

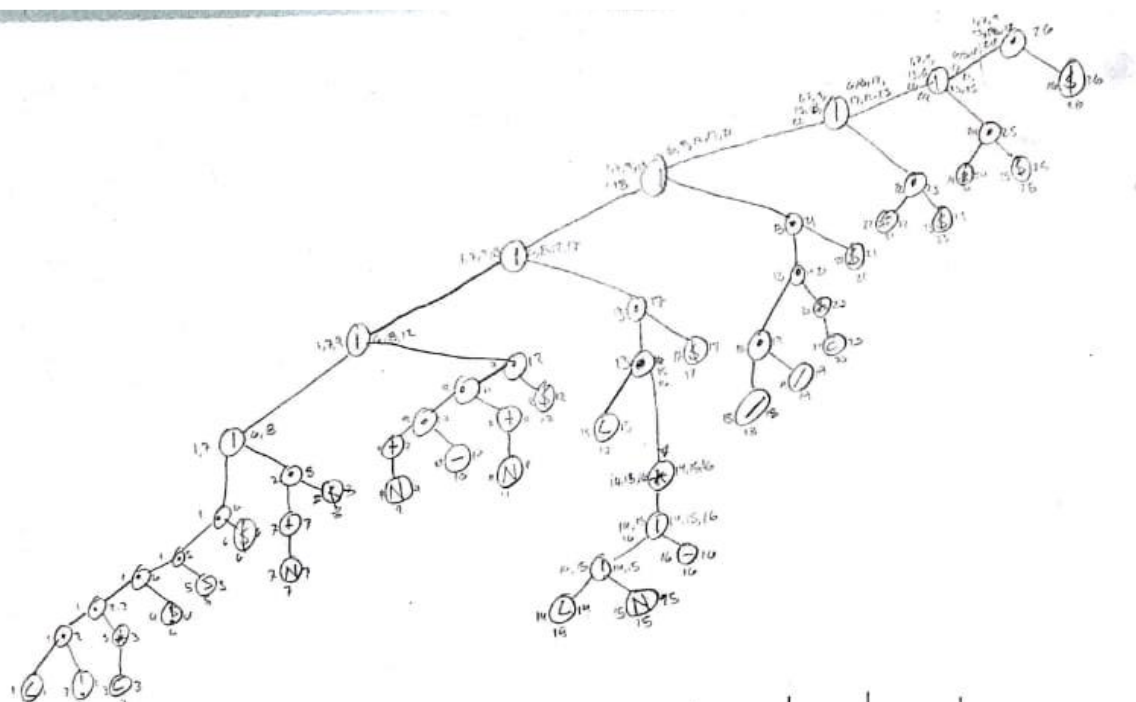
Manual Técnico

Para la realización del proyecto se emplearon los usos de diversas herramientas y metodologías de las cuales se basan para la realización de los analizadores, etc.

