```
//3 Métodos para la implementación de la PILA.
void M_Agregar_Elemento(char Informacion){
       Nodo_Actual= Nuevo_Nodo;
}

void M_Quitar_T_Pila(){
       Quitar = Nodo_Actual;
}


char M_Mostrar_Tope_Pila(){
       return Dato;
}

String M_tabla(int x,int y){
       String tabla[][]= new String [5][6];
         for(int i=0; i<5;i++){
               for(int j=0; j<6;j++){
                 tabla[i][j]="#";    //LA TABLA SE LLENA CON # PARA DESPUES SER LLENADA CON
                                     //VALORES SEGÚN GRAMATICA
               }
         }
         tabla[0][0]="eT"; //El símbolo "*" hace referencia a la cadena vacía dentro de la tabla
         tabla[0][3]="eT";
         tabla[1][1]="eT+";
         tabla[1][4]="*";
         tabla[1][5]="*";
         tabla[2][0]="tF";
         tabla[2][3]="tF";
         tabla[3][1]="*";
         tabla[3][2]="tF*";
         tabla[3][4]="*";
         tabla[3][5]="*";
         tabla[4][0]="d";
         tabla[4][3]=")E(";
       return tabla[x][y];
}

public int M_terminales(char caracter){ //METODO QUE EVALUA LOS TERMINALES DE LA GRAMATICA
        int valor=0;
                    switch(caracter){
                          case 'd': valor=0; break;
                          case '+': valor=1; break;
                          case '*': valor=2; break;
                          case '(': valor=3;  break;
                          case ')': valor=4; break;
                          case '$': valor=5; break;
                          default: valor=9; break;
                    }
       return valor;
}

int M_Noterminales(char caracter){ //METODO QUE EVALUA LOS NO TERMINALES DE LA GRAMATICA
//los caracteres en minúscula hacen referencia a los NO TERMINALES, es decir: E', y T'
       int valor=0;
                    switch(caracter){
                          case 'E': valor=0; break;
                          case 'e': valor=1; break;
                          case 'T': valor=2; break;
                          case 't': valor=3;  break;
                          case 'F': valor=4; break;
                          default: valor=9; break;
                    }
       return valor;
}
```

```
M_Analizdor_Cadena(){
       int  Numero_caracter, x,y;
       char pila=' ';
       char caracter=' ';
       boolean ban=true;
       M_Agregar_Elemento('$'); //SE AGREGA A PILA EL SIMBOLO $
       M_Agregar_Elemento('E');  //SE AGREGA A PILA LA VARIABLE INICIAL
      FileInputStream Lectura = new FileInputStream("d:/Prueba.txt");
              while((Numero_caracter = Lectura.read())!=-1){
                        pila=M_Mostrar_Tope_Pila();
                        y=M_terminales(caracter);
                        if(y==9){
                               "Error…
                            do{
                                    x=M_Noterminales(pila);
                                           if(x==9){
                                                 "Error…"
                                                 ban=false;
                                           cadena=M_tabla(x,y);
                                           if(cadena.equals("*")){
                                              M_Quitar_T_Pila();
                                           }else{
                                                 M_Quitar_T_Pila();
                                                 char letras[]
                                                       for (int i=0;i<cadena.length();i++)      {
                                                                    M_Agregar_Elemento(letras[i]);
                                                                  }
                                                              }
                                                                    pila=M_Mostrar_Tope_Pila();

                                                                }while(caracter!=pila);
                                                              M_Quitar_T_Pila();
                                                 if(ban==false){
                                                       break;
                                                 }
                                    }

                                    pila=M_Mostrar_Tope_Pila();
                                           if(caracter=='$' && pila==' ') {
                                           "Cadena aceptada"
                                    }else{
                                           "Error …
                                   }
              }
       }

       ////GRAMATICA QUE REPRESENTA EXPRESIONES ARITMETICAS SENCILLAS.
        G(L):
              E → TE´
              E'→ +TE´| cadena vacía
              T → FT´
              T´→ *FT´| cadena vacía
              F → (E) | id
```