COMPILADOR

Un compilador es un programa que convierte o traduce el código fuente de un programa hecho en lenguaje de alto nivel, a un lenguaje de bajo nivel (lenguaje de máquina).

Programa que convierte o traduce de un lenguaje de programación a de máquina.

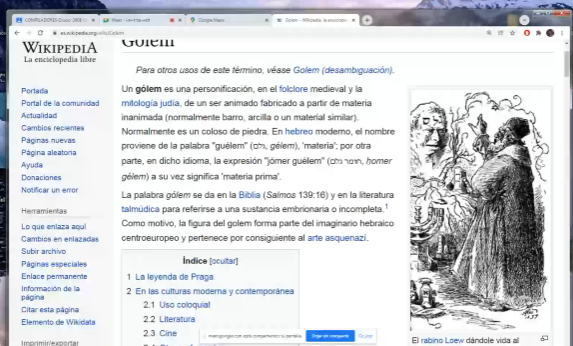
Interfaz de usuario gráfica, Texto, Aplicación, Correo electrónico

Descripción generada automáticamente

Los lenguajes naturales tienen ambigüedad

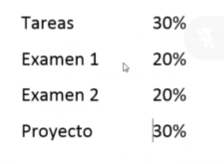
Lenguajes formales no tienen ambigüedad

-Clase 2-

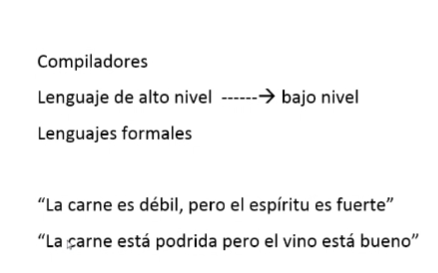


-Clase 3-

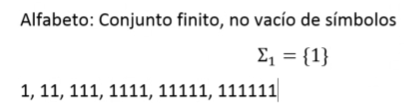
Escala de evaluacion



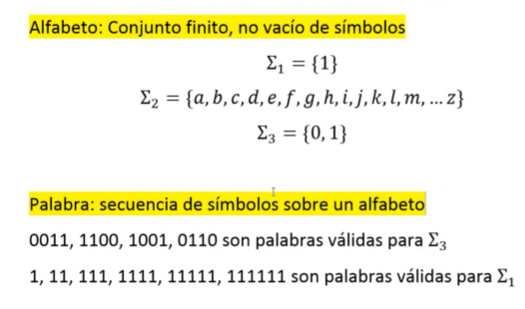
Los compiladores son lenguajes formales (No tiene problemas con la ambiguedad).



Alfabeto= Conjunto finito, no vacio de simbolos. Con el cual se pueden formar palabras



Palabra= Secuencia de simbolos sobre un alfabeto.

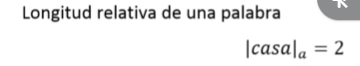


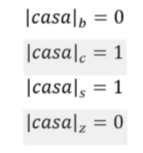
Longitud de una palabra

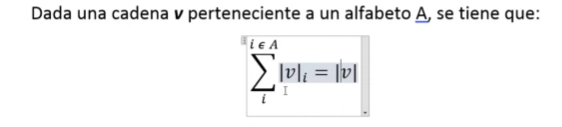
|casa|= La longitud de la palabra es 4.

|hoy|= 3

|λ| = 0 –Lamda repesenta cadena vacia.

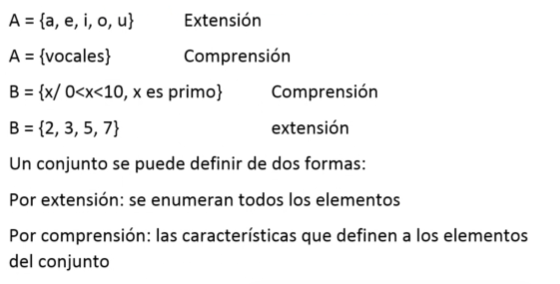
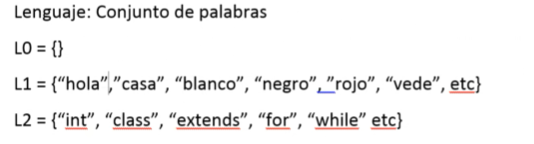


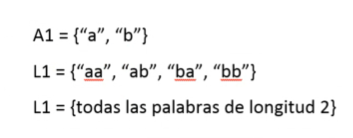


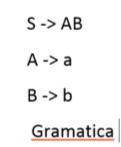


Un SIMBOLO= es un caracter, pueden o no tener significado, normalmente si los tienen.