Ambiente de desarrollo

- Desarrollado en java
- IDE Netbeans 8.2

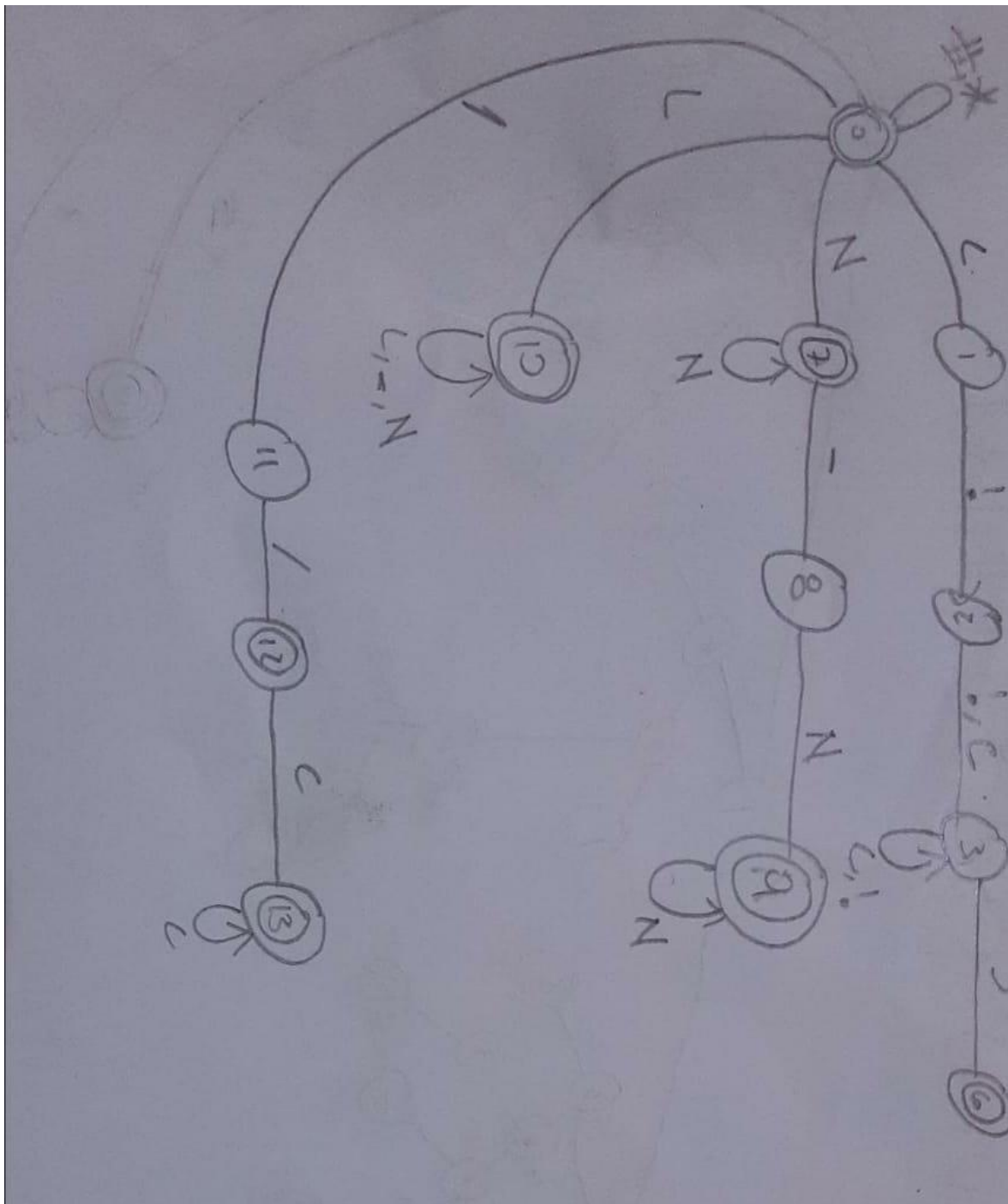
Para el primer archivo se utilizó el siguiente método del árbol para la realización del autómata.



	2	1	1
	3	2	3, 4
	4	3	3, 4
•	5	4	5
	6	5	6
→	7	6	—
	8	7	7, 8
→	9	8	—
	10	9	9, 10
	11	10	—
	12	11	11, 12
→	13	12	—
	14	13	14, 15, 16
	15	14	14, 15, 16, 17
	16	15	14, 15, 16, 17
→	17	16	—
	18	17	18
	19	18	20, 21
	20	19	20, 21
→	21	20	—
	22	21	23
→	23	22	—
	24	23	25
→	25	24	—
→	26	25	—

	C	!	C	>	N	-	L	/	*	#
\leq $S_0 = 1$ $S_1 = 2$ $S_2 = 3, 4$ $S_3 = 5$ $G = S_4 = \text{acceptor}$ $S_5 = 7, 8$ $S_6 = 9 = S_5$ $S_7 = 9, 10$ $S_8 = 11$ $S_9 = 13$ $S_{10} = 14, 15, 16$ $S_{11} = 18$ $S_{12} = 19$ $S_{13} = 20$ $S_{14} = 21$ $S_{15} = 25$		S_1 S_2 S_3	S_2 S_7	S_4	S_5 S_7 S_8 S_{10}	S_6	S_{10} S_{11}	S_{12}		

El cual al realizar los respectivos pasos de este se obtiene el siguiente autómata.



