



GRAMATICA DE LOS LENGUAJES

FIUSAC Copy Analyzer



3 DE SEPTIEMBRE DE 2021

ORGANIZACIÓN DE LENGUAJES Y COMPILADORES 1 - N
BRAYAN HAMLLELO ESTEVEM PRADO MARROQUIN - 201801369

CONTENIDO

| | |
|---|---|
| GRAMATICA DEL LENGUAJE FCA | 1 |
| GRAMATICA DEL LENGUAJE JAVASCRIPT | 5 |

GRAMATICA DEL LENGUAJE FCA

Terminales = { Compare, DefinirGlobales, GenerarReporteEstadistico, string, double, GraficaBarras, Titulo, Ejex, Valores, TituloX, TituloY, GraficaPie, GraficaLineas, Archivo, :, :, ,, {, }, (,), [,], =, IDENTIFICADOR, CADENA, DD }

Donde:

ENTERO: [0-9]+

LETRA: [a-zA-Z]

IDENTIFICADOR, es una expresión regular la cual es:
 $\{LETRA\}(\{LETRA\}|_|\{ENTERO\})^*$

DD: [0-9]+(".[0-9]+)?

BPCADENA = "[^"]*"

BPRUTA = "[^']*"

No terminales = {ini, contenido general, títulos, cuerpo, graficas, datosgraficabarra, datosgraficapie, datosgraficalineal, Intitulox, Intituloy, Arch, contenidox, contenidoy, ejes, invalores, compara}

No terminal inicial: ini

Producciones:

Ini::= GenerarReporteEstadistico { contenido general }

Contenido general::= DefinirGlobales { cuerpo } graficas

| compara DefinirGlobales { cuerpo } graficas

| DefinirGlobales { cuerpo } graficas compara

Cuerpo::= string IDENTIFICADOR = BPCADENA ;

| double IDENTIFICADOR = DD ;

|Cuerpo string IDENTIFICADOR = CADENA ;

| double IDENTIFICADOR = DD ;

Graficas::= GraficaBarras { datosgraficabarra }

| GraficaBarras { datosgraficabarra } compara

| GraficaPie { datosgraficapie }

| GraficaPie { datosgraficapie } compara

| GraficaLineas { datosgraficalineal }

| GraficaLineas { datosgraficalineal } compara

| Graficas GraficaBarras { datosgraficabarra }

| Graficas GraficaBarras { datosgraficabarra } compara

| Graficas GraficaPie { datosgraficapie }

| Graficas GraficaPie { datosgraficapie } compara

| Graficas GraficaLineas { datosgraficalineal }

| Graficas GraficaLineas { datosgraficalineal } compara

datosgraficabarra::=datosgraficabarra titulos

|datosgraficabarra ejesx

|datosgraficabarra valores

|datosgraficabarra titulox

|datosgraficabarra tituloy

|titulos

|ejesx

|valores

|titulox

|tituloy

datosgraficapie::=datosgraficapie titulos

|datosgraficapie ejesx

|datosgraficapie valores

|titulos

|ejesx

|valores

datosgraficalineal::=datosgraficalineal titulos

|datosgraficalineal Arch

|titulos

|Arch

titulos::= Titulo : IDENTIFICADOR ;

|Titulo : CADENA ;

ejesx::= Ejex: [contenidox]

contenidox::=

contenidox CADENA

|contenidox IDENTIFICADOR

|contenidox ,

|CADENA

|IDENTIFICADOR

invalores::= Valores : [contendov] ;

contendov::=

contendov DD

|contendov IDENTIFICADOR

|contendov ,

|DD

|IDENTIFICADOR

Intitulox::= TituloX: IDENTIFICADOR ;

| TituloX: CADENA ;

Intituly::= TituloY: IDENTIFICADOR ;

| TituloY : CADENA ;

Arch::= Archivo : IDENTIFICADOR ;

| Archivo : CADENA ;

compara::= Compare (CADENA , BPCADENA) ;

| Compare (BPRUTA , BPRUTA);

GRAMATICA DEL LENGUAJE JAVASCRIPT

Terminales = { class , var , let , const , if , else , for , while , do , switch , case , console.log , requiere , break , ; , : , . , , , (,) , { , } , default , + , - , * , ** , / , > , < , % , && , || , = , == , != , >= , <= , ! , IDENTIFICADOR , BPCADENA , DD , false , true , ++ , -- , LINECOMMENT , MULTILINECOMMENT , BPRUTA }

Donde=

LINECOMMENT = (//.*\n)((//.*\r\n)((//.*\r)

MULTILINECOMMENT = "<!"!""*([^\>]|["'"]>)"!["^>])*!"!""!>"

ENTERO=[0-9]+

DD=[0-9]+("."[0-9]+)?

