Pila_2

private int tope

private int max

private char elem[]

private char aux[]

private Boolean band

public int top()

public boolean pilaLlena()

public Boolean pilaVacia()

public void apilar()

public void desapilar()

public void imprimir()

public void vaciar()

public void info()

public void invertir()

public int contar()

public int buscar()

public Boolean vaciarPalindromo()

Código.

Método Main

```java
package Apps_Pila;
//@author Luis Ricardo Reyes Villar
import java.util.Scanner;
public class Main_Apps_Pila {
    public static void main(String[] args) {
        Scanner Leer = new Scanner(System.in);
        int opc, resp, tamaño;
        String texto, textM;
        char car;
        System.out.print("Escribe la palabra (sin espacios): ");
        texto = Leer.nextLine();
        textM = texto.replaceAll(" ", "");
        tamaño = textM.length();
        Pila_2 A = new Pila_2(tamaño);
        System.out.println("Especifique la Aplicacion que desea manejar sobre '"+texto+"'\nl. Palíndromos
            + "3. Conversión de notación infija a posfija");
        opc = Leer.nextInt();
        System.out.println("");
        switch (opc) {
            case 1:
                for (int i = 0; i < textM.length(); i++) {
                    car = (textM.toLowerCase()).charAt(i);
                    A.apilar(car);
                }
                A.invertir();
                if (A.validarPalindromo()) {
                    System.out.println("Sí es palíndromp.");
                } else {
                    System.out.println("No es palíndromo.");
                }
                break;
            case 2:
                if (A.validarParentesis(textM)) {
                    System.out.println(textM + " está balanceado.");
                } else {
                    System.out.println(textM + " no está balanceado.");
                }
                break;
            case 3:
                System.out.println("Notación Infija: " + textM);
                System.out.print("Notación Postfija: ");
                System.out.println(A.infijaPostfija(textM));
                break;
            default:
                System.out.println("No existe la opción seleccionada.");
        }
        System.out.println("");
    }
}
```

Clase

```java
package Apps_Pila;
//@author Luis Ricardo Reyes Villar
public class Pila_2 {
    private int tope;
    private int max;
    private char elem[];
    private char aux[];
    private boolean band;

    public Pila_2(int n) {
        this.max = n;
        this.elem = new char[max];
        this.aux = new char[max];
        this.tope = -1;
    }

    public int top() {
        return tope;
    }

    public boolean pilaLlena() {
        if (tope==max-1) {
            return true;
        }else{
            return false;
        }
    }

    public boolean pilaVacia() {
        if (tope==-1) {
            return true;
        }else{
            return false;
        }
    }

    public void apilar(char dato) {
        if (pilaLlena()==true) {
            System.out.println("La pila está llena.");
        }else{
            tope++;
            elem[tope] = dato;
        }
    }

    public void desapilar() {
        char dato;
        if (pilaVacia()==true) {
            System.out.println("La pila esta vacía.");
        }else{
            dato = elem[tope];
            tope--;
        }
    }

    public void imprimir() {
        if (pilaVacia()==true) {
            System.out.println("La pila no contiene datos.");
        }else{
            for (int i = 0; i <= tope; i++) {
                System.out.println((i + 1) + ". Elemento: " + elem[i]);
            }
        }
    }
}
```

```
66      public void vaciar() {
67          tope=-1;
68          System.out.println("Se ha vaciado la pila.");
69      }
70
71      public void info() {
72          if (tope>-1) {
73              System.out.println("El último elemento apilado es: " + elem[tope]);
74          }else{
75              System.out.println("No existen elementos en la pila");
76          }
77      }
78
79      public void invertir() {
80          if (pilaVacia()==true) {
81              System.out.println("La pila está vacía");
82          }else{
83              System.out.println("Pila Original");
84              imprimir();
85              int n=0;
86              System.out.println("");
87
88              System.out.println("Pila Invertida");
89              for (int i = tope; i > -1; i--) {
90                  aux[n] = elem[i];
91                  System.out.println((n + 1) + ". Elemento: " + aux[n]);
92                  n++;
93              }
94          }
95      }
96
97      public int contar() {
98          return tope+1;
99      }
100
101     public int buscar(int dato) {
102         int pos=-1;
103
104         for (int i = 0; i <= tope; i++) {
105             if (dato == elem[i]) {
106                 pos = i;
107                 return pos;
108             }
109         }
110         return pos;
111     }
112
113     public boolean validarPalindromo() {
114         for (int i = 0; i <= tope; i++) {
115             if (aux[i]!=elem[i]) {
116                 return false;
117             }
118         }
119         return true;
120     }
121
122     public boolean validarParentesis(String texto) {
123         char dato;
124
125         for (int i = 0; i < texto.length(); i++) {
126             if ((texto.charAt(i) == '(') || (texto.charAt(i) == '{') || (texto.charAt(i) == '[')) {
127                 dato = texto.charAt(i);
128                 apilar(dato);
```

```
129                 } else if ((texto.charAt(i) == ')') && (elem[tope] == '(') || (texto.charAt(i) == '}') && (e
130                     desapilar();
131                 }
132             }
133             return pilaVacia();
134         }
135 
136         public char[] infijaPostfija(String texto) {
137             String jerarquia = "^*/+- ";
138             int pos_inf = 0;
139             int pos_jer = 0;
140 
141             for (int i = 0; i < elem.length; i++) {
142                 elem[i] = texto.charAt(i);
143             }
144 
145             while (pos_jer != jerarquia.length() - 1) {
146                 if (pos_inf==texto.length()) {
147                     pos_inf = 0;
148                     pos_jer++;
149                 }
150                 if (texto.charAt(pos_inf) == jerarquia.charAt(pos_jer)) {
                        char aux = ' ';
152                     aux = elem[pos_inf];
153                     elem[pos_inf] = elem[pos_inf+1];
154                     elem[pos_inf+1] = aux;
155                 }
156                 pos_inf++;
157             }
158             return elem;
159         }
160     }
```

