**Gramática LL1**

EQUIPO

ALUMNO: JUAN GERARDO BARRON HERNANDEZ MATRICULA: 33142377

ALUMNO: EMMANUEL DE JESUS VELASQUEZ MARTINEZ MATRICULA: 35161479

**Gramática**

N =

{

<Programa>, <Bloque>, <Condición>, <Proposición>, <Expresión>, <Termino>, <Factor>, <A>, <B>, <C>, <D>, <E>

}

T =

{

const, id, =, num, \, , ,; , var, proced, ==, !=, <, >, <=, >=, begin, end, write, read, call, if, then, while, do, dto, +, -, \*, /, (, ), .

}

P =

{

<Programa> ::= <Bloque>.

<Bloque> ::= (const <A> | Ԑ) ( var <B> | Ԑ ) <C> <Proposición>

<Proposición> ::= begin <D> end | id= < Expresión > | write (id |num) | read id | call id | if <Condicion> then < Proposición > | while <C> do <P> | for id = < Expresión > (to | dto) < Expresión > do <P>

<Expresión> ::= <Termino> + <Expresión> |<Termino> - <Expresión>| <Termino>

<Condición> ::= <Expresión> == < Expresión > | < Expresión > != <Expresión >|< Expresión > < < Expresión >| < Expresión > > < Expresión > | < Expresión > <= < Expresión > | < Expresión > >= < Expresión >

<Termino> ::= <Factor> | <Factor> \* <Termino> | <Factor> / <Termino>

<Factor> ::= ( <Expresión> ) | id | num

<A> ::= id = num <E>

<E> ::= ,<A> | ;

<B> ::= id , <B> | id ;

<C> ::= proced id ; <Bloque>; <C> | Ԑ

<D> ::= < Proposición > | < Proposición >; <D>

}

S = <Programa>

**Eliminar conflictos first/first**

<B> ::= id <F>

<F> ::= , <B> | ;

<Condición> ::= <Expresión> <G>

<G> ::= == <Expresión> | != <Expresión> | < <Expresión> | > <Expresión> | <= <Expresión> | >= <Expresión>

<E> ::= <Proposicion> <H>

<H> ::= Ԑ | ; <E>

<Expresión> ::= <Termino> <I>

<I> ::= + <Expresión> | - <Expresión> | Ԑ

<Termino> ::= <Factor> <J>

<J> ::= Ԑ | \* <Termino> | / <Termino>

**Gramática final**

N =

{

<Programa>, <Bloque>, <Condición>, <Proposición>, <Expresión>, <Termino>, <Factor>, <A>, <B>, <C>, <D>, <E>, <F>, <G>, <H>, <I>, <J>

}

T =

{

const, id, = , num, \, , , ; , var, proced, ==, !=, <, >, <=, >=, begin, end, write, read, call, if, then, while, do, dto, +, -, \*, /, (, ), .

}

P =

{

<Programa> ::= <Bloque>.

<Bloque> ::= (const <A> | Ԑ) ( var <B> | Ԑ ) <C> <Proposición>

<Condición> ::= <Expresión> <G>

<Proposición> ::= begin <E> end | id= <Expresión> | write (id | num) | read id | call id | if <Condición> then <Proposición> | while <Condición> do <Proposición> | for id = <Expresión> (to | dto) <Expresión> do <Proposición>

<Expresion> ::= <Termino> <I>

<Termino> ::= <Factor> <J>

<Factor> ::= ( <Expresión> ) | id | num

<A> ::= id = num <D>

<B> ::= id <F>

<C> ::= proced id; <Bloque> ; <C> | Ԑ

<D> ::= , <A> | ;

<E> ::= <Proposición> <H>

<F> ::= , <B> | ;

<G> ::= == <Expresión> | != <Expresión> | < <Expresión> | > <Expresión> | <= <Expresión> | >= <Expresión>

