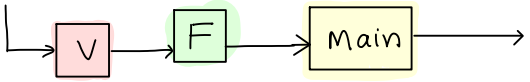
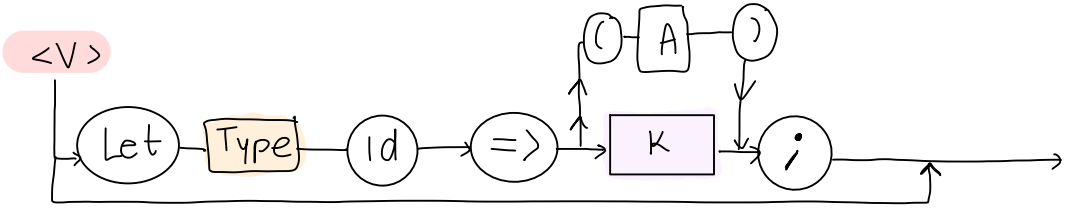
Kenia Gabriela Sanchez Duran
A01244940

# DIAGRAMAS DE SINTAXIS

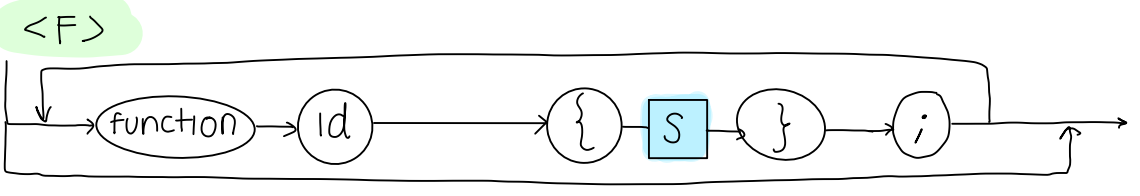
## <programa>



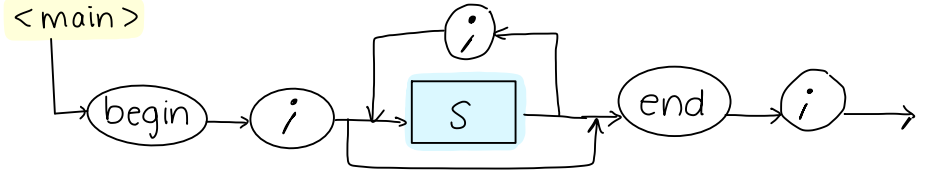

## <V>



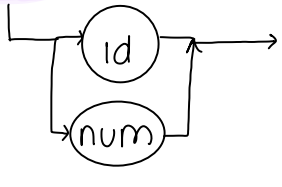

## <F>



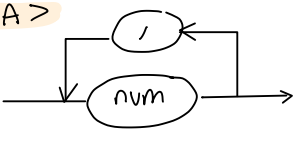

## <main>



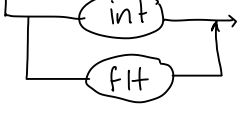

