****

1. **Universidad Tecnológica Nacional**

**Facultad Regional Concepción del Uruguay**

1. **Ingeniería en Sistemas de Información**

|  |
| --- |
| 1. **Sintaxis Y Semántica de los Lenguajes** |

1. **ChePas**
2. **Integrantes:**

Basso, Santiago

Bentancour, Felipe

Haffner, Ignacio

Wulfsohn, Yaco

1. **Docentes:**

Ing. Andrés Pascal

Ing. Claudia Álvarez

**Año 2022**

**Introducción:**

ChePas es un lenguaje de programación con muchas similitudes a el lenguaje pascal, consta de un cuerpo con varias sentencias entre las cuales tenemos la escritura, lectura, asignación, declaración de variables, estructura de control (if) y estructura cíclica (while), estas dos últimas pudiendo ser utilizadas con una condición la cual puede ser compleja, teniendo varios OR y AND combinados. ChePas también cuenta con expresiones aritméticas las cuales pueden ser operaciones combinadas utilizando los operadores siguientes: suma(+), resta(-), multiplicación(\*), división(/), potencia(^) y raíz cuadrada(raíz(valor)), estas respetan la prioridad que tiene cada operación sobre otra.

**¿Por qué ChePas?**

El nombre ChePas fue elegido por el grupo ya que la idea fue realizar un lenguaje con algunas palabras en español, utilizamos el “Che” porque los argentinos y uruguayos tienen como costumbre decir che para referirse a alguien, el origen del termino che proviene de el idioma mapuche y significa persona o gente, como dato de color la palabra mapu-che significa “gente de la tierra”. Y le pusimos “Pas” por el lenguaje donde lo programamos y donde nos basamos que es Pascal

**Gramática LL(1)**

Cuerpo -> inicio MultiSent fin

MultiSent -> Sentencia ; MultiSent | epsilon

Sentencia -> Asignacion | Lectura | Escritura | VarCondicional | VarMientras | Variables

Variables -> var inicio ListaVariables fin

ListaVariables -> id MultiVar

MultiVar-> id MultiVar | epsilon

Asignacion -> id := ExpArit

Lectura -> leer (texto, id)

Escritura -> escribir (texto, ExpArit )

Numero -> id | (ExpArit) | const | - Numero

Condicion -> Parte1Cond Disyuncion

Parte1Cond -> Negacion Conjuncion

Negacion -> not CondRel | CondRel

Disyuncion -> or Parte1Cond Disyuncion | epsilon

Conjuncion -> and Parte1Cond Conjuncion | epsilon

CondRel -> ExpArit OperadorRelacional ExpArit | [Condicion]

VarCondicional -> si Condicion entonces Cuerpo SINO

SINO -> sino Cuerpo | epsilon

VarMientras -> mientras Condicion hacer Cuerpo

ExpArit -> Op1 SumaRes

Op1 -> RaizPot MultDiv

SumaRes -> + Op1 SumaRes| - Op1 SumaRes | epsilon

RaizPot -> Numero Pot | raiz(RaizPot)

Pot -> ^ RaizPot | epsilon

MultDiv -> \* RaizPot MultDiv | / RaizPot MultDiv | épsilon

**Forma Normal Backus Naur**

<Cuerpo> ::= “inicio” <MultiSent> “fin”

<MultiSent> ::= <Sentencia> “;” <MultiSent> “fin”

<Sentencia> ::= <Asignacion> | <Lectura> | <Escritura> | <VarCondicional> | <VarMientras> | <Variables>

<Variables> ::= “var” “inicio” <ListaVariables> “fin”

<ListaVariables> ::= “id” <MultiVar>

<Asignacion> ::= “id” “:=” <ExpArit>

<Lectura> ::= “leer” “(“ “texto” “,” “id” “)”

<Escritura> ::= “escribir” “(“ “texto” “,” “ExpArit” “)”

<Numero> ::= “id” | “(“<ExpArit>”)” | “const” | “-“ <Numero>

<Condicion> ::= <Parte1Cond> <Disyuncion>

<Parte1Cond> ::= <Negacion> <Conjuncion>

<Negacion> ::= “not” <CondRel> | <CondRel>

<Disyuncion> ::= “or” <Parte1Cond> <Disyuncion> | epsilon

<Conjuncion> ::= “and” <Parte1Cond> <Conjuncion> | epsilon

<CondRel> ::= <ExpArit> <OperadorRelacional> <ExpArit> | [<Condicion>]