<H> ::= Ԑ | ; <E>

<I> ::= +<Expresión> | -<Expresión> | Ԑ

<J> ::= Ԑ | \*<Termino> | /<Termino>

}

S= <Programa>

**First y Followns de la Gramática**

|  |  |  |
| --- | --- | --- |
|  | **First** | **Followns** |
| **<Programa>** | First(<Programa>)= First(<Bloque>) ={const, var, proced, begin, id, write, read, call, if, while, for} | Follows(<Programa>)={$} |
| **<Bloque>** | First(<Bloque>)= {const} + {var} + (First(<A>)-Ԑ) + First(<Proposicion>) ={ const, var, proced, begin, id, write, read, call, if, while, for} | Follows(<Bloque>)={ \. , \;} |
| **<Condicion>** | First(<Condición>)= (First(<Expresión>) ={ (, id, num} | Follows(<Condición>) = {then, do} |
| **<Proposicion>** | First(<Proposición>)={begin, id, write, read, call, if, while, for} | Follows(<Proposición>)=Follows(<Bloque>) + Follows(<Proposición>) + (First(<H>) - Ԑ) + Follows(<E>) = {\. , end } |
| **<Expresion>** | First(<Expresión>)= First(<Termino>)={ (, id, num} | Follows(<Expresion>)=First (G)+ Follows(G) + Follows(<Proposición>) + {to, dto, do} +Follows(<I>) +{ ) } = { ==, !=, <, >, <=, >=, then, do, \. , end , to, dto, ) } |
| **<Termino>** | First(<Termino>)= First(<Factor>) ={ (, id, num} | Follows(<Termino>)= (First(<I>)-Ԑ) + Follows(<Expresión>) + Follows(<J>) = {+, - , ==, !=, <, >, <=, >=, then, do, \. , \; , end , to, dto, ) } |
| **<Factor>** | First(<Factor>)={(, id, num} | Follows(<Factor>) = (First(<J>)-Ԑ) + Follows(<Termino>) ={ \*, /,+, - , ==, !=, <, >, <=, >=, then, do, \. , \; , end , to, dto, ) } |
| **<A>** | First(<A>) ={id} | Follows(<A>)= {var} + (First (<C>) -Ԑ)+ First (<Proposición>) + Follows(<D>)= {var, proced, begin, id, write, read, call, if, while, for } |
| **<B>** | First(<B>)= {id} | Follows(<B>)= Follows(<F>) + (First (<C>) -Ԑ)+ First (<Proposición>) ={ proced, begin, id, write, read, call, if, while, for} |
| **<C>** | First(<C>)= {proced, Ԑ} | Follows(<C>)= First (<Proposición>) +Follows(<C>) = { begin, id, write, read, call, if, while, for} |
| **<D>** | First(<D>) ={\, , \;} | Follows(<D>)=Follows(<A>) = { var, proced, begin, id, write, read, call, if, while, for} |
| **<E>** | First(<E>)= First(<Proposición>) ={ begin, id, write, read, call, if, while, for} | Follows(<E>)={end} + Follows(<H>) = {end} |
| **<F>** | First(<F>)= {\, , \;} | Follows(<F>)=Follows(<B>)={ proced, begin, id, write, read, call, if, while, for} |
| **<G>** | First(<G>)= {==,!=,<,>,<=,>=} | Follows(<G>)=Follows(<Condición>)= {then, do } |
| **<H>** | First(<H>)= {Ԑ, \;} | Follows(<H>)= Follows(<E>)= {end} |
| **<I>** | First(<I>)= { +, -, Ԑ} | Follows(<I>)= Follows(<Expresión>) = { ==, !=, <, >, <=, >=, then, do, \. , \; , end , to, dto, ) } |
| **<J>** | First(<J>)= {Ԑ , \*, / } | Follows(<J>)= Follows(<Termino>) ={ +, - , ==, !=, <, >, <=, >=, then, do, \. , \; , end , to, dto, ) } |