# 

Teoría de Autómatas

## Tarea del segundo bimestre

Nombre:

# Cueva Castillo Tania Piedad

# Guachisaca Apolo Carlos Andres

# Márquez Tepán Juan Fernando

# Pazmiño Cardoso David Sebastian

# Vargas Rubio Dayana Alexandra

Tema:

# Gramáticas libres de contexto

Tutor:

# Martha Vanessa Agila Palacios

Fecha:

# 16 de enero del 2019

Tabla de contenido

1. Descripción del Problema 2
   1. Reglas 2
   2. Protocolo 3
   3. Diagrama Formal AFD 3
2. Definición de las gramáticas independientes del contexto 5
3. Cadenas de aceptación 5
4. Reglas 6
5. Derivaciones y Árboles 6
6. Transformación a FNC 9
   1. Eliminación de producciones vacías 10
      1. Identificación de variable nulas 10
      2. Simplificar sin variables nulas 10
   2. Producciones Inútiles 10
   3. Producciones Unitarias 11
   4. Transformar a FNC 11
7. Bibliografía: 14

## **Descripción del Problema**

La Subsecretaria de Gobierno Electrónico -SGE entre sus objetivos tiene el evaluar proyectos tecnológicos de las instituciones del Estado con el fin emitir el oficio de aprobación del proyecto postulado.

El actor para este proceso sería el proyecto

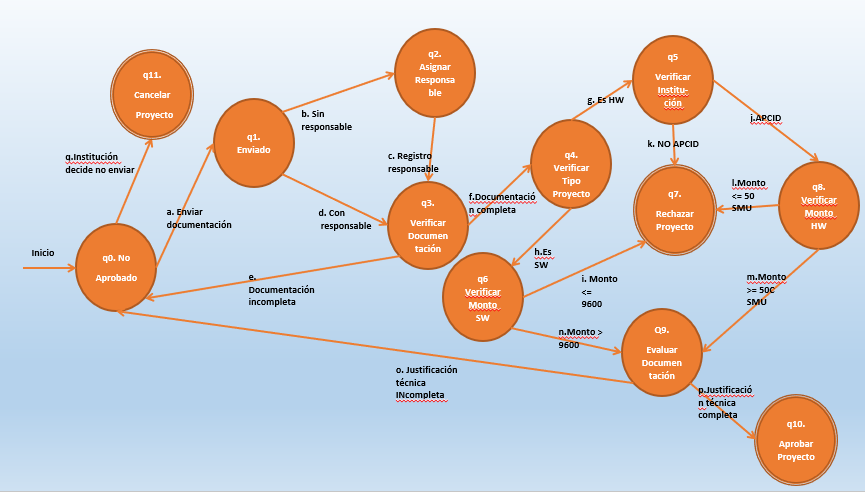
## **Reglas**

1. El proceso empieza cuando la institución envia la documentación del proyecto a SGE siendo su estado inicial *No Aprobado.*
2. La institución procede a enviar la documentación cambiado el estado de proyecto a *Enviado.*
3. Estando el proyecto en el estado de Envidado las SGE verifica si dicho proyecto ha sido asignado responsable en caso que tenga un responsable el proyecto pasa al estado de *Verificación de Documentación,* En el caso que el proyecto no le ha asignado responsable el proyecto pasa a un estado de *Asignación de Responsable* una vez que se asigna el responsable pasa la estado de *Verificación de Documentación.*
4. Estando el estado de *Verificación de Documentación* se procese a verificar la documentación pasando al estado de *Verificación Tipo de Proyecto* si el tipo de proyecto es de Hardware (HW) se procede *a Verificar la institución* que postula si la institución no pertenece al Poder Ejecutivo (APCID) se rechaza el proyecto y termina el proceso, en caso que sea un proyecto de Software se debe verificar el monto de aprobación de software quedando en el proyecto en estado de *Verificar monto de software .*
5. Estando en estado de *Verificar Monto de Software* se debe verificar los montos de aprobación si el monto es menor a 9600 se *Rechaza proyecto* y termina el proceso, si el monto es mayor a 9600 el proyecto pasa al estado de *Evaluación de documentación.*
6. Estando el proceso de *Verificar Institución* se valida la institución que postulo el proyecto si esta no pertenece al poder Ejecutivo se Rechaza el Proyecto en caso que la institución pertenezca al poder Ejecutivo se debe verificar el monto de aprobación para el Hardware en caso que el monto sea menor a 50 SMU se Rechaza el proyecto y termina el proceso , en caso que sea mayor 50 SMU este pasa al estado de *Evaluación de documentación.*
7. En el estado de Evaluación de documentación se verifica la Justificación técnica si estas están completas se *Aprueba el proyecto* y se emite el informe de aprobación y termina el proceso en caso que no esten completas se procede a devolver el proceso volviendo nuevamente al estado de No Aprobado.
8. Estando el estado de No Aprobado la institución puede decir no enviar y cancelar el proyecto quedando el proyecto en estado cancelado y terminando el proceso.