<VarCondicional> ::= “si” <Condicion> “entonces” <Cuerpo> <SINO>

<SINO> ::= “sino” <Cuerpo> | epsilon

<VarMientras> ::= “mientras” <Condicion> “hacer” <Cuerpo>

<ExpArit> ::= <Op1> <SumaRes>

<Op1> ::= <RaizPot> <MultDiv>

<SumaRes> ::= “+” <Op1> <SumaRes>| “-“ <Op1> <SumaRes> | epsilon

<RaizPot> ::= <Numero> <Pot> | “raiz” ”(“<RaizPot>”)”

<Pot> ::= “^” <RaizPot> | epsilon

<MultDiv> ::= “\*” <RaizPot> <MultDiv> | “/” <RaizPot> <MultDiv> | épsilon

**Operaciones para llegar a la TAS**

---------------------------------------------------------------------------------------------------------------------

Cuerpo -> inicio MultiSent fin

Primero(Cuerpo)->{inicio}

TAS[Cuerpo,inicio]->inicio MultiSent fin

---------------------------------------------------------------------------------------------------------------------

MultiSent -> Sentencia ; MultiSent | epsilon

Primero(MultiSent)->Primero(Sentencia)->Primero(Asignacion)->{id}

TAS[MultiSent,id]=Sentencia ; MultiSent

Primero(MultiSent)->Primero(Sentencia)->Primero(Lectura)->{leer}

TAS[MultiSent,leer]=Sentencia ; MultiSent

Primero(MultiSent)->Primero(Sentencia)->Primero(Escritura)->{escribir}

TAS[MultiSent,escribir]=Sentencia ; MultiSent

Primero(MultiSent)->Primero(Sentencia)->Primero(VarCondicional)->{si}

TAS[MultiSent,si]=Sentencia ; MultiSent

Primero(MultiSent)->Primero(Sentencia)->Primero(VarMientras)->{mientras}

TAS[MultiSent,mientras]=Sentencia ; MultiSent

Primero(MultiSent)->Primero(Sentencia)->Primero(Variables)->{var}

TAS[MultiSent,var]=Sentencia ; MultiSent

Siguiente(MultiSent)->{fin}

TAS[MultiSent,fin]=epsilon

---------------------------------------------------------------------------------------------------------------------

Sentencia -> Asignacion | Lectura | Escritura | VarCondicional | VarMientras | Variables

Primero(Sentencia)->Primero(Asignacion)->{id}

TAS[Sentencia,id]=Asignacion

Primero(Sentencia)->Primero(Lectura)->{leer}

TAS[Sentencia,leer]=Lectura

Primero(Sentencia)->Primero(Escritura)->{escribir}

TAS[Sentencia,escribir]=Escritura

Primero(Sentencia)->Primero(VarCondicional)->{si}

TAS[Sentencia,si]=VarCondicional

Primero(Sentencia)->Primero(VarMientras)->{mientras}

TAS[Sentencia,mientras]=VarMientras

Primero(Sentencia)->Primero(Variables)->{var}

TAS[Sentencia,var]=Variables

---------------------------------------------------------------------------------------------------------------------

Variables -> var inicio ListaVariables fin

Primero(Variables)->{var}

TAS[Variables,var]=var inicio ListaVariables fin

---------------------------------------------------------------------------------------------------------------------

ListaVariables -> id MultiVar

Primero(ListaVariables)->{id}

TAS[ListaVariables,id]=id MultiVar

---------------------------------------------------------------------------------------------------------------------

MultiVar-> id MultiVar | epsilon

Primero(MultiVar)->{id}

TAS[MultiVar,id]=id MultiVar

Siguiente(MultiVar)->Siguiente(ListaVariables)->{fin}

TAS[MultiVar,fin]=epsilon

---------------------------------------------------------------------------------------------------------------------

