

1. Todo el archivo se asume está correctamente escrito y sigue las reglas que a continuación se describen.
2. Se inicia y finaliza el programa con un 'Inicio' y un 'Pare' respectivamente.
3. Ninguna línea tendrá espacios en blanco al inicio. Todas las líneas empiezan un carácter que sería una letra.
4. Los comandos 'inicio', 'pare', 'lea', 'esc', 'para', 'fpara', 'si', 'fsi',, pueden ser escritos en mayúsculas, minúsculas, o la primera letra en mayúscula y el resto en minúscula y el programa debe de igual forma funcionar en cualquier caso.
5. No habrá una línea que inicie con un 'lea' y que tenga varias variables a ser leída, como el caso de "lea a, b, c". Para usar los "lea", será una variable por línea. Es decir que se tendrían en el ejemplo anterior 3 líneas de "lea", "lea a", "lea b" y "lea c".
6. Lo anterior se aplica de igual forma para el comando "esc".
7. Las líneas que inician con un "para" tendrán la siguiente estructura: "para var=inicio,fin,incremento". Se inicia con la palabra "para", se continúa con un espacio en blanco, luego el nombre de la variable y sin dejar espacio en blanco se coloca el símbolo de igual "=", luego se colocan en orden y separados solamente por comas el valor de inicio, el valor de fin y el incremento.
8. El incremento o decremento de un ciclo para debe iniciar con un símbolo de "+" o de "-".
9. Un ejemplo de una línea de "para" válida sería: "para variable=1,10,+2".
10. Para finalizar un ciclo para, se usa la instrucción "fpara".
11. Las líneas que inician con "si" serán las condicionales. Tienen la estructura de "si (expresión)". Se inicia con la palabra "si", seguido de un espacio, y luego entre paréntesis, la expresión a evaluar.
12. En las líneas de "si", la expresión a evaluar que están dentro de paréntesis solo tiene una expresión booleana a evaluar. Ejemplo: "si (A<B)", sería una expresión válida, mientras que "si (A<B Y B>C)" no sería una expresión válida.
13. Para hacer múltiples evaluaciones en una línea de "si", se debe escribir de la forma: "si (A<B) Y (B>C)"
14. No habrá más de dos evaluaciones juntas en un solo condicional, es decir, no habrá algo como "(A<B) Y (B>C) o (D ==C)"
15. Para evaluar la igualdad en una condicional, se usará doble igual "==".
16. Las líneas que contienen "sino" serán independientes, no tendrán nada más escrito.
17. Las condicionales solo tendrán "si", "fsi", y posiblemente en algunos casos "sino".
18. Para finalizar una condicional, se usa la instrucción "Fsi".
19. No puede haber un ciclo dentro de una condicional.
20. Solo habrá máximo 2 tres ciclos "para" anidados.
21. No habrá condicionales anidados.
22. En caso de tener dos ciclos anidados, a lo máximo existirá una condicional dentro del segundo ciclo "para".
23. No habrá un ciclo para dentro de una condicional.
24. Dentro del condicional o "si" no habrá ciclos o condicionales, solo instrucciones "simples".
25. Los casos que existirán serán:
 - 25.1. Para, para, si, fsi, fpara, fpara.
 - 25.2. Para, si, fsi, fpara.
 - 25.3. Para, para, fpara, fpara.

- 25.4. Para, si, fsi, fpara, para, si, fsi, fpara.
- 25.5. Para, si, fsi, fpara, para, fpara.
- 25.6. Para, para, si, fsi, fpara, fpara, para, si, fsi, fpara.
- 26. Donde esté un condicional, pueden haber solo dos casos:
 - 26.1. Si, ffsi.
 - 26.2. Si, sino, fsi.
- 27. En un caso donde existe un “si”, “sino” y “fsi”, existirá un bloque1 entre el “si” y el “sino” y un bloque2 entre el “sino” y el “fsi”, prevalece el que tenga más instrucciones entre los dos bloques.
- 28. Lista de palabras reservadas:
 - 28.1. Inicio
 - 28.2. Pare
 - 28.3. Si
 - 28.4. Sino
 - 28.5. Fsi
 - 28.6. Para
 - 28.7. Fpara
 - 28.8. Lea
 - 28.9. Esc
 - 28.10. Y
 - 28.11. O
 - 28.12. \neg
 - 28.13. $==$