

## Equipo !Jitters

### 1. Datos Generales de la Reunión:

#### Medios que utilizamos para reunirnos:

- **Unidad en drive**, para así trabajar en conjunto en línea. (Docs y Collab)
- Tuvimos llamada por **Google Hangouts**
- Usamos **Whatsapp** para estar en contacto.

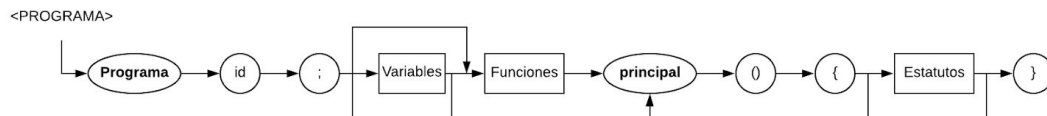
#### Fecha de la reunión: ¿Cuándo nos reunimos o interactuamos?

- 9 de abril
- 12 de abril
- 13 de abril

### 2. Avances:

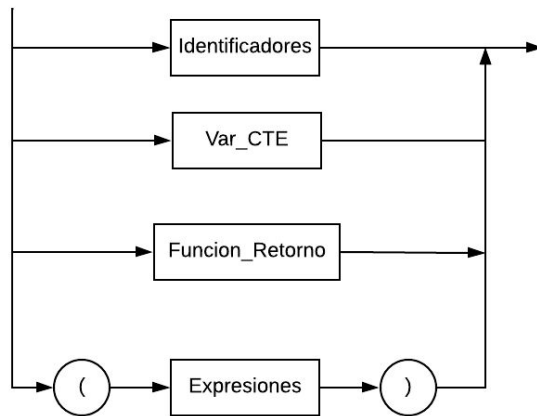
#### Diagramas de sintaxis:

Se analizó la gramática del lenguaje. Para mejor visualización hicimos los siguientes diagramas:

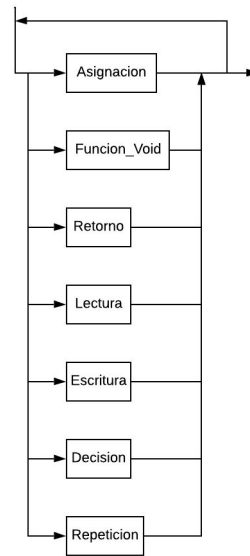




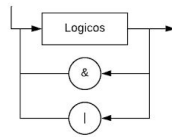
<TERMINOS>



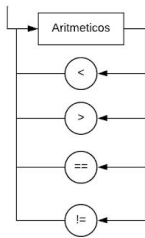
<ESTATUTOS>



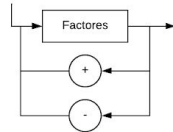
<EXPRESION>



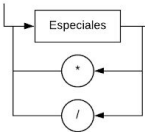
<LOGICOS>



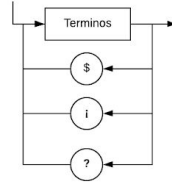
<ARITMETICOS>



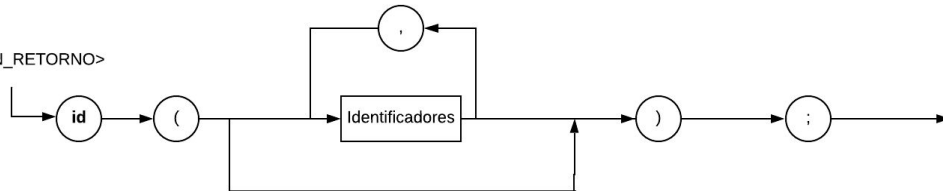
<FACTORES>



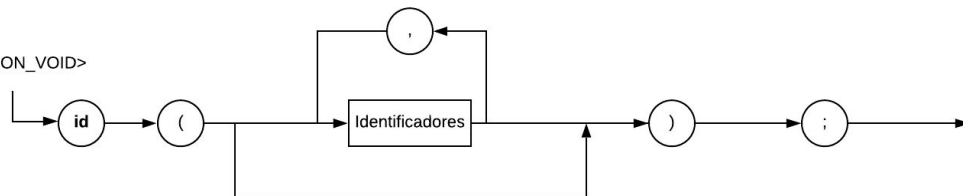
<ESPECIALES>



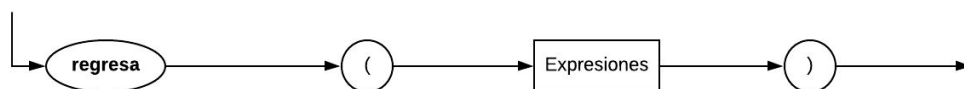
<FUNCION\_RETORNO>

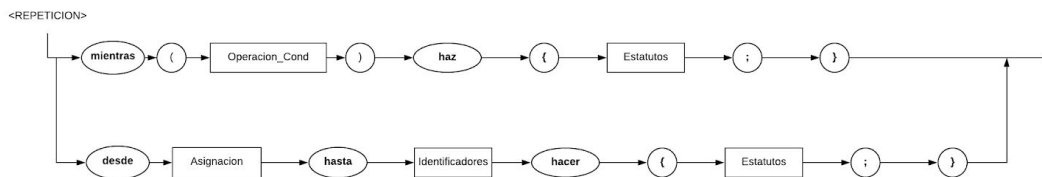
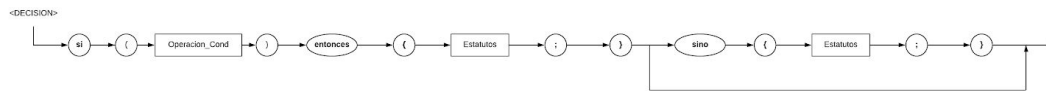
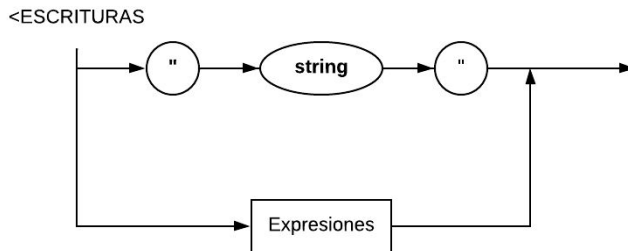
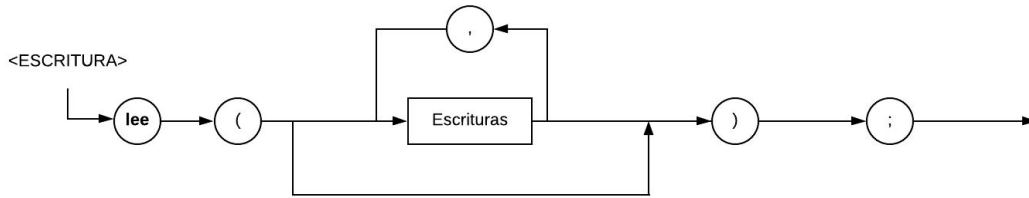
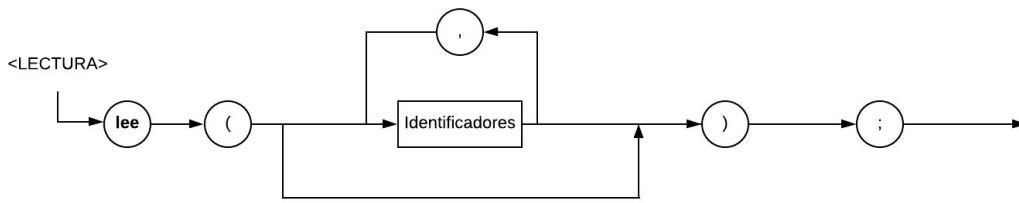


<FUNCION\_VOID>



<RETORNO>





Para ver los diagramas en grande puede acceder este [enlace](#).