Asignacion -> id := ExpArit

Primero(Asignacion)->{id}

TAS[Asignacion,id]=id:=ExpArit

---------------------------------------------------------------------------------------------------------------------

Lectura -> leer (texto, id)

Primero(Lectura)->{leer}

TAS[Lectura,leer]=leer(texto,id);

---------------------------------------------------------------------------------------------------------------------

Escritura -> escribir (texto, ExpArit )

Primero(Escritura)->{escribir}

TAS[Escritura,escribir]=escribir (texto, ExpArit )

---------------------------------------------------------------------------------------------------------------------

Numero -> id | (ExpArit) | const | - Numero

Primero(Numero)->{id}

TAS[Numero,id]=id

Primero(Numero)->{(}

TAS[Numero,(]=(ExpArit)

Primero(Numero)->{const}

TAS[Numero,const]=const

Primero(Numero)->{-}

TAS[Numero,-]= - Numero

---------------------------------------------------------------------------------------------------------------------

Condicion -> Parte1Cond Disyuncion

Primero(Condicion)->Primero(Parte1Cond)->Primero(Negacion)->{not}

TAS[Condicion,not]=Parte1Cond Disyuncion

Primero(Condicion)->Primero(Parte1Cond)->Primero(Negacion)->Primero(CondRel)->Primero(ExpArit)->Primero(Op1)->Primero(RaizPot)->Primero(Numero)->{id}

TAS[Condicion,id]=Parte1Cond Disyuncion

Primero(Condicion)->Primero(Parte1Cond)->Primero(Negacion)->Primero(CondRel)->Primero(ExpArit)->Primero(Op1)->Primero(RaizPot)->Primero(Numero)->{(}

TAS[Condicion,(]=Parte1Cond Disyuncion

Primero(Condicion)->Primero(Parte1Cond)->Primero(Negacion)->Primero(CondRel)->Primero(ExpArit)->Primero(Op1)->Primero(RaizPot)->Primero(Numero)->{const}

TAS[Condicion,const]=Parte1Cond Disyuncion

Primero(Condicion)->Primero(Parte1Cond)->Primero(Negacion)->Primero(CondRel)->Primero(ExpArit)->Primero(Op1)->Primero(RaizPot)->Primero(Numero)->{-}

TAS[Condicion,-]=Parte1Cond Disyuncion

Primero(Condicion)->Primero(Parte1Cond)->Primero(Negacion)->Primero(CondRel)->Primero(ExpArit)->Primero(Op1)->Primero(RaizPot)->{raiz}

TAS[Condicion,raiz]=Parte1Cond Disyuncion

Primero(Condicion)->Primero(Parte1Cond)->Primero(Negacion)->Primero(CondRel)->{[}

TAS[Condicion,[ ]=Parte1Cond Disyuncion

---------------------------------------------------------------------------------------------------------------------

Parte1Cond -> Negacion Conjuncion

Primero(Parte1Cond)->Primero(Negacion)->{not}

TAS[Parte1Cond,not]=Negacion Conjuncion

Primero(Parte1Cond)->Primero(Negacion)->Primero(CondRel)->Primero(ExpArit)->Primero(Op1)->Primero(RaizPot)->Primero(Numero)->{id}

TAS[Parte1Cond,id]=Negacion Conjuncion

Primero(Parte1Cond)->Primero(Negacion)->Primero(CondRel)->Primero(ExpArit)->Primero(Op1)->Primero(RaizPot)->Primero(Numero)->{(}

TAS[Parte1Cond,(]=Negacion Conjuncion

Primero(Parte1Cond)->Primero(Negacion)->Primero(CondRel)->Primero(ExpArit)->Primero(Op1)->Primero(RaizPot)->Primero(Numero)->{const}

TAS[Parte1Cond,const]=Negacion Conjuncion

Primero(Parte1Cond)->Primero(Negacion)->Primero(CondRel)->Primero(ExpArit)->Primero(Op1)->Primero(RaizPot)->Primero(Numero)->{-}

TAS[Parte1Cond,-]=Negacion Conjuncion

Primero(Parte1Cond)->Primero(Negacion)->Primero(CondRel)->Primero(ExpArit)->Primero(Op1)->Primero(RaizPot)->{raiz}