Corridas

```
run:
Escribe la palabra (sin espacios): dabalearrozalazorraelabad
Especifique la Aplicacion que desea manejar sobre 'dabalearrozalazorraelabad'
1. Pal□ndromos
2. Validar Par□ntesis
3. Conversi□n de notaci□n infija a posfija
1

Pila Original
1. Elemento: d
2. Elemento: a
3. Elemento: b
4. Elemento: a
5. Elemento: l
6. Elemento: e
7. Elemento: a
8. Elemento: r
9. Elemento: r
10. Elemento: o
11. Elemento: z
12. Elemento: a
13. Elemento: l
14. Elemento: a
15. Elemento: z
16. Elemento: o
17. Elemento: r
18. Elemento: r
19. Elemento: a
20. Elemento: e
21. Elemento: l
22. Elemento: a
23. Elemento: b
24. Elemento: a
25. Elemento: d

Pila Invertida
```

```
Pila Invertida
1. Elemento: d
2. Elemento: a
3. Elemento: b
4. Elemento: a
5. Elemento: l
6. Elemento: e
7. Elemento: a
8. Elemento: r
9. Elemento: r
10. Elemento: o
11. Elemento: z
12. Elemento: a
13. Elemento: l
14. Elemento: a
15. Elemento: z
16. Elemento: o
17. Elemento: r
18. Elemento: r
19. Elemento: a
20. Elemento: e
21. Elemento: l
22. Elemento: a
23. Elemento: b
24. Elemento: a
25. Elemento: d
S□ es pal□ndromp.

BUILD SUCCESSFUL (total time: 37 seconds)
```

```
run:
Escribe la palabra (sin espacios): holabebe
Especifique la Aplicacion que desea manejar sobre 'holabebe'
1. Pal□ndromos
2. Validar Par□ntesis
3. Conversi□n de notaci□n infija a posfija
1

Pila Original
1. Elemento: h
2. Elemento: o
3. Elemento: l
4. Elemento: a
5. Elemento: b
6. Elemento: e
7. Elemento: b
8. Elemento: e

Pila Invertida
1. Elemento: e
2. Elemento: b
3. Elemento: e
4. Elemento: b
5. Elemento: a
6. Elemento: l
7. Elemento: o
8. Elemento: h
No es pal□ndromo.

BUILD SUCCESSFUL (total time: 4 seconds)

run:
Escribe la palabra (sin espacios): (1+2)*(4+2)
Especifique la Aplicacion que desea manejar sobre '(1+2)*(4+2)'
1. Pal□ndromos
2. Validar Par□ntesis
3. Conversi□n de notaci□n infija a posfija
2

(1+2)*(4+2) est□ balanceado.

BUILD SUCCESSFUL (total time: 17 seconds)
```

```
run:
Escribe la palabra (sin espacios): ((1+2*3)(1+3*2)
Especifique la Aplicacion que desea manejar sobre '((1+2*3)(1+3*2)'
1. Pal□ndromos
2. Validar Par□ntesis
3. Conversi□n de notaci□n infija a posfija
2

((1+2*3)(1+3*2) no est□ balanceado.

BUILD SUCCESSFUL (total time: 22 seconds)


run:
Escribe la palabra (sin espacios): (a+b)/a*c*d
Especifique la Aplicacion que desea manejar sobre '(a+b)/a*c*d'
1. Pal□ndromos
2. Validar Par□ntesis
3. Conversi□n de notaci□n infija a posfija
3

Notaci□n Infija: (a+b)/a*c*d
Notaci□n Postfija: (ab+)a/c*d*

BUILD SUCCESSFUL (total time: 24 seconds)
run:
Escribe la palabra (sin espacios): a+b*c
Especifique la Aplicacion que desea manejar sobre 'a+b*c'
1. Pal□ndromos
2. Validar Par□ntesis
3. Conversi□n de notaci□n infija a posfija
3

Notaci□n Infija: a+b*c
Notaci□n Postfija: ab+c*

BUILD SUCCESSFUL (total time: 4 seconds)
```