## <K>

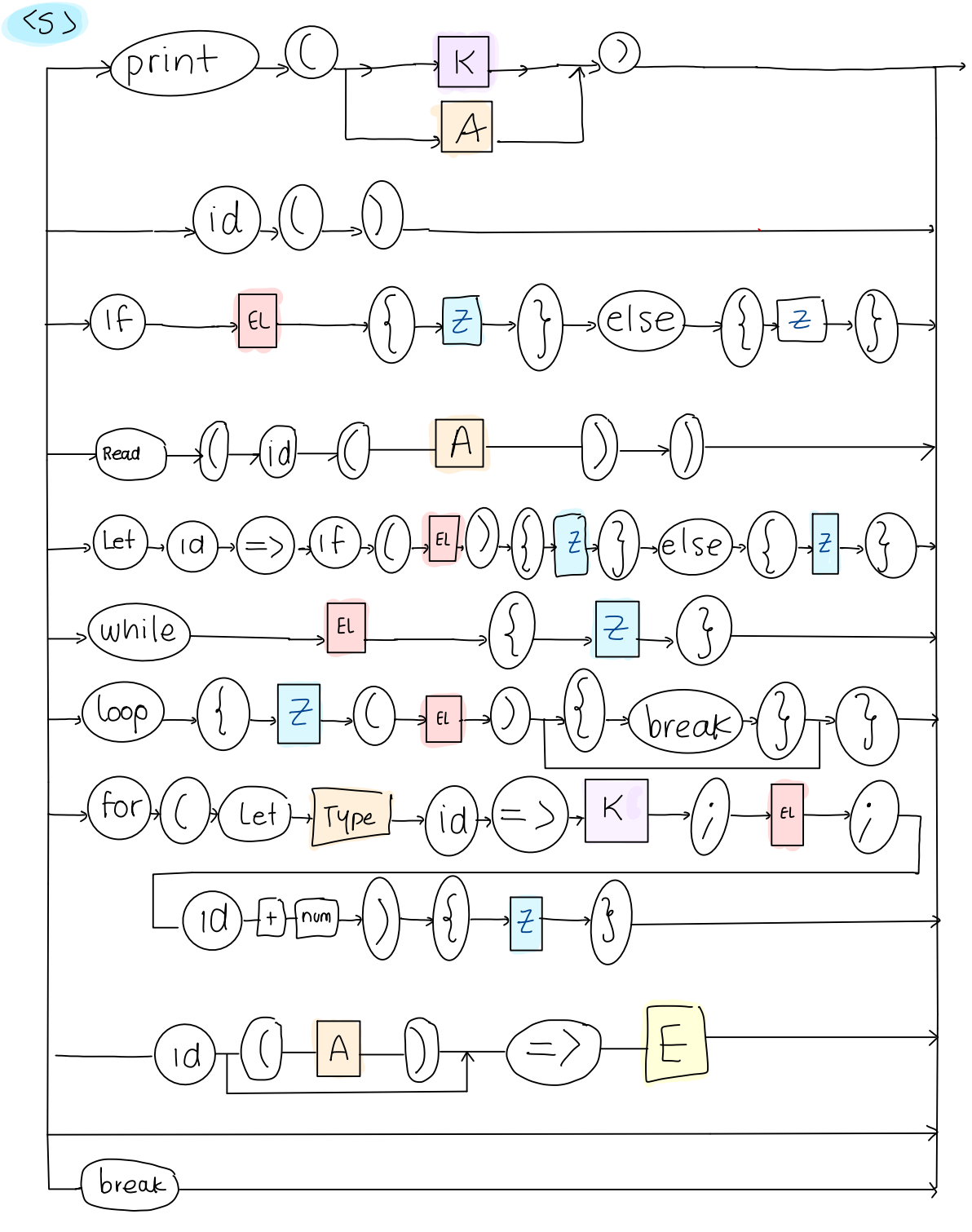



## <A>

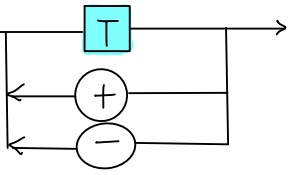



## <Type>

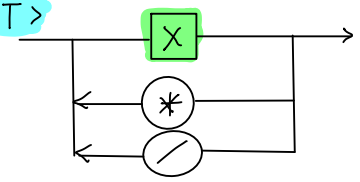



<S>
print
(
K
A
)
id
(
)
if
EL
{
Z
}
else
{
Z
}
Read
(
id
(
A
)
)
Let
id
=>
if
(
EL
)
{
Z
}
else
{
Z
}
while
EL
{
Z
}
loop
{
Z
}
(
EL
)
{
break
}
}
for
(
Let
Type
id
=>
K
;
EL
;
id
+
num
)
{
Z
}
id
(
A
)
=>
E
break