TAS[Parte1Cond,raiz]=Negacion Conjuncion

Primero(Parte1Cond)->Primero(Negacion)->Primero(CondRel)->Primero(CondRel)->{[}

TAS[Parte1Cond,[ ]=Negacion Conjuncion

---------------------------------------------------------------------------------------------------------------------

Negacion -> not CondRel | CondRel

Primero(Negacion)->{not}

TAS[Negacion,not]=not CondRel

Primero(Negacion)->Primero(CondRel)->Primero(ExpArit)->Primero(Op1)->Primero(RaizPot)->Primero(Numero)->{id}

TAS[Negacion,id]=CondRel

Primero(Negacion)->Primero(CondRel)->Primero(ExpArit)->Primero(Op1)->Primero(RaizPot)->Primero(Numero)->{(}

TAS[Negacion,(]=CondRel

Primero(Negacion)->Primero(CondRel)->Primero(ExpArit)->Primero(Op1)->Primero(RaizPot)->Primero(Numero)->{const}

TAS[Negacion,const]=CondRel

Primero(Negacion)->Primero(CondRel)->Primero(ExpArit)->Primero(Op1)->Primero(RaizPot)->Primero(Numero)->{-}

TAS[Negacion,-]=CondRel

Primero(Negacion)->Primero(CondRel)->Primero(ExpArit)->Primero(Op1)->Primero(RaizPot)->{raiz}

TAS[Negacion,raiz]=CondRel

Primero(Negacion)->Primero(CondRel)->Primero(CondRel)->{[}

TAS[Negacion,[ ]=CondRel

---------------------------------------------------------------------------------------------------------------------

Disyuncion -> or Parte1Cond Disyuncion | epsilon

Primero(Disyuncion)->{or}

TAS[Disyuncion,or]=or Parte1Cond Disyuncion

Siguiente(Disyuncion)->Siguiente(Condicion)->{]}

TAS[Disyuncion,] ] =epsilon

Siguiente(Disyuncion)->Siguiente(Condicion)->{entonces}

TAS[Disyuncion,entonces] =epsilon

Siguiente(Disyuncion)->Siguiente(Condicion)->{hacer}

TAS[Disyuncion, hacer] =epsilon

---------------------------------------------------------------------------------------------------------------------

Conjuncion -> and Parte1Cond Conjuncion | epsilon

Primero(Conjuncion)->{and}

TAS[Conjuncion,and]=and Parte1Cond Conjuncion

Siguiente(Conjuncion)->Siguiente(Parte1Cond)->Primero(Disyuncion)->{or}

TAS[Conjuncion,or]=epsilon

Siguiente(Conjuncion)->Siguiente(Parte1Cond)->Siguiente(Disyuncion)->Siguiente(Condicion)->{]}

TAS[Conjuncion,] ]=epsilon

Siguiente(Conjuncion)->Siguiente(Parte1Cond)->Siguiente(Disyuncion)->Siguiente(Condicion)->{entonces}

TAS[Conjuncion,entonces]=epsilon

Siguiente(Conjuncion)->Siguiente(Parte1Cond)->Siguiente(Disyuncion)->Siguiente(Condicion)->{hacer}

TAS[Conjuncion,hacer]=epsilon

---------------------------------------------------------------------------------------------------------------------

CondRel -> ExpArit OperadorRelacional ExpArit | [Condicion]

Primero(CondRel)->Primero(ExpArit)->Primero(Op1)->Primero(RaizPot)->Primero(Numero)->{id}

TAS[CondRel,id]=ExpArit OperadorRelacional ExpArit

Primero(CondRel)->Primero(ExpArit)->Primero(Op1)->Primero(RaizPot)->Primero(Numero)->{(}

TAS[CondRel,(]=ExpArit OperadorRelacional ExpArit

Primero(CondRel)->Primero(ExpArit)->Primero(Op1)->Primero(RaizPot)->Primero(Numero)->{const}

TAS[CondRel,const]=ExpArit OperadorRelacional ExpArit

Primero(CondRel)->Primero(ExpArit)->Primero(Op1)->Primero(RaizPot)->Primero(Numero)->{-}