LETRA = [a-zA-Z]

IDENTIFICADOR = {LETRA}({LETRA}|" _" {ENTERO})*

BPCADENA = \"[^\"]*\"

BPRUTA = '[^']*'

No Terminal = { ini , procesos , parámetros , variable , impresión , opmate , concatenar , condición , forvariable , condicionfor , foraumento , casos , signos , comentario , siguiente }

No terminal Inicial = ini

Producciones:

ini::=ini comentario class IDENTIFICADOR () { procesos }

|ini class IDENTIFICADOR { comentario procesos }

|ini class IDENTIFICADOR { procesos comentario }

|ini class IDENTIFICADOR { procesos }

|ini IDENTIFICADOR () { procesos }

|ini IDENTIFICADOR (parametros) { procesos }

|ini var IDENTIFICADOR = variable

|ini let IDENTIFICADOR = variable

|ini const IDENTIFICADOR = variable

|ini var IDENTIFICADOR

|ini let IDENTIFICADOR

|ini const IDENTIFICADOR

|ini var IDENTIFICADOR

|ini let IDENTIFICADOR

|ini const IDENTIFICADOR

|class IDENTIFICADOR () { procesos }

|IDENTIFICADOR () { procesos }

|IDENTIFICADOR () ;

|IDENTIFICADOR (parametros) ;

|IDENTIFICADOR ()

|IDENTIFICADOR (parametros)

|IDENTIFICADOR (parametros) { procesos }

|var IDENTIFICADOR = variable

|let IDENTIFICADOR = variable

|const IDENTIFICADOR = variable

|var IDENTIFICADOR

|let IDENTIFICADOR

|const IDENTIFICADOR

|var IDENTIFICADOR ;

|let IDENTIFICADOR ;

|const IDENTIFICADOR ;

comentario::=comentario LINECOMMENT

|comentario MULTILINECOMMENT

| LINECOMMENT

|MULTILINECOMMENT

parametros::=IDENTIFICADOR

|parametros , IDENTIFICADOR

|parametros signos DD

|parametros signos (variable)

variable::=BPCADENA ;

|BPCADENA

|BPRUTA ;

|BPRUTA

|DD ;

|DD

|DD signos DD ;

|DD signos DD

|true ;

|true

|false ;

|false

| (opmate) opmate ;

| (opmate) opmate

| (opmate) - DD ;

| (opmate) - DD

| (opmate) + DD ;

| (opmate) + DD

| (opmate) * DD ;

| (opmate) * DD

| (opmate) / DD ;

| (opmate) / DD

| (opmate) ** DD ;

| (opmate) ** DD

| (opmate) % DD ;

| (opmate) % DD

| (opmate) - opmate ;

| (opmate) - opmate

| (opmate) + opmate ;

| (opmate) + opmate

| (opmate) * opmate ;

| (opmate) * opmate

| (opmate) / opmate ;

| (opmate) / opmate

| (opmate) ** opmate ;

| (opmate) ** opmate

| (opmate) % opmate ;

| (opmate) % opmate

| (opmate) ;

| (opmate)

| requiere (BPCADENA) ;

| requiere (BPCADENA)

| signos DD signos IDENTIFICADOR signos DD

|signos DD signos IDENTIFICADOR signos DD ;
|signos DD signos IDENTIFICADOR signos DD signos variable
|DD signos IDENTIFICADOR signos DD
|DD signos IDENTIFICADOR signos DD ;
|DD signos IDENTIFICADOR signos DD signos variable
|signos DD signos IDENTIFICADOR signos IDENTIFICADOR
|signos DD signos IDENTIFICADOR signos IDENTIFICADOR ;
|signos DD signos IDENTIFICADOR signos IDENTIFICADOR signos
variable
|DD signos IDENTIFICADOR signos IDENTIFICADOR
|DD signos IDENTIFICADOR signos IDENTIFICADOR ;
|DD signos IDENTIFICADOR signos IDENTIFICADOR signos variable
|signos IDENTIFICADOR signos DD signos DD
|signos IDENTIFICADOR signos DD signos DD ;
|signos IDENTIFICADOR signos DD signos DD signos variable
|IDENTIFICADOR signos DD signos DD
|IDENTIFICADOR signos DD signos DD ;
|IDENTIFICADOR signos DD signos DD signos variable
|signos IDENTIFICADOR signos IDENTIFICADOR signos DD
|signos IDENTIFICADOR signos IDENTIFICADOR signos DD ;
|signos IDENTIFICADOR signos IDENTIFICADOR signos DD signos
variable
|IDENTIFICADOR signos IDENTIFICADOR signos DD
|IDENTIFICADOR signos IDENTIFICADOR signos DD ;
|IDENTIFICADOR signos IDENTIFICADOR signos DD signos variable
|signos IDENTIFICADOR signos DD signos IDENTIFICADOR