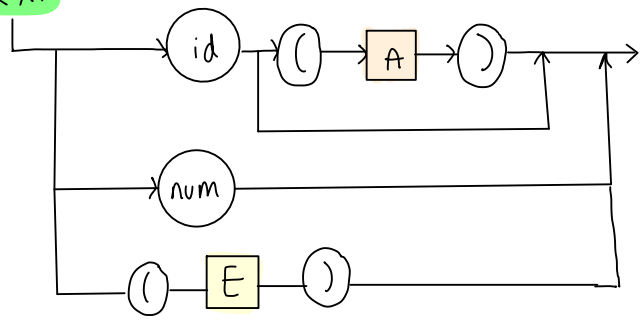
<E>



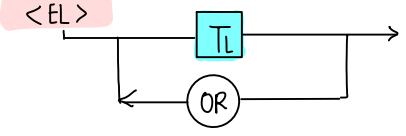

<T>



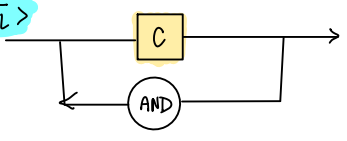

<X>



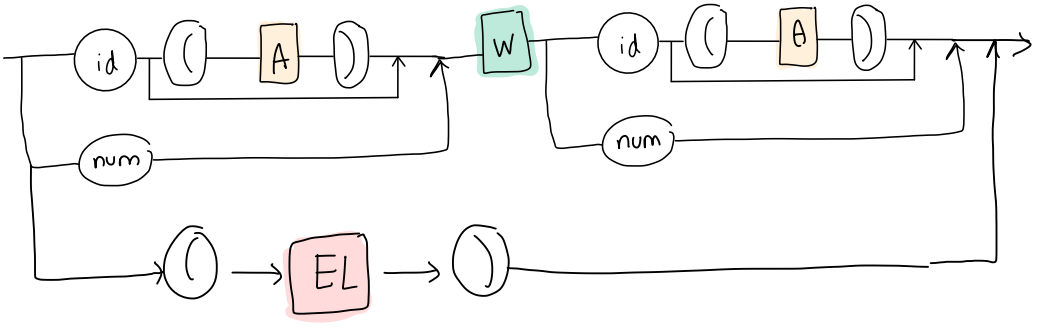

<EL>



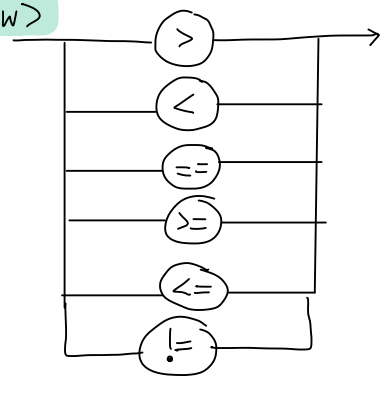

<TL>




<C>

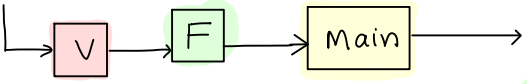



<W>



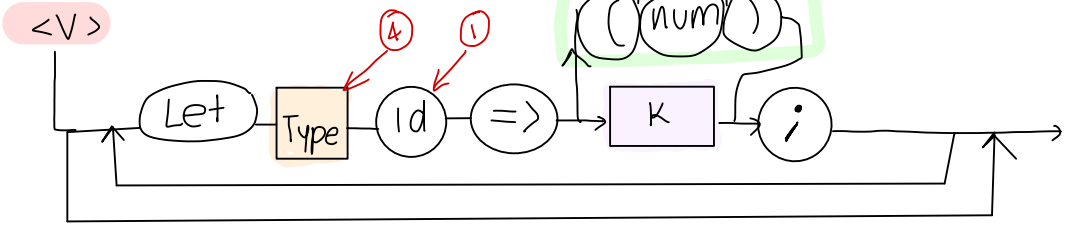
# GRAMÁTICA

## < programa >




Programa → V F Main

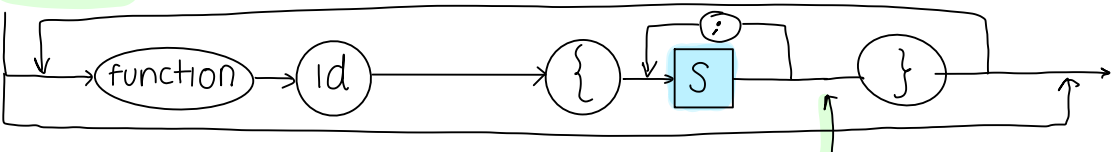
## <V>




V → V Let type id ⇒ K ;  V → ε  A → num

V → V Let type id ⇒ ( A)  A → A, num

## <F>




F → F function id { Z }

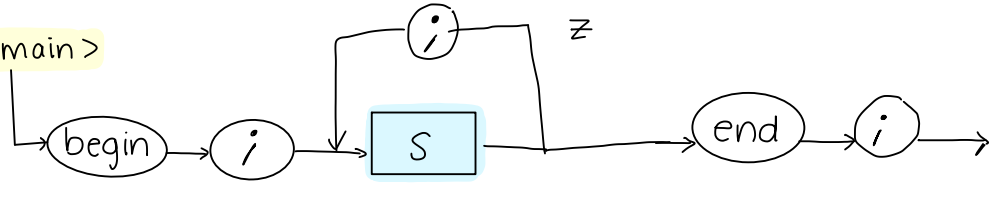
F → ε

Z → S ;

Z → Z S ;

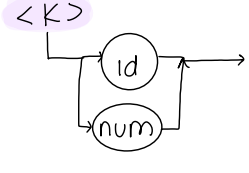
Z

## <main>




Main → begin ; Z end ;