Dado esto se puede empezar a realizar el autómata ya en el código, el cual es realizado por medio de recorrer la entrada carácter a carácter y luego por medio de un método switch determinar el estado del autómata donde nos encontramos,

para por último en cada estado comparar las condiciones de transición a otros o simplemente para quedarnos en el mismo o aceptar algún valor.

```
public void AnalisisTableroNivelesV2(String cadena){
    tablero=new ArrayList<Character>();
    int linea = 1;
    int columna = 0;
    String cadconcat="";
    int estado=0;
    int caracter =0;
    char caracteractual;
    String simbolo="";
    String posicion="";
    int nivel =0;
    int N=0;
    int P=0;
    String id="";
    ArrayList<Character> tableroaux =new ArrayList<Character>();
    for(int i=0;i<cadena.length();i++){
        caracteractual=cadena.charAt(i);
        caracter=(int)caracteractual;
        columna++;
        switch(estado){
            case 0:
                if(caracter == 60){
                    cadconcat+=String.valueOf(caracteractual);
                    estado = 1;
                }else if(caracter>47&&caracter<58){
                    cadconcat+=String.valueOf(caracteractual);
                    if(cadena.charAt(i+1)=='\n' || cadena.charAt(i+1)=='\r'){
                        if(Integer.parseInt(cadconcat)<=10&&Integer.parseInt(cadconcat)>=3){
                            nivel=Integer.parseInt(cadconcat);
                            tokens.add(new Tokens(linea,columna,"Numero",String.valueOf(nivel)));
                            cadconcat="";
                        }else{
                            System.out.println("ERROR EL NIVEL NO ES PERMITIDO");
                        }
                    }
                }
            if(this.tablero.size() !=0){
                tableroaux=tablero;
                tab=new char[N][P];
                Llenado(tableroaux,N,P);
                niveles.add(new Nivel(nivel,N,P,id,tab,tableroaux));
                id="";
                N=0;
                P=0;
                tableroaux=new ArrayList<Character>();
                tablero =new ArrayList<Character>();
                this.tablero.clear();
                tableroaux.clear();
            }
        }
    }
}
```

```

        tableroaux.clear();

        }
        estado=7;
    }else if ((caracter>64&&caracter<91) || (caracter>96&&caracter<123)) {
        cadconcat+=String.valueOf(caracteractual);
        estado=10;
    }else if(caracter==47){
        cadconcat+=String.valueOf(caracteractual);
        estado=11;
    }else if(caracter==35||caracter==42){
        estado=0;
        tokens.add(new Tokens(linea,columna,"Token: "+caracteractual,String.valueOf(caracteractual)));
        tablero.add(caracteractual);
        /*if(((int)cadena.charAt(i+2)>47&&(int)cadena.charAt(i+2)<58) || ((int)cadena.charAt(i+3)>47&&(int)cadena
            tableroaux=tablero;
            niveles.add(new Nivel(nivel,N,P,id,tableroaux));
            id="";
            N=0;
            P=0;
        }*/
    }else if(caracter==32||caracter==10||caracter==13||caracter==9){
        if(caracter==10||caracter==13){
            linea++;
            columna=0;
        }
    }else{
        Errores.add("Caracter erroneo: "+caracteractual+" Linea: "+linea+" Columna: "+columna);
    }
    break;
case 1:
    if(caracter==33){
        cadconcat+=String.valueOf(caracteractual);
        estado=2;
    }else{
        Errores.add("Caracter erroneo: "+caracteractual+" Linea: "+linea+" Columna: "+columna);
        estado=2;
    }
    break;
case 2:
    if(caracter==10||caracter==13){
        linea++;
        columna=0;
    }
    cadconcat+=String.valueOf(caracteractual);
    estado=3;
    break;