|signos IDENTIFICADOR signos DD signos IDENTIFICADOR ;

|signos IDENTIFICADOR signos DD signos IDENTIFICADOR signos
variable

|IDENTIFICADOR signos DD signos IDENTIFICADOR

|IDENTIFICADOR signos DD signos IDENTIFICADOR ;

|IDENTIFICADOR signos DD signos IDENTIFICADOR signos variable

|signos DD signos DD signos IDENTIFICADOR

|signos DD signos DD signos IDENTIFICADOR ;

|signos DD signos DD signos IDENTIFICADOR signos variable

|DD signos DD signos IDENTIFICADOR

|DD signos DD signos IDENTIFICADOR ;

|DD signos DD signos IDENTIFICADOR signos variable

|signos IDENTIFICADOR signos IDENTIFICADOR signos IDENTIFICADOR

|signos IDENTIFICADOR signos IDENTIFICADOR signos IDENTIFICADOR
;

|signos IDENTIFICADOR signos IDENTIFICADOR signos IDENTIFICADOR
signos variable

|IDENTIFICADOR signos IDENTIFICADOR signos IDENTIFICADOR

|IDENTIFICADOR signos IDENTIFICADOR signos IDENTIFICADOR ;

|IDENTIFICADOR signos IDENTIFICADOR signos IDENTIFICADOR signos
variable

signos::=+

|-

|*

|/

|**

|%

```
procesos::=procesos IDENTIFICADOR ( ) { procesos }  
|procesos IDENTIFICADOR ( parametros ) { procesos }  
|procesos var IDENTIFICADOR = variable  
|procesos let IDENTIFICADOR = variable  
|procesos const IDENTIFICADOR = variable  
|procesos var IDENTIFICADOR  
|procesos let IDENTIFICADOR  
|procesos const IDENTIFICADOR  
|procesos var IDENTIFICADOR ;  
|procesos let IDENTIFICADOR ;  
|procesos const IDENTIFICADOR ;  
|procesos if ( condicion ) { procesos } siguiente  
|procesos if ( condicion ) { procesos }  
|procesos for ( forvariable condicionfor foraumento ) { procesos }  
|procesos while ( condicion ) { procesos }  
|procesos do:var1 { procesos } while ( condicion )  
|procesos do:var1 { procesos } while ( condicion ) ;  
|procesos switch ( IDENTIFICADOR ) { casos }  
|procesos console.log ( impresion ) ;  
|procesos console.log ( impresion )  
|procesos IDENTIFICADOR = IDENTIFICADOR  
|procesos IDENTIFICADOR = concatenar  
|procesos IDENTIFICADOR . IDENTIFICADOR ( parametros )  
|procesos IDENTIFICADOR . IDENTIFICADOR ( )
```

|procesos IDENTIFICADOR . IDENTIFICADOR (parametros) ;

|procesos IDENTIFICADOR . IDENTIFICADOR () ;

|procesos IDENTIFICADOR (parametros) ;

|procesos IDENTIFICADOR (parametros)

|IDENTIFICADOR () { procesos }

|IDENTIFICADOR (parametros) { procesos }

|var IDENTIFICADOR = variable

|let IDENTIFICADOR = variable

|const IDENTIFICADOR = variable

|var IDENTIFICADOR

|let IDENTIFICADOR

|const IDENTIFICADOR

|var IDENTIFICADOR ;

|let IDENTIFICADOR ;

|const IDENTIFICADOR ;

| if (condicion) { procesos } siguiente

| if (condicion) { procesos }

|for (forvariable condicionfor foraumento) { procesos }

|while (condicion) { procesos }

|do:var1 { procesos } while (condicion)

|do:var1 { procesos } while (condicion) ;

|switch (IDENTIFICADOR) { casos }

|console.log (impresion) ;

|console.log (impresion)

|IDENTIFICADOR = IDENTIFICADOR

|IDENTIFICADOR = concatenar

|IDENTIFICADOR . IDENTIFICADOR (parametros)
|IDENTIFICADOR . IDENTIFICADOR ()
|IDENTIFICADOR . IDENTIFICADOR (parametros) ;
|IDENTIFICADOR . IDENTIFICADOR () ;
|IDENTIFICADOR (parametros) ;
|IDENTIFICADOR (parametros)