## <K>



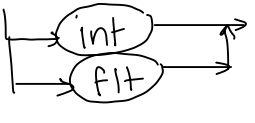

K → id

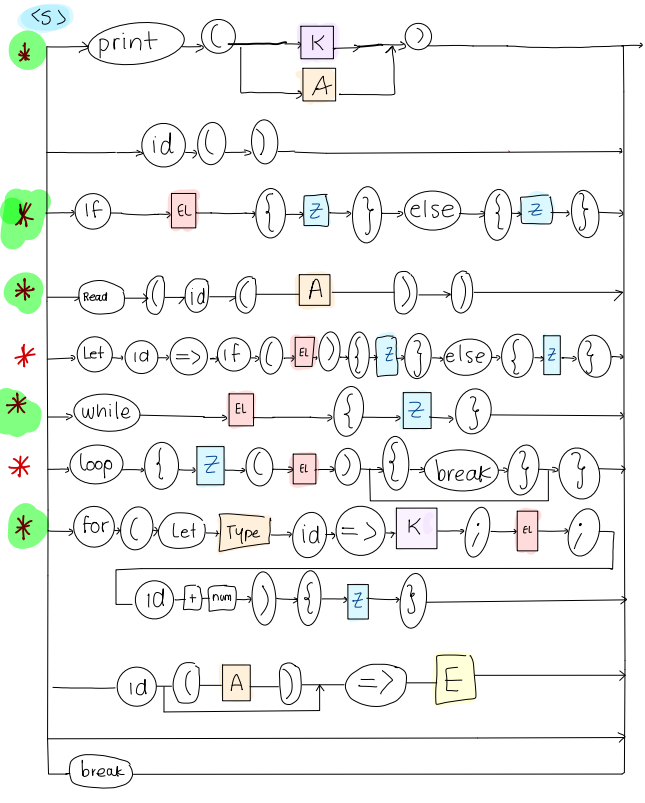
K → num

## < Type >

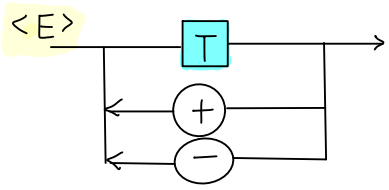
Type → int

Type → flt

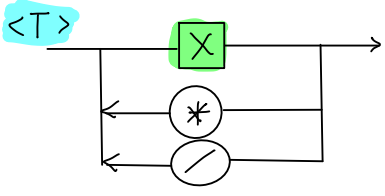




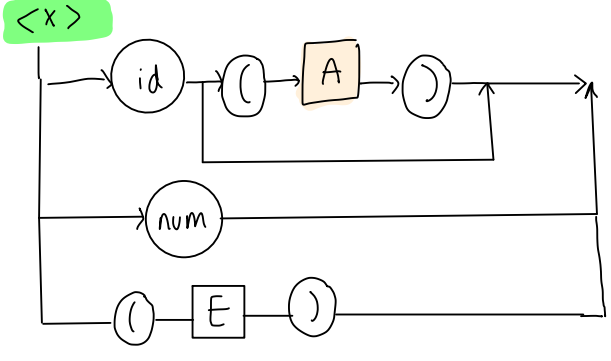

- S→ print ( M)
  M→ "K"
  M → A
- S→ id ( )
- S→ for ( Let Type id ⇒ K ; EL ; id + num) {Z}
- S→ if EL { Z } else { Z }
- Read( id(A))
- Let id ⇒ if (EL) { Z } else { Z }
- while EL { Z }
- Loop { Z (EL) B }
  B → {break}
  B → ε
- S→ id D ⇒ E
  D → ε
  D → (A)
- S → ε
- S→ Break



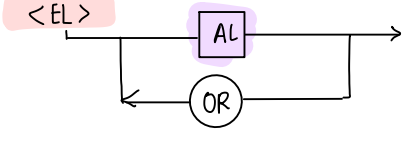

E → E + X (2)
E → E − X (2)
E → X



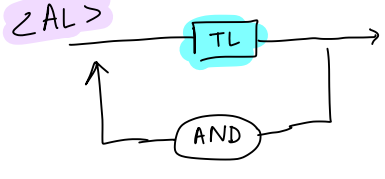

T → T * F (3)
T → T / F (3)
T → F




(1)
F → id
F → id ( A )
F → num
F → ( E )