```

```

        estado=3;
        break;
    case 3:
        if(caracter==62){
            cadconcat+=String.valueOf(caracteractual);
            cadconcat.replace('\r', ' ');
            cadconcat.replace(':', ' ');
            estado=0;
            tokens.add(new Tokens(linea,columna,"Comentario Multilinea",cadconcat.replace('\n', ' ')));
            cadconcat="";
        }else{
            if(caracter==10||caracter==13){
                linea++;
                columna=0;
            }
            cadconcat+=String.valueOf(caracteractual);
            estado=3;
        }
        break;
    case 7:
        if(caracter>47&&caracter<58){
            cadconcat+=String.valueOf(caracteractual);
            if(cadena.charAt(i+1)=='\n' || cadena.charAt(i+1)=='r'){
                if(Integer.parseInt(cadconcat)<=10&&Integer.parseInt(cadconcat)>=3){
                    nivel=Integer.parseInt(cadconcat);
                    tokens.add(new Tokens(linea,columna,"Numero",String.valueOf(nivel)));
                    cadconcat="";
                    linea++;
                    columna=0;
                }else{
                    System.out.println("ERROR EL NIVEL NO ES PERMITIDO");
                }
            }
            estado=7;
        }else if(caracter==45){
            N=Integer.parseInt(cadconcat);
            tokens.add(new Tokens(linea,columna,"Numero",String.valueOf(N)));
            cadconcat="";
            tokens.add(new Tokens(linea,columna,"Token: "+caracteractual,String.valueOf(caracteractual)));
            estado = 8;
        }else if(caracter==32||caracter==9||caracter==10||caracter==13){
            if(caracter==10||caracter==13){
                linea++;
                columna=0;
            }
        }else{
            Errores.add("Caracter erroneo: "+caracteractual+" Linea: "+linea+" Columna: "+columna);
        }
    }
}