siguiente::=siguiente else:var1 { procesos }
|siguiente else:var1 if (condicion) { procesos }
|else:var1 { procesos }
|else:var1 if (condicion) { procesos }

concatenar::=concatenar + IDENTIFICADOR
|concatenar + DD
|concatenar + BPCADENA
|concatenar + BPRUTA
|IDENTIFICADOR + DD
|IDENTIFICADOR + BPCADENA
|IDENTIFICADOR + BPRUTA
|DD + IDENTIFICADOR
|BPCADENA + IDENTIFICADOR
|BPRUTA + IDENTIFICADOR
|IDENTIFICADOR + IDENTIFICADOR

forvariable::=var IDENTIFICADOR = variable
|let IDENTIFICADOR = variable

|IDENTIFICADOR = IDENTIFICADOR ;

|IDENTIFICADOR opmate ;

|IDENTIFICADOR ;

|IDENTIFICADOR = variable

condicionfor::=IDENTIFICADOR > IDENTIFICADOR ;

|IDENTIFICADOR < IDENTIFICADOR ;

|IDENTIFICADOR >= IDENTIFICADOR ;

|IDENTIFICADOR <= IDENTIFICADOR ;

|IDENTIFICADOR > DD ;

|IDENTIFICADOR < DD ;

|IDENTIFICADOR >= DD ;

|IDENTIFICADOR <= DD ;

|IDENTIFICADOR > opmate ;

|IDENTIFICADOR < opmate ;

|IDENTIFICADOR >= opmate ;

|IDENTIFICADOR <= opmate ;

foraumentor::=IDENTIFICADOR ++

|IDENTIFICADOR --

|IDENTIFICADOR + = DD

condicion::=condicion BPCADENA == BPCADENA

|condicion DD == DD

|condicion DD > DD

|condicion DD < DD

|condicion DD >= DD
|condicion DD <= DD
|condicion opmate == DD
|condicion opmate > DD
|condicion opmate < DD
|condicion opmate >= DD
|condicion opmate <= DD
|condicion opmate == IDENTIFICADOR
|condicion opmate > IDENTIFICADOR
|condicion opmate < IDENTIFICADOR
|condicion opmate >= IDENTIFICADOR
|condicion opmate <= IDENTIFICADOR
|condicion IDENTIFICADOR == DD
|condicion IDENTIFICADOR > DD signos DD
|condicion IDENTIFICADOR > IDENTIFICADOR signos DD
|condicion IDENTIFICADOR > DD signos IDENTIFICADOR
|condicion IDENTIFICADOR > (opmate)
|condicion IDENTIFICADOR == true
|condicion IDENTIFICADOR == false
|condicion IDENTIFICADOR == BPCADENA
|condicion IDENTIFICADOR > DD
|condicion IDENTIFICADOR < DD
|condicion IDENTIFICADOR < DD signos DD
|condicion IDENTIFICADOR < IDENTIFICADOR signos DD
|condicion IDENTIFICADOR < DD signos IDENTIFICADOR
|condicion IDENTIFICADOR < (opmate)

|condicion IDENTIFICADOR >= DD
|condicion IDENTIFICADOR >= DD signos DD
|condicion IDENTIFICADOR >= IDENTIFICADOR signos DD
|condicion IDENTIFICADOR >= DD signos IDENTIFICADOR
|condicion IDENTIFICADOR >= (opmate)
|condicion IDENTIFICADOR <= DD
|condicion IDENTIFICADOR <= DD signos DD
|condicion IDENTIFICADOR <= IDENTIFICADOR signos DD
|condicion IDENTIFICADOR <= DD signos IDENTIFICADOR
|condicion IDENTIFICADOR <= (opmate)
|condicion IDENTIFICADOR > IDENTIFICADOR
|condicion IDENTIFICADOR < IDENTIFICADOR
|condicion IDENTIFICADOR >= IDENTIFICADOR
|condicion IDENTIFICADOR <= IDENTIFICADOR
|condicion IDENTIFICADOR == IDENTIFICADOR
|condicion DD ! DD
|condicion BPCADENA ! BPCADENA
|condicion IDENTIFICADOR ! DD
|condicion IDENTIFICADOR ! true
|condicion IDENTIFICADOR ! false
|condicion IDENTIFICADOR ! BPCADENA
|condicion IDENTIFICADOR ! IDENTIFICADOR
|condicion != IDENTIFICADOR
|condicion (condicion)
|condicion IDENTIFICADOR
|condicion | |