## **Protocolo**

Acciones que deben realizar las instituciones del Estado para obtener la aprobación de los proyectos tecnológicos.

## **Diagrama Formal AFD**



1. Un conjunto finito de *estados*, a menudo designado como *Q*. para este caso tenemos

|  |  |  |
| --- | --- | --- |
|  | **Estados** | **Descripción** |
| .q0 | NoApr | No Aprobado |
| .q1 | Env | Enviado |
| .q2 | AsigRes | Asignar Responsable |
| .q3 | VerDoc | Verificar Documentación |
| .q4 | VerTipPro | Verificar Tipo Proyecto |
| .q5 | VerIns | Verificar Institución |
| .q6 | VerMonSw | Verificar Monto Sw |
| .q7 | RecProy | Rechazar Proyecto |
| .q8 | VerMonHw | Verificar Monto HW |
| .q9 | EvaDoc | Evaluar Documentación |
| .q10 | AprProy | Aprobar Proyecto |
| .q11 | CanProy | Cancelar Proyecto |

Q = {NoApr,Env,AsigRes,VerDoc,VerTipPro,VerIns,

VerMonSw, RecProy, VerMonHw, EvaDoc, AprProy, CanProy}

2. Un conjunto finito de *símbolos de entrada*, a menudo designado como .∑

|  |  |  |
| --- | --- | --- |
|  | Alfabeto | Transiciones |
| a | EnvDoc | Enviar documentación |
| b | SinRes | Sin Responsable |
| c | RegRes | Registro Responsable |
| d | ConRes | Con Responsable |
| e | DocInc | Documentos Incompletos |
| f | DocCom | Documentos completos |
| g | EsHw | Es Hw |
| h | EsSw | Es Sw |
| k | NoAPCID | No APCID |
| j | APCID | APCID |
| i | MonMen9 | Monto < 9600 |
| n | MonMay9 | Monto > 9600 |
| l | MonMen5 | Monto < 50 SMU |
| m | MonMay5 | Monto > 50 SMU |
| p | JusTecCom | Justificación Técnica completa |
| o | JusTecInc | Justificación Técnica incompleta |
| q | InsNoEnv | Institución decide no enviar |

## **Definición de las gramáticas independientes del contexto**

1. Conjunto de símbolos que forma las cadenas del lenguaje que se denomina alfabeto terrminal

V={EnvDoc, SinRes, RegRes, ConRes, DocInc, DocCom, EsHw, EsSw, NoAPCID, APCID, MonMen9, MonMay9, MonMen5, MonMay5, JusTecCom, JusTecInc, InsNoEnv }

1. Conjunto finito de variables, denominado símbolos NO terminales

T={AproProy, VerificarResponsable, AsignarRespon, VerificarDocumen, VerifTipoProy}

1. Un símbolo inicial en este caso AproProy
2. Conjunto de producciones P o Reglas

## **Cadenas de aceptación**

|  |  |  |
| --- | --- | --- |
| **cadenas** | **estado** |  |
| abceq | Accept | Carlos |
| abcfgjl | Accept | Carlos |
| abcfgjmoq | Accept | Carlos |
| abcfgjmp | Accept | Carlos |
| abcfgk | Accept | David |
| abcfhi | Accept | David |
| abcfhnoq | Accept | David |
| abcfhnp | Accept | David |
| adeq | Accept | Dayana |
| adeq | Accept | Dayana |
| adfgjl | Accept | Dayana |
| adfgjmoq | Accept | Dayana |
| adfgjmp | Accept | Fernando |
| adfgk | Accept | Fernando |
| adfhi | Accept | Fernando |
| adfhnoq | Accept | Tania |
| adfhnp | Accept | Tania |
| q | Accept | Tania |

