**Nombre:** Kelly Marie Aguilar Andino

**No. de Cuenta:** 20151020008

**Ejercicio 1:** BNF y autómata de gramática que reconoce:

* Declarar variables de tipo primitivo
* Ciclo while
* Operaciones aritméticas
* Imprimir el contenido de variables
* Ingresar contenido a variables

**Código de Ejemplo**

inicio{

int variable2 = variable+4;

int variable = "dfdf";

string nombre;

while(variable>variable2)

{

int x=4;

out("hola");

nombre = in();

}

}fin

**Notación BNF**

<programa> ::= ‘**inicio {’** <bloque\_programa> ‘**} fin’**

<bloque\_programa> ::= <sentencia> |

<bloque\_programa><sentencia>

<sentencia> ::= <declaración><delimitador> |

<asignación><delimitador> | <bucle> |

<impresión><delimitador> | <lectura><delimitador>

<declaración> ::= <tipo\_dato> <identificador> <op\_asignación> <operando>

<identificador> ::= <letra> | <identificador><letra> | <identificador><digito>

<asignación> ::= <identificador><op\_asignación><exp\_matematica> |

<identificador><op\_asignación><cadena>

<cadena> ::= <palabra> | <cadena><palabra>

<palabra> ::= <letra> | <palabra><letra> | <palabra><carácter\_especial>

<exp \_ matematica > ::= <operando> |

< exp\_matematica> <op\_aritmético> <operando>

<operando> ::= <identificador> | <numero\_real> | <cadena>

<numero\_real> ::= <signo><entero> | <signo><decimal>

<entero> ::= <digito> | <entero><digito>

<decimal> ::= < entero >’**.**’ < entero >

<condición> ::= <exp \_ matematica><op\_comparación><exp \_ matematica> |

<identificador><op\_comparación><cadena> |

<condición><op\_conjuncion><condición>

<bucle> ::= <ciclo\_while>

<ciclo\_ while> ::= ‘**while(**’<condición> ’**){**’ <bloque\_programa> ‘**}**’

<impresión> ::= ‘**out(**’ <operando>’**)**’

<lectura> ::= <identificador><op\_asignación>’**in(**’<entrada>’**)**’

<entrada> ::= <cadena> | <numero\_real>

<carácter\_especial> ::= ‘ ‘ | ‘**,**’ | ‘**.**’ | ‘**;**’

<op\_asignación> ::= ‘**=**’

<tipo\_dato> ::= ‘**int**’ | ’**char**’

<signo> ::= ‘**+**’ | ‘**-**’ | ‘ ‘

<op\_aritmético> ::= ‘**+**’ | ‘**-**’ | ‘**\***’ | ‘**/**’

<digito> :: = ‘**0**’ | ’**1**’ | ’**2**’ | ’**3**’ | ’**4**’ | ’**5**’ | ’**6**’ | ’**7**’ | ’**8**’ | ’**9**’

<letra> ::=‘**a**’ | ’**b**’ | ’**c**’ | ’**d**’ | ’**e**’ | ’**f**’ | ’**g**’ | ’**h**’ | ’**i**’ | ’**j**’ | ’**k**’ | ’l’ | ’**m**’ | ’**n**’ | ’**ñ**’ |

’**o**’ | ’**p**’ | ’**q**’ | ’**r**’ | ’**s**’ | ’**t**’ | ’**u**’ | ’**v**’ | ’**w**’ | ’**x**’ | ’**y**’ | ’**z**’ | ’**A**’ | ’**B**’ | ’**C**’ | ’**D**’ |

’**E**’ | ’**F**’ | ’**G**’ | ’**H**’ | ’**I**’ | ’**J**’ | ’**K**’ | ’**L**’ | ’**M**’ | ’**N**’ | ’**Ñ**’ | ’**O**’ | ’**P**’ | ’**Q**’ | ’**R**’ |

’**S**’ | ’**T**’ | ’**U**’ | ’**V**’ | ’**W**’ | ’**X**’ | ’**Y**’ | ’**Z**’

<op\_comparación> ::= ‘**==**‘ | ’**===**‘ | ’**>**‘ | ’**<**‘ | ’**>=**‘ | ’**<=**‘ | ’**!=**‘ | ‘**<>**’

<op\_conjuncion> ::= ‘**˄**’ | ‘**˅**’

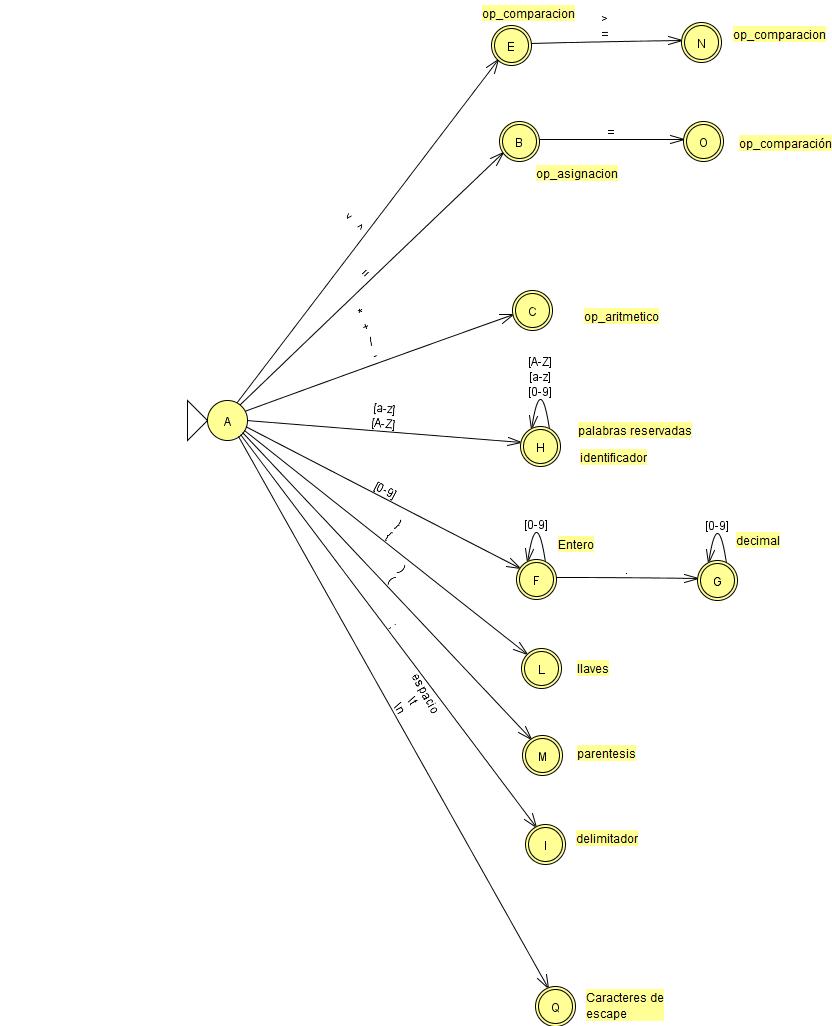
<delimitador> ::= ‘**;**’

**Autómata Programable**

Código disponible en GitHub:

https://github.com/Kelly97/Lenguajes\_prog/tree/master/Tareas/Ejercicios%20Examen%201/Ejercicio%201/Aut%C3%B3mata\_programado%20(c%2B%2B)

**Autómata Finito**

****