Lenguaje= es un conjunto de palabras.

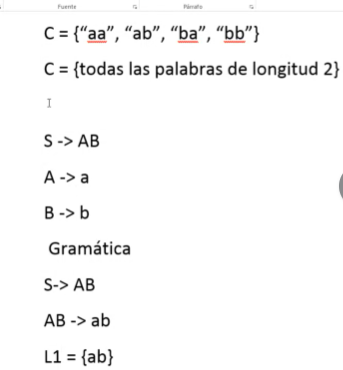




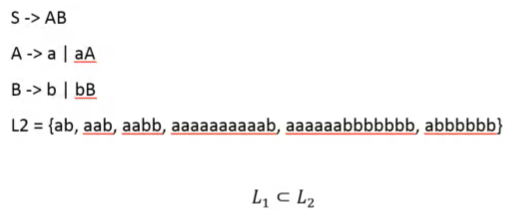


Gramatica= estencion de un lenguaje.

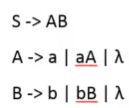
Una gramatica y un lenguaje son equivalentes.

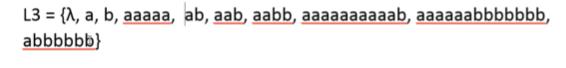


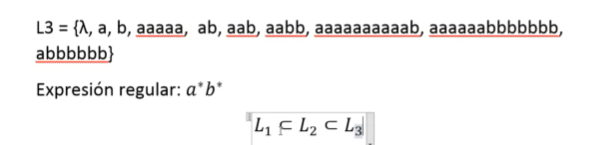
L1- expresion regular ab

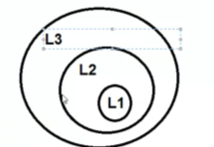


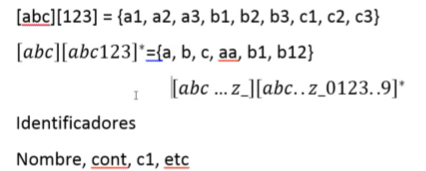
L2- expresion regular. a+b+



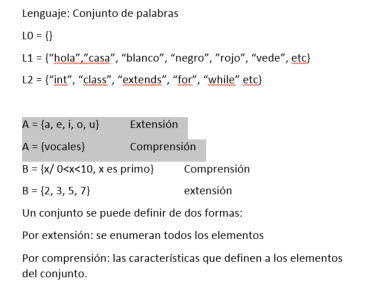








**-------------CLASE 4---------------**



Conjunto se puede definir por extension o por compresion.

Cuando es infinito solo se puede definir por compresion.

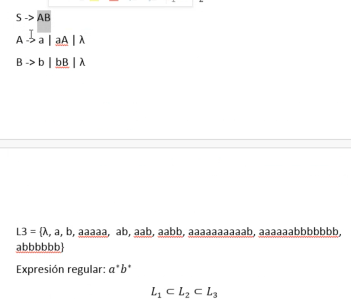


Gramatica nos define por extension un lenguaje.



Lenguaje por compresion.

La gramatica regular, lenguaje regular y expresion regular son equivalentes.



Gramatica; tiene 3 conjuntos.

**Conjunto de no terminales o ramas(N);** conjunto de simbolos, que se representan normalmente con Mayusculas, sirven para producir otros simbolos (Terminales o No terminales) y no podemos terminar en un no terminal.

N={S,A,B}

**Conjunto de terminales(T);** un conjunto de simbolos(normalmente se representan con Minusculas), se producen a partir de los no terminales, estos ya no producen ningun simbolo, tambien se les denomina hojas.

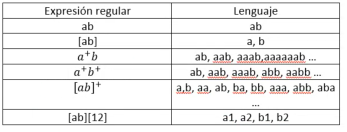
T={a,b}

**Producciones(P);** Son un conjunto de reglas que indican como se transforman los No terminales;



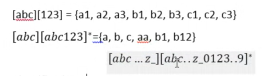


GRAMATICAS REGULARES E INDEPENDIENTES DEL CONTEXTO.