## **Reglas**

|  |  |
| --- | --- |
|  | **Propuesta de la gramática** |
| S-> aA | <AproProy> ::= envDoc<VerificarResponsable> |
| S->qK | <AproProy> ::= insNoEnv<K> |
| K-> l | <K> ::= l |
| J-> l | <J> ::= l |
| G-> l | <G> ::= l |
| A->bB | <VerificarResponsable> ::= sinRes<AsignarRespon> |
| A-> dC | <VerificarResponsable> ::= conRes<VerificarDocumen> |
| B->cC | <AsignarRespon>::= regRes<VerificarDocumen> |
| C->eS | <VerificarDocumen> ::= docIncom<AproProy> |
| C-> fD | <VerificarDocumen> ::= docCom<VerifTipoProy> |
| D->gE | <VerifTipoProy> ::=esHw <VerifTipoIns> |
| D-> hF | <VerifTipoProy> ::= esSw<ValidarMontoSw> |
| E-> jH | <VerifTipoIns> ::=apcid<ValidarMontoHw>|noApcid |
| E-> kG | <VerifTipoIns> ::= noApcid<G> |
| F-> nI | <ValidarMontoSw>::= monMay9 <EvaluarProy> |
| F-> iG | <ValidarMontoSw>::=montoMen9<G> |
| H-> mI | <ValidarMontoHw>::=montoMay5<EvaluarProy> |
| H-> lG | <ValidarMontoHw>::= monMen5 <G> |
| I-> pJ | <EvaluarProy> ::= jusTecCom <J> |
| I-> oS | <EvaluarProy> ::= jusTecInc<AproProy> |

## **Derivaciones y Árboles**

abceq

envDoc.sinRes.regRes.docIncom.insNoEnv

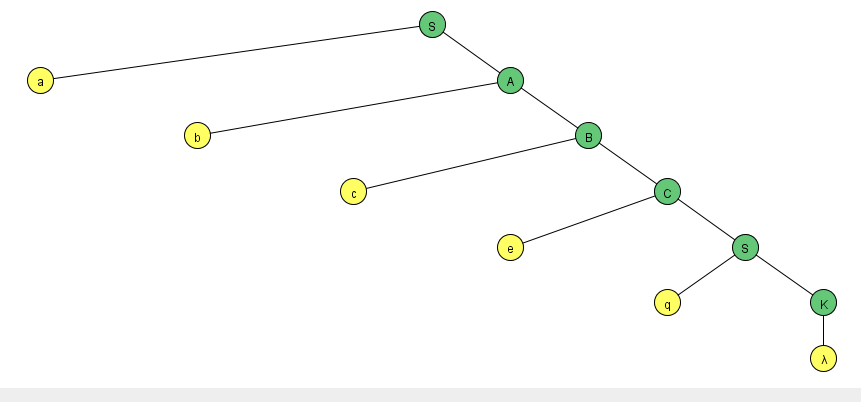
<AproProy> => envDoc<VerificarResponsable>

<AproProy> => envDoc.sinRes<AsignarRespon>

<AproProy> => envDoc.sinRes.regRes<VerificarDocumen>

<AproProy> => envDoc.sinRes.regRes.docIncom<AproProy>

<AproProy> => envDoc.sinRes.regRes.docIncom. insNoEnv



abcfhnoq

envDoc. sinRes.regRes.docCom.esSw.montoMay9. jusTecInc. insNoEnv

<AproProy> => envDoc<VerificarResponsable>

<AproProy> => envDoc.sinRes<AsignarRespon>

<AproProy> => envDoc.sinRes. regRes<VerificarDocumen>

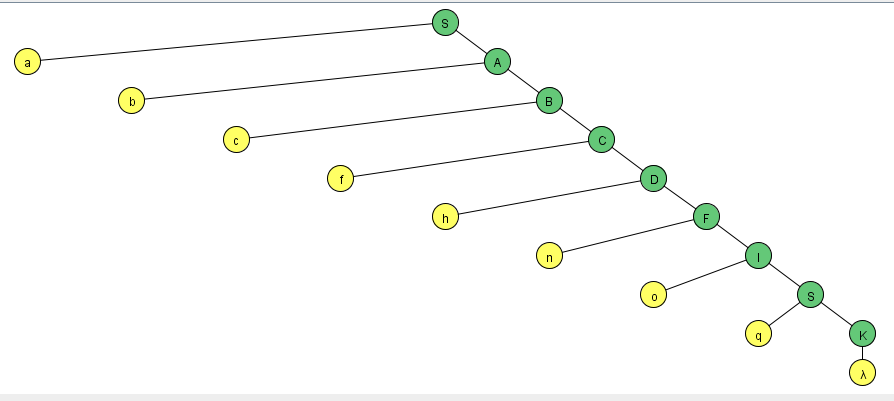
<AproProy> => envDoc.sinRes. regRes.docCom<VerifTipoProy>