|condicion &&

|BPCADENA == BPCADENA

|DD == DD

|DD > DD

|DD < DD

|DD >= DD

|DD <= DD

|opmate == DD

|opmate > DD

|opmate < DD

|opmate >= DD

|opmate <= DD

|opmate ! DD

|opmate == IDENTIFICADOR

|opmate > IDENTIFICADOR

|opmate < IDENTIFICADOR

|opmate >= IDENTIFICADOR

|opmate <= IDENTIFICADOR

|opmate ! IDENTIFICADOR

|IDENTIFICADOR

|IDENTIFICADOR == DD

|IDENTIFICADOR == DD signos DD

|IDENTIFICADOR == IDENTIFICADOR signos DD

|IDENTIFICADOR == DD signos IDENTIFICADOR

|IDENTIFICADOR == (opmate)

|IDENTIFICADOR == true

|IDENTIFICADOR == false
|IDENTIFICADOR == BPCADENA
|IDENTIFICADOR > DD
|IDENTIFICADOR > DD signos DD
|IDENTIFICADOR > IDENTIFICADOR signos DD
|IDENTIFICADOR > DD signos IDENTIFICADOR
|IDENTIFICADOR > (opmate)
|IDENTIFICADOR < DD
|IDENTIFICADOR < DD signos DD
|IDENTIFICADOR < IDENTIFICADOR signos DD
|IDENTIFICADOR < DD signos IDENTIFICADOR
|IDENTIFICADOR < (opmate)
|IDENTIFICADOR >= DD
|IDENTIFICADOR >= DD signos DD
|IDENTIFICADOR >= IDENTIFICADOR signos DD
|IDENTIFICADOR >= DD signos IDENTIFICADOR
|IDENTIFICADOR >= (opmate)
|IDENTIFICADOR <= DD
|IDENTIFICADOR <= DD signos DD
|IDENTIFICADOR <= IDENTIFICADOR signos DD
|IDENTIFICADOR <= DD signos IDENTIFICADOR
|IDENTIFICADOR <= (opmate)
|IDENTIFICADOR > IDENTIFICADOR
|IDENTIFICADOR < IDENTIFICADOR
|IDENTIFICADOR >= IDENTIFICADOR
|IDENTIFICADOR <= IDENTIFICADOR

|IDENTIFICADOR == IDENTIFICADOR

|DD ! DD

|BPCADENA ! BPCADENA

|IDENTIFICADOR ! DD

|IDENTIFICADOR ! true

|IDENTIFICADOR ! false

|IDENTIFICADOR ! BPCADENA

|IDENTIFICADOR ! IDENTIFICADOR

|!= IDENTIFICADOR

| (condicion)

casos::=casos case:var1 DD : procesos break ;

|casos default:var1 : procesos break ;

|casos default:var1 : procesos

|casos case:var1 BPCADENA : procesos break ;

|case:var1 DD : procesos break ;

|default:var1 : procesos break ;

|default:var1 : procesos

|case:var1 BPCADENA : procesos break ;

impresion::=impresion + DD{:RESULT=var1;;}

|impresion + BPCADENA:var1 {:RESULT=var1;;}

|impresion + != IDENTIFICADOR {:RESULT=var1;;}

|impresion + IDENTIFICADOR {:RESULT=var1;;}

|impresion + (opmate)

|BPCADENA:var1 {:RESULT=var1;;}

|IDENTIFICADOR {:RESULT=var1;;}

|DD

|- DD

|- (opmate)

|- (DD)

|- (IDENTIFICADOR)

|(opmate)

|IDENTIFICADOR ++

|IDENTIFICADOR --

|!= IDENTIFICADOR

opmate::=opmate (opmate)

|opmate DD ** DD

|opmate DD / DD

|opmate DD % DD

|opmate DD * DD

|opmate DD + DD

|opmate DD - DD

|opmate DD ** IDENTIFICADOR

|opmate DD / IDENTIFICADOR

|opmate DD % IDENTIFICADOR

|opmate DD * IDENTIFICADOR

|opmate DD + IDENTIFICADOR

|opmate DD - IDENTIFICADOR

|(opmate)

|DD ** DD

|DD / DD

|DD % DD

|DD * DD

|DD + DD

|DD - DD

|DD ** IDENTIFICADOR

|DD / IDENTIFICADOR

|DD % IDENTIFICADOR

|DD * IDENTIFICADOR

|DD + IDENTIFICADOR

|DD – IDENTIFICADOR