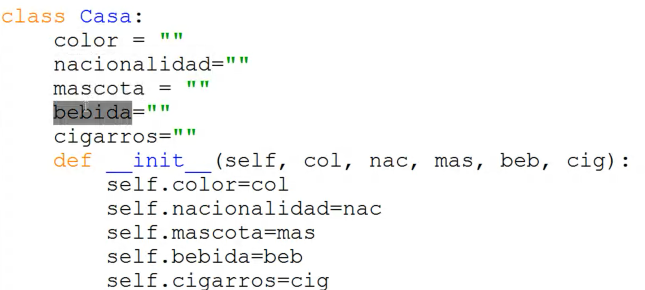
[]- significa opcion(a o b).

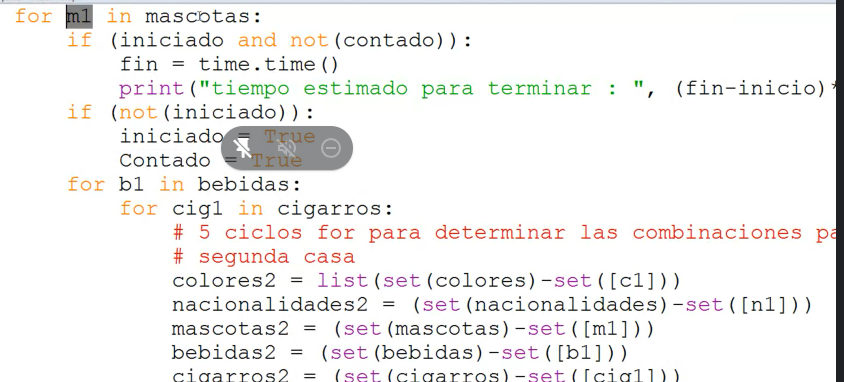
+- una o mas a, seguida de una b.



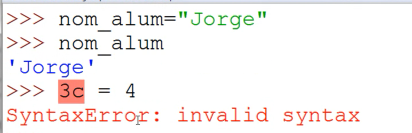


Es la expresion regulas de todos los identificadores.



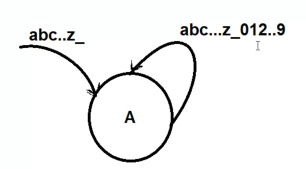


Identificadores; c1,c2,col,nac,mas,color,nacionalidad,etc.... Todos los nombres de variables, metodos, funciones... cumplen con esta regla.



3c no esta dentro del lenguaje de los identificadores.

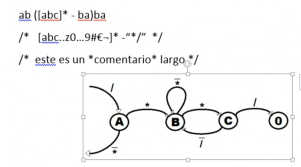






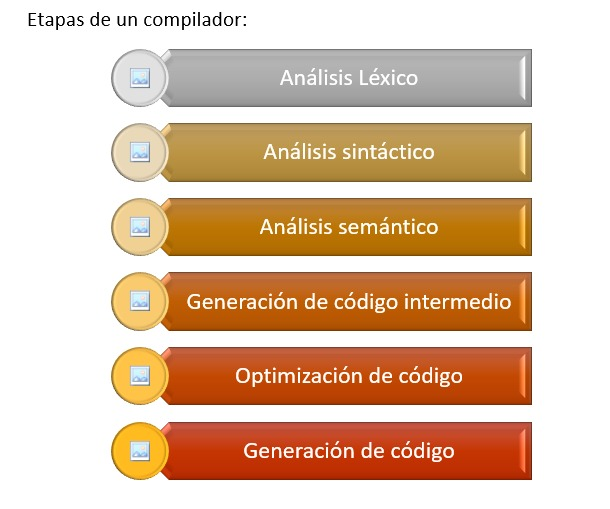


/\*asjdkajsdhkajds\*/ - Todos los comentarios en java o c, cumplen con esto.



Todos los lenguajes de programacion son lenguajes formales y se pueden programar con automatas.