<AproProy> => envDoc.sinRes. regRes.docCom.esSw<ValidarMontoSw>

<AproProy> => envDoc.sinRes. regRes.docCom.esSw.monMay9 <EvaluarProy>

<AproProy> => envDoc.sinRes. regRes.docCom.esSw.monMay9 jusTecInc<AproProy>

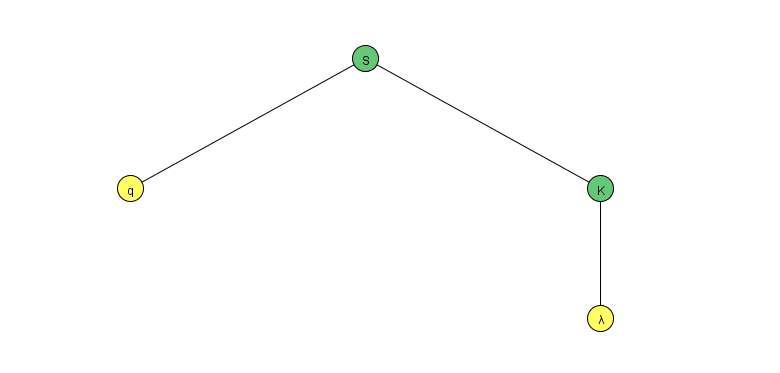
<AproProy> => envDoc.sinRes. regRes.docCom.esSw.monMay9 . jusTecInc. insNoEnv



.q

insNoEnv

<AproProy> => insNoEnv



.adfhnp

envDoc.conRes.docCom.esSw.monMay9.jusTecCom

<AproProy> => envDoc<VerificarResponsable>

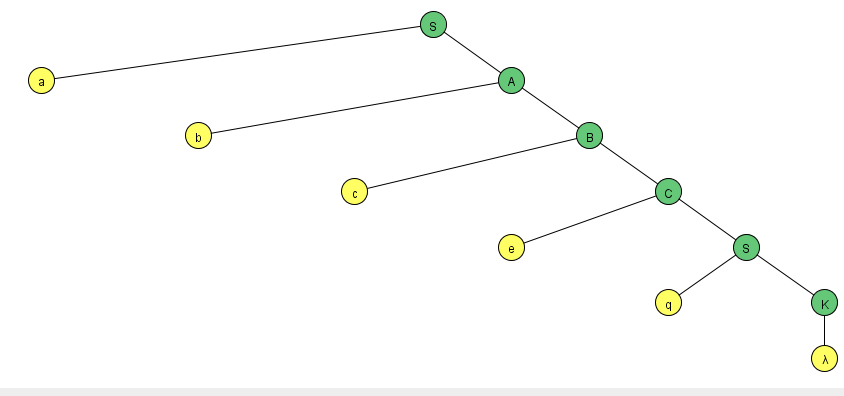
<AproProy> => envDoc. conRes <VerificarDocumen>

<AproProy> => envDoc. conRes .docCom <VerifTipoProy>

<AproProy> => envDoc. conRes .docCom. esSw <ValidarMontoSw>

<AproProy> => envDoc. conRes .docCom. esSw .MonMay9 <EvaluarProy>

<AproProy> => envDoc. conRes .docCom. esSw .MonMay9. jusTecCom



## **Transformación a FNC**

|  |  |
| --- | --- |
|  | **Propuesta de la gramática** |
| S-> aA | <AproProy> ::= envDoc<VerificarResponsable> |
| S->qK | <AproProy> ::= insNoEnv<K> |
| K-> l | <K> ::= l |
| J-> l | <J> ::= l |
| G-> l | <G> ::= l |
| A->bB | <VerificarResponsable> ::= sinRes<AsignarRespon> |
| A-> dC | <VerificarResponsable> ::= conRes<VerificarDocumen> |
| B->cC | <AsignarRespon>::= regRes<VerificarDocumen> |
| C->eS | <VerificarDocumen> ::= docIncom<AproProy> |
| C-> fD | <VerificarDocumen> ::= docCom<VerifTipoProy> |
| D->gE | <VerifTipoProy> ::=esHw <VerifTipoIns> |
| D-> hF | <VerifTipoProy> ::= esSw<ValidarMontoSw> |
| E-> jH | <VerifTipoIns> ::=apcid<ValidarMontoHw> |
| E-> kG | <VerifTipoIns> ::= noApcid<G> |
| F-> nI | <ValidarMontoSw>::= monMay9 <EvaluarProy> |
| F-> iG | <ValidarMontoSw>::=montoMen9<G> |
| H-> mI | <ValidarMontoHw>::=montoMay5<EvaluarProy> |
| H-> lG | <ValidarMontoHw>::= monMen5 <G> |
| I-> pJ | <EvaluarProy> ::= jusTecCom <J> |
| I-> oS | <EvaluarProy> ::= jusTecInc<AproProy> |