TAS[CondRel,-]=ExpArit OperadorRelacional ExpArit

Primero(CondRel)->Primero(ExpArit)->Primero(Op1)->Primero(RaizPot)->{raiz}

TAS[CondRel,raiz]=ExpArit OperadorRelacional ExpArit

Primero(CondRel)->{[}

TAS[CondRel,[ ]=[Condicion]

---------------------------------------------------------------------------------------------------------------------

VarCondicional -> si Condicion entonces Cuerpo SINO

Primero(VarCondicional)->{si}

TAS[VarCondicional,si]=si Condicion entonces Cuerpo SINO

---------------------------------------------------------------------------------------------------------------------

SINO -> sino Cuerpo | epsilon

Primero(SINO)->{sino}

TAS[SINO,sino]=sino Cuerpo

Siguiente(SINO)->Siguiente(VarCondicional)->Siguiente(Sentencia)->{;}

TAS[SINO,;]=epsilon

---------------------------------------------------------------------------------------------------------------------

VarMientras -> mientras Condicion hacer Cuerpo

Primero(VarMientras)->{mientras}

TAS[VarMientras,mientras]=mientras Condicion hacer Cuerpo

---------------------------------------------------------------------------------------------------------------------

ExpArit -> Op1 SumaRes

Primero(ExpArit)->Primero(Op1)->Primero(RaizPot)->Primero(Numero)->{id}

TAS[ExpArit,id]=Op1 SumaRes

Primero(ExpArit)->Primero(Op1)->Primero(RaizPot)->Primero(Numero)->{(}

TAS[ExpArit,(]=Op1 SumaRes

Primero(ExpArit)->Primero(Op1)->Primero(RaizPot)->Primero(Numero)->{const}

TAS[ExpArit,const]=Op1 SumaRes

Primero(ExpArit)->Primero(Op1)->Primero(RaizPot)->Primero(Numero)->{-}

TAS[ExpArit,-]=Op1 SumaRes

Primero(ExpArit)->Primero(Op1)->Primero(RaizPot)->{raiz}

TAS[ExpArit,raiz]=Op1 SumaRes

---------------------------------------------------------------------------------------------------------------------

Op1 -> RaizPot MultDiv

Primero(Op1)->Primero(RaizPot)->Primero(Numero)->{id}

TAS[Op1,id]=RaizPot MultDiv

Primero(Op1)->Primero(RaizPot)->Primero(Numero)->{(}

TAS[Op1,(]=RaizPot MultDiv

Primero(Op1)->Primero(RaizPot)->Primero(Numero)->{const}

TAS[Op1,const]=RaizPot MultDiv

Primero(Op1)->Primero(RaizPot)->Primero(Numero)->{-}

TAS[Op1,-]=RaizPot MultDiv

Primero(Op1)->Primero(RaizPot)->{raiz}

TAS[Op1,raiz]=RaizPot MultDiv

---------------------------------------------------------------------------------------------------------------------

SumaRes -> + Op1 SumaRes| - Op1 SumaRes | epsilon

Primero(SumaRes)->{+}

TAS[SumaRes,+]= + Op1 SumaRes

Primero(SumaRes)->{-}

TAS[SumaRes,-]= - Op1 SumaRes

Siguiente(SumaRes)->Siguiente(ExpArit)->Siguiente(Asignacion)->Siguiente(Sentencia)->{;}

TAS[SumaRes,;]=epsilon

Siguiente(SumaRes)->Siguiente(ExpArit)->{)}

TAS[SumaRes,)]=epsilon

Siguiente(SumaRes)->Siguiente(ExpArit)->{OperadorRelacional}

TAS[SumaRes,OperadorRelacional]=epsilon

Siguiente(SumaRes)->Siguiente(ExpArit)->Siguiente(CondRel)->Siguiente(Negacion)->Primero(Conjuncion)->{and}

TAS[SumaRes,and]=epsilon

Siguiente(SumaRes)->Siguiente(ExpArit)->Siguiente(CondRel)->Siguiente(Negacion)->Siguiente(Conjuncion)->Siguiente(Parte1Cond)->Primero(Disyuncion)->{or}