```

```

        }else{
            Errores.add("Caracter erroneo: "+caracteractual+" Linea: "+linea+" Columna: "+columna);
        }
        break;
    case 8:
        if(caracter>47&&caracter<58){
            cadconcat+=String.valueOf(caracteractual);
            estado=9;
            if(((int)cadena.charAt(i+1)>64&&(int)cadena.charAt(i+1)<91)||((int)cadena.charAt(i+1)>96&&(int)cadena.charAt(i+1)<123)){
                P=Integer.parseInt(cadconcat);
                tokens.add(new Tokens(linea,columna,"Numero",String.valueOf(P)));
                cadconcat="";
                estado=0;
            }
        }else if(caracter==32||caracter==9){
        }else{
            Errores.add("Caracter erroneo: "+caracteractual+" Linea: "+linea+" Columna: "+columna);
        }
        break;
    case 9:
        if(caracter>47&&caracter<58){
            cadconcat+=String.valueOf(caracteractual);
            estado=9;
            if(((int)cadena.charAt(i+1)>64&&(int)cadena.charAt(i+1)<91)||((int)cadena.charAt(i+1)>96&&(int)cadena.charAt(i+1)<123)||((int)cadena.charAt(i+1)==32||((int)cadena.charAt(i+1)==9){
                P=Integer.parseInt(cadconcat);
                tokens.add(new Tokens(linea,columna,"Numero",String.valueOf(P)));
                cadconcat="";
                estado=0;
            }
        }else if(caracter==32||caracter==9){
        }else{
            Errores.add("Caracter erroneo: "+caracteractual+" Linea: "+linea+" Columna: "+columna);
        }
        break;
    case 10:
        if((caracter>64&&caracter<91)||((caracter>96&&caracter<123)){
            cadconcat+=String.valueOf(caracteractual);
            estado=10;
        }else if(caracter==95){
            cadconcat+=String.valueOf(caracteractual);
            estado=10;
        }else if(caracter>47&&caracter<58){
            cadconcat+=String.valueOf(caracteractual);
            estado=10;
        }else if(caracter==10||caracter==13){
        }

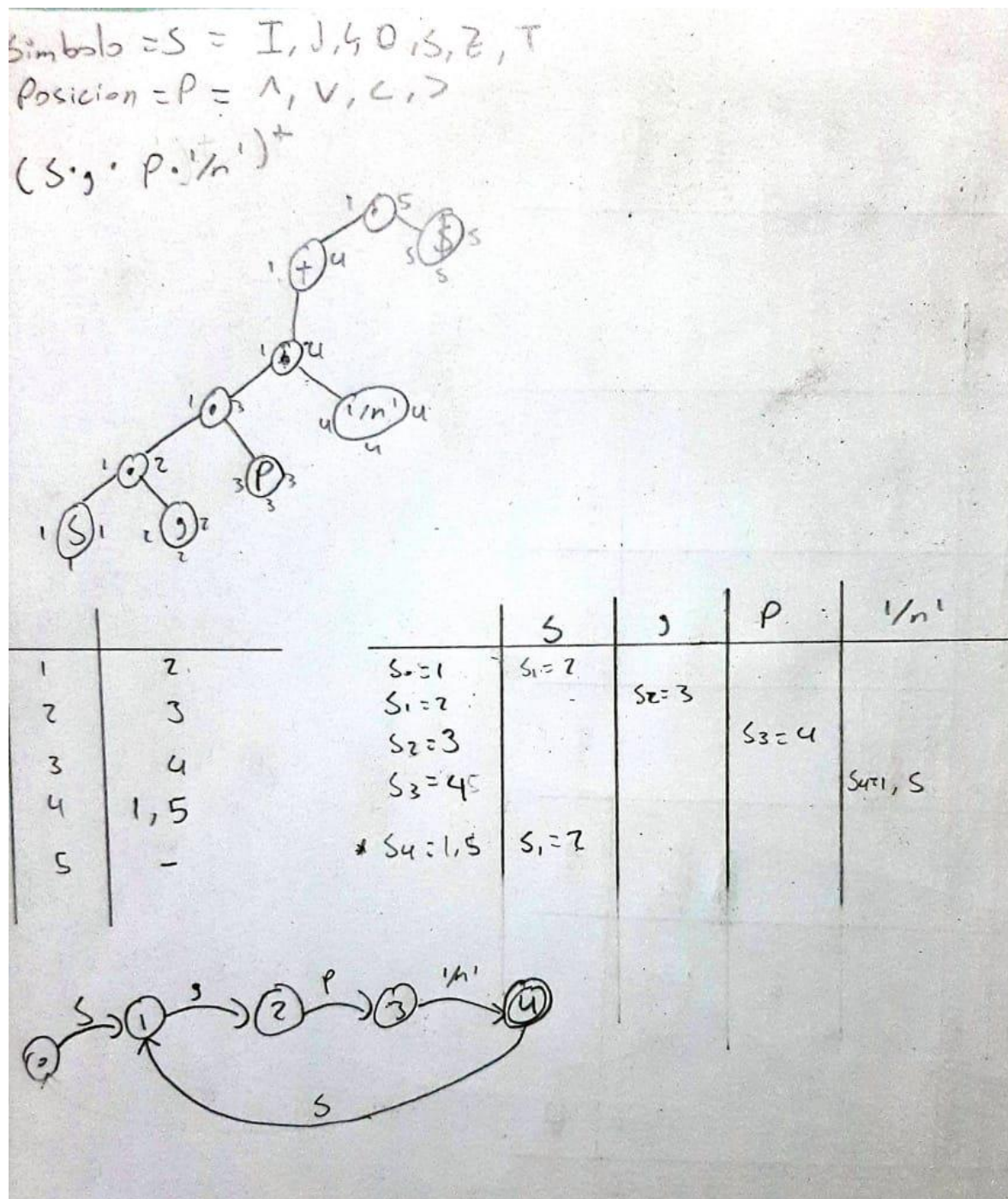
```

```

        break;
    case 10:
        if((caracter>64&&caracter<91)||((caracter>96&&caracter<123)){
            cadconcat+=String.valueOf(caracteractual);
            estado=10;
        }else if(caracter==95){
            cadconcat+=String.valueOf(caracteractual);
            estado=10;
        }else if(caracter>47&&caracter<58){
            cadconcat+=String.valueOf(caracteractual);
            estado=10;
        }else if(caracter==10||caracter==13){
            id=cadconcat;
            linea++;
            columna=0;
            tokens.add(new Tokens(linea,columna,"Token Identificador",id));
            estado=0;
            cadconcat="";
        }else{
            Errores.add("Caracter erroneo: "+caracteractual+" Linea: "+linea+" Columna: "+columna);
        }
        break;
    case 11:
        if(caracter==47){
            cadconcat+=String.valueOf(caracteractual);
            estado=12;
        }else{
            Errores.add("Caracter erroneo: "+caracteractual+" Linea: "+linea+" Columna: "+columna);
        }
        break;
    case 12:
        cadconcat+=String.valueOf(caracteractual);
        estado=13;
        break;
    case 13:
        if(caracter==10||caracter==13){
            tokens.add(new Tokens(linea,columna,"Comentario Unilinea",cadconcat));
            linea++;
            columna=0;
            estado=0;
            cadconcat="";
        }else{
            cadconcat+=String.valueOf(caracteractual);
            estado=13;
        }
    }
}