## **Eliminación de producciones vacías**

## **Identificación de variable nulas**

N0 = {G, J, K}

N1 = N0 .U vacío = {G, J, K}

## **Simplificar sin variables nulas**

|  |  |
| --- | --- |
|  | **Propuesta de la gramática** |
| S-> aA | <AproProy> ::= envDoc<VerificarResponsable> |
| S->qK|q | <AproProy> ::= insNoEnv<K>| insNoEnv |
| A->bB | <VerificarResponsable> ::= sinRes<AsignarRespon> |
| A-> dC | <VerificarResponsable> ::= conRes<VerificarDocumen> |
| B->cC | <AsignarRespon>::= regRes<VerificarDocumen> |
| C->eS | <VerificarDocumen> ::= docIncom<AproProy> |
| C-> fD | <VerificarDocumen> ::= docCom<VerifTipoProy> |
| D->gE | <VerifTipoProy> ::=esHw <VerifTipoIns> |
| D-> hF | <VerifTipoProy> ::= esSw<ValidarMontoSw> |
| E-> jH | <VerifTipoIns> ::=apcid<ValidarMontoHw> |
| E-> kG|k | <VerifTipoIns> ::= noApcid<G>| noApcid |
| F-> nI | <ValidarMontoSw>::= monMay9 <EvaluarProy> |
| F-> iG|i | <ValidarMontoSw>::=montoMen9<G>| montoMen9 |
| H-> mI | <ValidarMontoHw>::=montoMay5<EvaluarProy> |
| H-> lG|l | <ValidarMontoHw>::= monMen5 <G>| monMen5 |
| I-> pJ|p | <EvaluarProy> ::= jusTecCom <J> | jusTecCom |
| I-> oS | <EvaluarProy> ::= jusTecInc<AproProy> |

## **Producciones Inútiles**

Decimos que un símbolo *X* es *útil* para una gramática *G* = (*V,T,P,S*) si existe alguna derivación de la forma *S ⇒∗* α*X*β *⇒∗ w*, donde *w* pertenece a *T∗*. Si *X* no es útil, decimos que es *inútil*.

Se elimina los símbolos que no son generadores en primer lugar y luego eliminamos de la gramática resultante aquellos símbolos que no son alcanzables, tendremos sólo los símbolos útiles, como demostraremos.

Para nuestra gramática no existen **símbolos inútiles todas nuestras variables son derivables**

Derivables = {A,B,C,D,E,F,H,I,S}

## **Producciones Unitarias**

•Se identifica las producciones (reglas) unitarias

A -> B

•Se reemplaza la parte derecha de regla con el equivalente

No existen

## **Transformar a FNC**

Se analiza cuada de las reglas que tenga la forma S->AB, S->a para lo cual se van creando variable del ser el caso

S->AB

S->a

|  |  |
| --- | --- |
|  | **Propuesta de la gramática** |
| S-> aA | <AproProy> ::= envDoc<VerificarResponsable> |
| S->qK | <AproProy> ::= insNoEnv<K> |
| S-> q | <AproProy> ::= insNoEnv |
| A->bB | <VerificarResponsable> ::= sinRes<AsignarRespon> |
| A-> dC | <VerificarResponsable> ::= conRes<VerificarDocumen> |
| B->cC | <AsignarRespon>::= regRes<VerificarDocumen> |
| C->eS | <VerificarDocumen> ::= docIncom<AproProy> |
| C-> fD | <VerificarDocumen> ::= docCom<VerifTipoProy> |
| D->gE | <VerifTipoProy> ::=esHw <VerifTipoIns> |
| D-> hF | <VerifTipoProy> ::= esSw<ValidarMontoSw> |
| E-> jH | <VerifTipoIns> ::=apcid<ValidarMontoHw> |
| E-> kG | <VerifTipoIns> ::= noApcid<G> |
| E-> k | <VerifTipoIns> ::= noApcid |
| F-> nI | <ValidarMontoSw>::= monMay9 <EvaluarProy> |
| F-> iG | <ValidarMontoSw>::=montoMen9<G> |
| F-> i | <ValidarMontoSw>::= montoMen9 |
| H-> mI | <ValidarMontoHw>::=montoMay5<EvaluarProy> |
| H-> lG | <ValidarMontoHw>::= monMen5 <G> |
| H-> l | <ValidarMontoHw>::= monMen5 |
| I-> pJ | <EvaluarProy> ::= jusTecCom <J> |
| I-> p | <EvaluarProy> ::= jusTecCom |
| I-> oS | <EvaluarProy> ::= jusTecInc<AproProy> |