TAS[SumaRes,or]=epsilon

Siguiente(SumaRes)->Siguiente(ExpArit)->Siguiente(CondRel)->Siguiente(Negacion)->Siguiente(Conjuncion)->Siguiente(Parte1Cond)->Primero(Disyuncion)->Siguiente(Condicion)->{]}

TAS[SumaRes,] ]=epsilon

Siguiente(SumaRes)->Siguiente(ExpArit)->Siguiente(CondRel)->Siguiente(Negacion)->Siguiente(Conjuncion)->Siguiente(Parte1Cond)->Primero(Disyuncion)->Siguiente(Condicion)->{entonces}

TAS[SumaRes,entonces]=epsilon

Siguiente(SumaRes)->Siguiente(ExpArit)->Siguiente(CondRel)->Siguiente(Negacion)->Siguiente(Conjuncion)->Siguiente(Parte1Cond)->Primero(Disyuncion)->Siguiente(Condicion)->{hacer}

TAS[SumaRes,hacer]=epsilon

---------------------------------------------------------------------------------------------------------------------

RaizPot -> Numero Pot | raiz(RaizPot)

Primero(RaizPot)->Primero(Numero)->{id}

TAS[RaizPot,id]=Numero Pot

Primero(RaizPot)->Primero(Numero)->{(}

TAS[RaizPot,(]=Numero Pot

Primero(RaizPot)->Primero(Numero)->{const}

TAS[RaizPot,const]=Numero Pot

Primero(RaizPot)->Primero(Numero)->{-}

TAS[RaizPot,-]=Numero Pot

Primero(RaizPot)->{raiz}

TAS[RaizPot,raiz]=raiz(RaizPot)

---------------------------------------------------------------------------------------------------------------------

Pot -> ^ RaizPot | epsilon

Primero(Pot)->{^}

TAS[Pot,^]=^ RaizPot

Siguiente(Pot)->Siguiente(RaizPot)->Primero(MultDiv)->{\*}

TAS[Pot,\*]=epsilon

Siguiente(Pot)->Siguiente(RaizPot)->Primero(MultDiv)->{/}

TAS[Pot,/]=epsilon

Siguiente(Pot)->Siguiente(RaizPot)->Siguiente(MultDiv)->Siguiente(Op1)->Primero(SumaRes)->{+}

TAS[Pot,+]=epsilon

Siguiente(Pot)->Siguiente(RaizPot)->Siguiente(MultDiv)->Siguiente(Op1)->Primero(SumaRes)->{-}

TAS[Pot,-]=epsilon

Siguiente(Pot)->Siguiente(RaizPot)->Siguiente(MultDiv)->Siguiente(Op1)->Siguiente(SumaRes)->Siguiente(ExpArit)->Siguiente(Asignacion)->Siguiente(Sentencia)->{;}

TAS[Pot,;]=epsilon

Siguiente(Pot)->Siguiente(RaizPot)->Siguiente(MultDiv)->Siguiente(Op1)->Siguiente(SumaRes)->Siguiente(ExpArit)->{)}

TAS[Pot,)]=epsilon

Siguiente(Pot)->Siguiente(RaizPot)->Siguiente(MultDiv)->Siguiente(Op1)->Siguiente(SumaRes)->Siguiente(ExpArit)->{OperadorRelacional}

TAS[Pot,OperadorRelacional]=epsilon

Siguiente(Pot)->Siguiente(RaizPot)->Siguiente(MultDiv)->Siguiente(Op1)->Siguiente(SumaRes)->Siguiente(ExpArit)->Siguiente(CondRel)->Siguiente(Negacion)->Primero(Conjuncion)->{and}

TAS[Pot,and]=epsilon

Siguiente(Pot)->Siguiente(RaizPot)->Siguiente(MultDiv)->Siguiente(Op1)->Siguiente(SumaRes)->Siguiente(ExpArit)->Siguiente(CondRel)->Siguiente(Negacion)->Siguiente(Conjuncion)->Siguiente(Parte1Cond)->Primero(Disyuncion)->{or}

TAS[Pot,or]=epsilon

Siguiente(Pot)->Siguiente(RaizPot)->Siguiente(MultDiv)->Siguiente(Op1)->Siguiente(SumaRes)->Siguiente(ExpArit)->Siguiente(CondRel)->Siguiente(Negacion)->Siguiente(Conjuncion)->Siguiente(Parte1Cond)->Siguiente(Disyuncion)->Siguiente(Condicion)->{]}