```


En el caso del segundo analizador se empleó el siguiente árbol con el siguiente autómata de resultado.



Y realizando el mismo procedimiento del analizador anterior obtenemos la siguiente función.

```

public void AnalisisFiguras(String cadena) {
    int linea = 1;
    int columna = 0;
    String cadconcat="";
    int estado=0;
    int caracter =0;
    char caracteractual;
    String simbolo="";
    String posicion="";
    for(int i=0;i<cadena.length();i++){
        caracteractual=cadena.charAt(i);
        caracter=(int)caracteractual;
        columna++;
        switch(estado){
            case 0:
                if(caracter==73||caracter==74||caracter==76||caracter==79||caracter==83||caracter==84||caracter==90){
                    simbolo=String.valueOf(caracteractual);
                    tokens.add(new Tokens(linea,columna,"Pieza Tetris",String.valueOf(simbolo)));
                    estado=1;
                }else{
                    Errores.add("Caracter erroneo en archivo 2:"+caracteractual+" Linea: "+linea+" Columna: "+columna);
                    System.out.println("Existe un error en los caracteres, case 0 archivo 2");
                    //estado=90;
                }
                break;
            case 1:
                if(caracter==32){
                    estado=1;
                }else if(caracter==44){
                    tokens.add(new Tokens(linea,columna,"Coma",","));
                    estado=2;
                }else{
                    Errores.add("Caracter erroneo en archivo 2: "+caracteractual+" Linea: "+linea+" Columna: "+columna);
                    System.out.println("Existe un error en los caracteres, case 1 archivo 2");
                    //estado=90;
                }
                break;
            case 2:
                if(caracter==94||caracter==118||caracter==60||caracter==62){
                    posicion=String.valueOf(caracteractual);
                    tokens.add(new Tokens(linea,columna,"Posicion Pieza",String.valueOf(posicion)));
                    estado=3;
                }else if(caracter==32){
                    estado=2;
                }
                break;

```

```

            case 3:
                if (cadena.charAt(i)=='\r' || cadena.charAt(i+1)=='\n'){
                    figuras.add(new Figuras(simbolo,posicion));
                    estado=4;
                    linea++;
                    columna=0;
                }else{
                    Errores.add("Caracter erroneo en archivo 2: "+caracteractual+" Linea: "+linea+" Columna: "+columna);
                    System.out.println("Existe un error en los caracteres, case 3 archivo 2");
                    //estado=90;
                }
                break;
            case 4:
                if (caracter==13||caracter==10){
                    estado=0;
                }else if(caracter==73||caracter==74||caracter==76||caracter==79||caracter==83||caracter==84||caracter==90){
                    simbolo=String.valueOf(caracteractual);
                    tokens.add(new Tokens(linea,columna,"Pieza Tetris",String.valueOf(simbolo)));
                    estado=1;
                }else{
                    Errores.add("Caracter erroneo en archivo 2: "+caracteractual+" Linea: "+linea+" Columna: "+columna);
                    System.out.println("Existe un error en los caracteres, case 4 archivo 2");
                    //estado=90;
                }
                break;
        }
    }
}

```

Los cuales ya aceptan los archivos de entrada que se vaya a ingresar al programa.