBlock.java X
Source History
1 package Blockchain;
2
3 + import ...4 lines
7
8 public class nodoBlock {
9
10     String llave;
11     nodoBlock derecha;
12     String lugarOrigen;
13     String lugarDestino;
14     String fechaHora;
15     nodoParaRuta ruta;
16     NodoH cliente;
17     NodoC conductor;
18     Carro vehiculo;
19
20 + public Carro getVehiculo() {...3 lines }
23
24 + public void setVehiculo(Carro vehiculo) {...3 lines }
27
28 + public nodoBlock(String lugarOrigen, String lugarDestino) {...5 lines }
33
34 + public NodoH getCliente() {...3 lines }
37
38 + public void setCliente(NodoH cliente) {...3 lines }
41
42 + public NodoC getConductor() {...3 lines }
45
46 + public void setConductor(NodoC conductor) {...3 lines }
49
50 + public String getLlave() {...3 lines }
53
54 + public void setLlave(String llave) {...3 lines }
57
58 + public nodoBlock getDerecha() {...3 lines }
61
62 + public void setDerecha(nodoBlock derecha) {...3 lines }
65
66 + public String getLugarOrigen() {...3 lines }
69
70 + public void setLugarOrigen(String lugarOrigen) {...3 lines }
73
74 + public String getLugarDestino() {...3 lines }
```



```
1 package Blockchain;
2
3 public class nodoTopBC {
4
5     String nombre;
6     int top;
7
8     public String getNombre() { ...3 lines }
9
10
11     public void setNombre(String nombre) { ...3 lines }
12
13
14     public int getTop() { ...3 lines }
15
16
17     public void setTop(int top) { ...3 lines }
18
19
20     public nodoTopBC() { ...4 lines }
21
22
23
24
25
26
27
28 }
```

Paquete Huffman (método de huffman, diccionario, repitencias, librería)



```
Start Page X Libreria.java X
Source History
1 + ...5 lines
6 package HuffmaLib;
7
8 + /**...4 lines */
12 public class Libreria {
13
14     NodoLib no1 = null, no2 = null, no3;
15     NodoLib raiz[] = new NodoLib[200];
16
17 + public Libreria() {...2 lines }
19
20 + public void LlenarR(Repitencia n[]) {...5 lines }
25
26 + void ParPeque(int numnodes) {...20 lines }
46
47 + public void GenerarArbol(Repitencia n[], Diccionario jd[]) {...19 lines }
66
67 + void LlenarDic(int ss,Diccionario dic[]) {...26 lines }
93 }
```

```
Start Page X Diccionario.java X
Source History
1 + ...5 lines
6 package HuffmaLib;
7
8 + /**...4 lines */
12 public class Diccionario {
13     char simbolo;
14     String binario;
15     int cc;
16
17 + public Diccionario(char simbolo, String binario, int cc) {...5 lines }
22
23
24
25 + public String getBinario() {...3 lines }
28
29 + public void setBinario(String binario) {...3 lines }
32
33 + public char getSimbolo() {...3 lines }
36
37 + public void setSimbolo(char simbolo) {...3 lines }
40
41 + public int getCc() {...3 lines }
44
45 + public void setCc(int cc) {...3 lines }
48
49
```



StartPage x Nodolib.java x

Source History

1 + ...5 lines

6 package HuffmaLib;

7

8 + /\*\*...4 lines \*/

12 public class Nodolib {

13 char letra;

14 char tipoHijo;

15 int repitencia;

16 Nodolib siguiente;

17

18 + public Nodolib(char letra, int repitencia) {...6 lines }

24 + public Nodolib(char letra, int repitencia, Nodolib sig) {...6 lines }

30 + public Nodolib() {...6 lines }

36

37 + public char getLetra() {...3 lines }

40

41 + public Nodolib getSiguiente() {...3 lines }

44

45 + public void setSiguiente(Nodolib siguiente) {...3 lines }

48

49 + public void setLetra(char letra) {...3 lines }

52

53 + public char getTipoHijo() {...3 lines }

56

57 + public void setTipoHijo(char tipoHijo) {...3 lines }

60

61 + public int getRepitencia() {...3 lines }

64

65 + public void setRepitencia(int repitencia) {...3 lines }

68

69 + public Nodolib getPadre() {...3 lines }

72

73 + public void setPadre(Nodolib padre) {...3 lines }

```
1  ...5 lines
6  package HuffmaLib;
7
8  /**...4 lines */
12 public class Repitencia {
13
14     char letra;
15     int repi;
16
17     public Repitencia(char letra, int repi) {...4 lines }
21
22     public Repitencia() {...4 lines }
26
27     public char getLetra() {...3 lines }
30
31     public void setLetra(char letra) {...3 lines }
34
35     public int getRepi() {...3 lines }
38
39     public void setRepi(int repi) {...3 lines }
42
```

Paquete Menu Principal (llamado a menú general)