TAS[Pot,] ]=epsilon

Siguiente(Pot)->Siguiente(RaizPot)->Siguiente(MultDiv)->Siguiente(Op1)->Siguiente(SumaRes)->Siguiente(ExpArit)->Siguiente(CondRel)->Siguiente(Negacion)->Siguiente(Conjuncion)->Siguiente(Parte1Cond)->Siguiente(Disyuncion)->Siguiente(Condicion)->{entonces}

TAS[Pot,entonces]=epsilon

Siguiente(Pot)->Siguiente(RaizPot)->Siguiente(MultDiv)->Siguiente(Op1)->Siguiente(SumaRes)->Siguiente(ExpArit)->Siguiente(CondRel)->Siguiente(Negacion)->Siguiente(Conjuncion)->Siguiente(Parte1Cond)->Siguiente(Disyuncion)->Siguiente(Condicion)->{hacer}

TAS[Pot,hacer]=epsilon

---------------------------------------------------------------------------------------------------------------------

MultDiv -> \* RaizPot MultDiv | / RaizPot MultDiv | epsilon

Primero(MultDiv)->{\*}

TAS[MultDiv,\*]= \* RaizPot MultDiv

Primero(MultDiv)->{/}

TAS[MultDiv,/]= / RaizPot MultDiv

Siguiente(MultDiv)->Siguiente(Op1)->Primero(SumaRes)->{+}

TAS[MultDiv,+]=epsilon

Siguiente(MultDiv)->Siguiente(Op1)->Primero(SumaRes)->{-}

TAS[MultDiv,-]=epsilon

Siguiente(MultDiv)->Siguiente(Op1)->Siguiente(SumaRes)->Siguiente(ExpArit)->Siguiente(Asignacion)->Siguiente(Sentencia)->{;}

TAS[MultDiv,;]=epsilon

Siguiente(MultDiv)->Siguiente(Op1)->Siguiente(SumaRes)->Siguiente(ExpArit)->{)}

TAS[MultDiv,)]=epsilon

Siguiente(MultDiv)->Siguiente(Op1)->Siguiente(SumaRes)->Siguiente(ExpArit)->{OperadorRelacional}

TAS[MultDiv,OperadorRelacional]=epsilon

Siguiente(MultDiv)->Siguiente(Op1)->Siguiente(SumaRes)->Siguiente(ExpArit)->Siguiente(CondRel)->Siguiente(Negacion)->Primero(Conjuncion)->{and}

TAS[MultDiv,and]=epsilon

Siguiente(MultDiv)->Siguiente(Op1)->Siguiente(SumaRes)->Siguiente(ExpArit)->Siguiente(CondRel)->Siguiente(Negacion)->Siguiente(Conjuncion)->Siguiente(Parte1Cond)->Primero(Disyuncion)->{or}

TAS[MultDiv,or]=epsilon

Siguiente(MultDiv)->Siguiente(Op1)->Siguiente(SumaRes)->Siguiente(ExpArit)->Siguiente(CondRel)->Siguiente(Negacion)->Siguiente(Conjuncion)->Siguiente(Parte1Cond)->Siguiente(Disyuncion)->Siguiente(Condicion)->{]}

TAS[MultDiv,] ]=epsilon

Siguiente(MultDiv)->Siguiente(Op1)->Siguiente(SumaRes)->Siguiente(ExpArit)->Siguiente(CondRel)->Siguiente(Negacion)->Siguiente(Conjuncion)->Siguiente(Parte1Cond)->Siguiente(Disyuncion)->Siguiente(Condicion)->{entonces}

TAS[MultDiv,entonces]=epsilon

Siguiente(MultDiv)->Siguiente(Op1)->Siguiente(SumaRes)->Siguiente(ExpArit)->Siguiente(CondRel)->Siguiente(Negacion)->Siguiente(Conjuncion)->Siguiente(Parte1Cond)->Siguiente(Disyuncion)->Siguiente(Condicion)->{hacer}

TAS[MultDiv,hacer]=epsilon