**Etapas de un compilador**

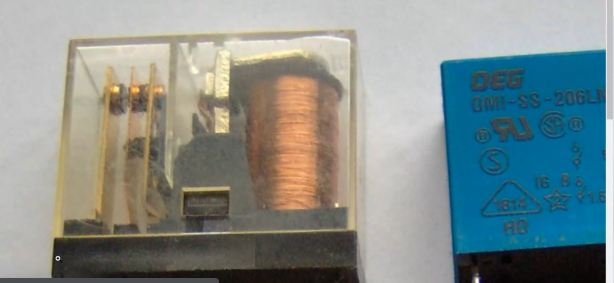


Analisis= Descomponer un todo en sus partes

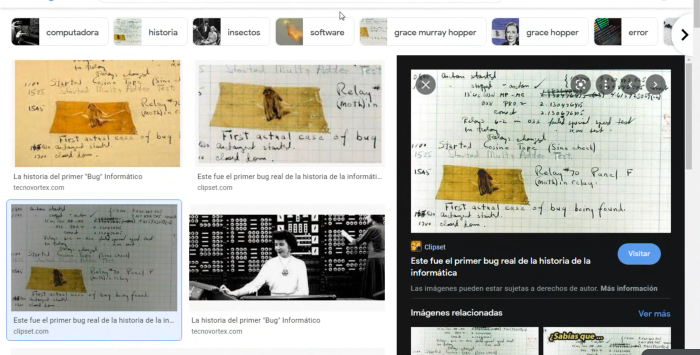
Sintesis= Unir las partes para formar un todo.

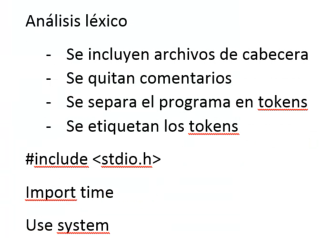
Resumen= primero se hace el analisis luego la sintesis y ya se crea el resumen.

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



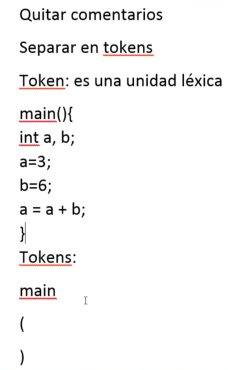
Primer bug.





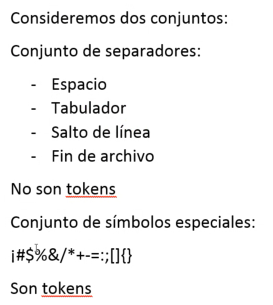
Separar por Tokens;

Token= Es una unidad lexica.

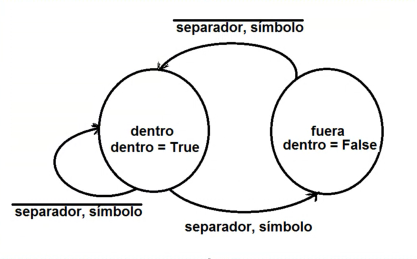


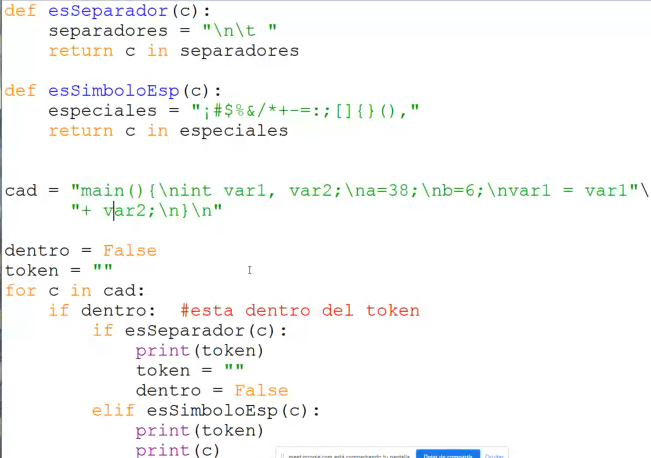


Inserting image...



Todo lo que no sea simbolo especial ni separador es un token y esta delimitado por un simbolo especial o un separador.





**CLASIFICAR LOS TOKENS**

main palabra reservada

( simbolo

) simbolo

{ simbolo

int palabra reservada(tipo de dato)

var1 identificador

, simbolo

var2 identificador

; simbolo

var1 identificador

= simbolo

38 numero entero

; simbolo

var2 identificador

= simbolo

6 numero entero

; simbolo

var1 identificador

= simbolo

var1 identificador

+ simbolo (operador)

var2 identificador

; simbolo

} simbolo

**Pasos que hemos seguido; (Analisis lexico)**

* Quitamos comentarios
* Lo separamos en piezas (tokens)
* Etiquetamos los tokens