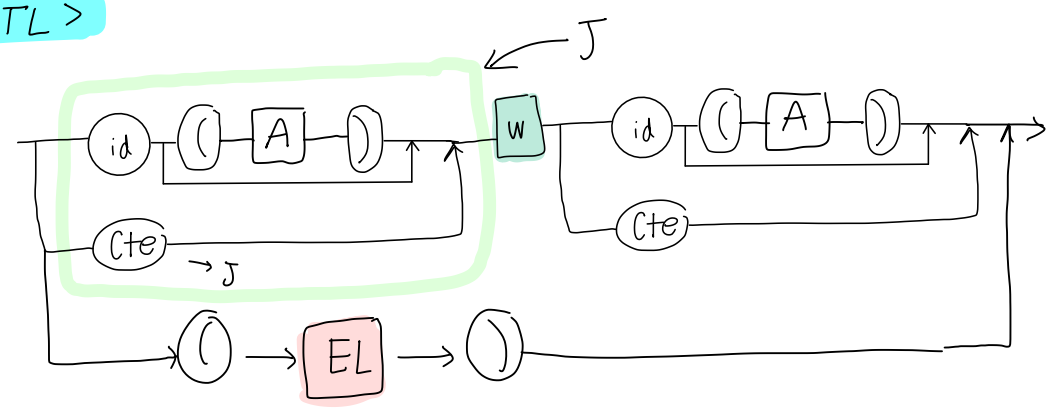



EL → AL
EL → EL or AL (2)




AL → TL
AL → AL AND TL (2)

<TL>

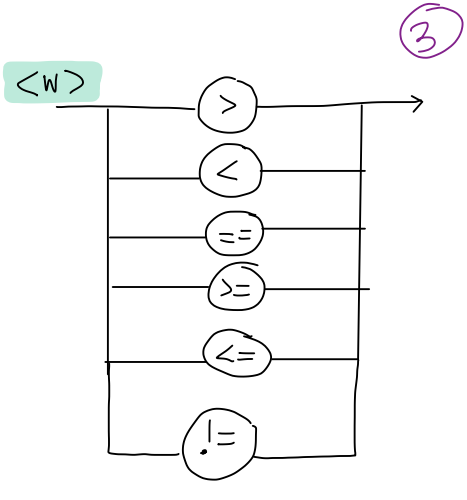



TL → ( EL )

TL → J W J

J → id ⟶ ①

J → id ( A )




X → >

X → <

X → ==

X → >=

X → <=

X → !=

① Push pila de operandos

② Generar cuadruplo ( ):

op2 ⇒ pop pila de operandos

op1 ⇒ pop pila de operandos

result = Temp avail

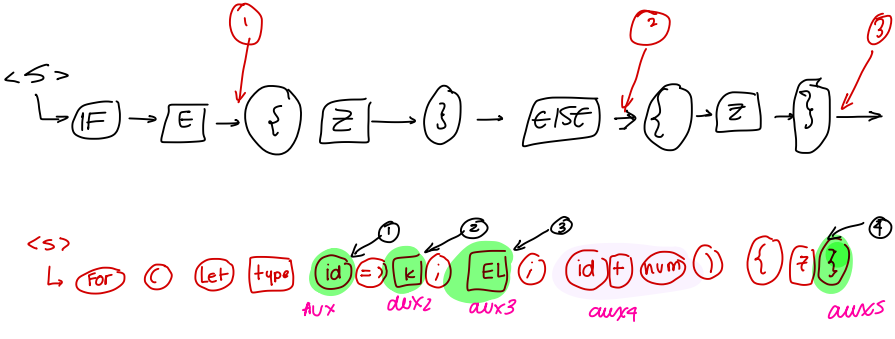
regresar el result a pila operandos

agregar result al stack cuadruplos con "+", op1, op2, result

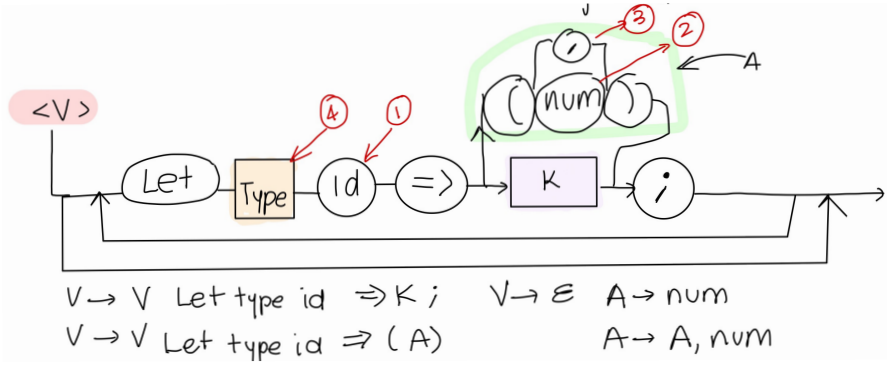
③ lo mismo que el 2 pero con *.

En el logico es lo mismo
pero con AND /OR/ comparadores

- CICLOS -




- INTENTO DE VAR DIM -




Let int id => ( 4, 4 );