|  |  |
| --- | --- |
|  | **Propuesta de la gramática transformada** |
| S-> XaA | <AproProy> ::= <Xa><VerificarResponsable> |
| S->XqK | <AproProy> ::= <Xq><K> |
| S-> q | <AproProy> ::= insNoEnv |
| A->XbB | <VerificarResponsable> ::= <AsignarRespon> |
| A-> XdC | <VerificarResponsable> ::= conRes<VerificarDocumen> |
| B->XcC | <AsignarRespon>::= <Xc><VerificarDocumen> |
| C->XeS | <VerificarDocumen> ::= <Xe><AproProy> |
| C-> XfD | <VerificarDocumen> ::= <Xf><VerifTipoProy> |
| D->XgE | <VerifTipoProy> ::=< Xg ><VerifTipoIns> |
| D-> XhF | <VerifTipoProy> ::= <Xh><ValidarMontoSw> |
| E-> XjH | <VerifTipoIns> ::= <Xj> <ValidarMontoHw> |
| E-> XkG | <VerifTipoIns> ::= <Xk> <G> |
| E-> k | <VerifTipoIns> ::= noApcid |
| F-> Xn I | <ValidarMontoSw>::= <Xn> <EvaluarProy> |
| F-> Xi G | <ValidarMontoSw>::=<Xi><G> |
| F-> i | <ValidarMontoSw>::= montoMen9 |
| H-> XmI | <ValidarMontoHw>::=<Xm> <EvaluarProy> |
| H-> XlG | <ValidarMontoHw>::= <Xl ><G> |
| H-> l | <ValidarMontoHw>::= monMen5 |
| I-> Xp J | <EvaluarProy> ::= <Xp> <J> |
| I-> p | <EvaluarProy> ::= jusTecCom |
| I-> Xo S | <EvaluarProy> ::= <Xo> <AproProy> |
| Xa->a | <Xa> ::= envDoc |
| Xq->q | <Xq> ::= insNoEnv |
| Xb->b | <Xb>  ::= sinRes |
| Xc->c | <Xc>::= regRes |
| Xd->d | < Xd>::= conRes |
| Xe->e | < Xe>::= docIncom |
| Xf->f | <Xf>::= docCom |
| Xg->g | <Xg> ::= esHw |
| Xh->h | <Xh> ::= esSw |
| Xi->i | <Xi> ::= montoMen9 |
| Xj->j | <Xj> ::= apcid |
| Xk->k | <Xk> ::= noApcid |
| Xl->l | <Xl> ::= monMen5 |
| Xm->m | <Xm> ::= montoMay5 |
| Xn->n | <Xn> ::= monMay9 |
| Xo->o | <Xo> ::= jusTecInc |
| Xp->p | <Xp> ::= jusTecCom |

## **Bibliografía:**

* Contreras Zambrano, H. (2013). Gramáticas Libres de Contexto. Retrieved from <https://www.youtube.com/watch?v=V8a2-O20Ebo>
* Derivaciones para gramáticas libres de contexto. (2011). Retrieved from <https://www.youtube.com/watch?v=qkJQpTzYPqs>
* Faccioni, H. (2014). Simplificação de Gramáticas Livres de Contexto, Forma normal de Chomsky e Forma Normal de Greibach. Retrieved from <https://www.youtube.com/watch?v=juo5WmTJn5E>
* Gramaticas libres de contexto.mp4. (2011). Retrieved from <https://www.youtube.com/watch?v=eHo4Qlqoc3k>
* Lezcano, L. (2011). Video Tutorial - Gramáticas con JFLAP. Retrieved from <https://www.youtube.com/watch?v=WhZ2HxK0M5c>