## Reglas de gramática:

Se realizó el primer avance del analizador léxico y sintaxis. Hay algunas dudas en cuanto las operaciones condicionales, sin embargo se generó la gramática de la mejor manera posible a disposición de cambio.

Tuvimos cita con la maestra y nos resolvió la duda de las operaciones condicionales. Optamos por contar tanto operaciones normales, como condicionales, en uno solo y dejar que el analizador sintáctico nos llore. Asimismo, marcamos prioridades en la gramática.

Así quedaron:

<Programa> → **Programa** id ; <Variables> <Funciones> **principal ()** { <Estatuto> }

<Variables> → **var** <Tipo> : <Lista\_Ids>; <Variables\_1> | ε

<Variables\_1> → <Tipo> : <Lista\_Ids>; <Variables\_1> | ε

<Lista\_Ids> → <Identificadores> <Lista>

<Lista> → , <Identificador> <Lista> | ε

<Funciones> → **funcion** <Tipo\_retorno> id ( <Parametros> ) ; <Variables>  
{ <Estatuto> } <Funciones> | ε

<Tipo\_retorno> → **int** | **float** | **char** | **void**

<Tipo> → **int** | **float** | **char**

<Parametros> → <Tipo> : id <Variables\_2> | ε

<Variables\_2> → , <Tipo> : id <Variables\_2> | ε

<Estatuto> → <Estatuto\_1><Estatuto> | ε

<Estatuto\_1> → <Asignacion> | <Funcion\_Void> | <Retorno> | <Lectura> |  
<Escritura> | <Decision> | <Repeticion>

<Asignacion> → <Identificadores> = <Expresiones> ;

<Identificadores> → id [num] | id [num][num] | id

<Terminos> → <Identificadores> | <Funcion\_Retorno> | <Var\_CTE> |  
<( <Expresiones> )>

<Especiales> → <Terminos> | <Terminos> <Especiales\_1> <Especiales>

<Especiales\_1> → \$ | i | ?

<Factores> → <Especiales> | <Especiales> <Factores\_1> <Factores>

<Factores\_1> → \* | /

<Aritmeticos> → <Factores> | <Factores> <Aritmeticos\_1> <Aritmeticos>

<Aritmeticos\_1> → + | -

<Logicos> → <Aritmeticos> | <Aritmeticos> <Logicos\_1> <Logicos>

<Logicos\_1> → < | > | == | !=

$\langle \text{Expresiones} \rangle \rightarrow \langle \text{Logicos} \rangle \mid \langle \text{Logicos} \rangle \langle \text{Expresiones}_1 \rangle \langle \text{Expresiones} \rangle$   
 $\langle \text{Expresiones}_1 \rangle \rightarrow \& \mid \mid$

<Funcion\_Retorno> → id(lista\_ids) | id()

```
<Funcion_Void> → id(lista_ids); | id();
```

<Retorno> → **regresa**(<Expresiones>)

<Lectura> → **lee**(<Lista\_Ids>);

<Escritura> → **escribe**( <Escritura\_1>);

$$\langle \text{Escritura}_1 \rangle \rightarrow \text{"string"} \langle \text{Escritura}_2 \rangle \mid \langle \text{Expresiones} \rangle \langle \text{Escritura}_2 \rangle$$
$$\langle \text{Escritura}_2 \rangle \rightarrow , \langle \text{Escritura}_1 \rangle \mid \varepsilon$$

<Decision> → **si** (<Operacion\_Cond>) **entonces** { <Estatuto> } <Decision\_1>

$$\langle \text{Decision\_1} \rangle \rightarrow \mathbf{sino} \{ \langle \text{Estatuto} \rangle \} \mid \varepsilon$$

<Repeticion> → <Repeticion\_Cond> | <Repeticion\_No\_Cond>

<Repeticion\_Cond> → **mientras** (<Operacion\_Cond>) **haz** { <Estatuto> ; }

```
<Repeticion_No_Cond> → desde <Asignacion> hasta <Expresiones> hacer {  
    <Estatuto> }
```

<Var\_CTE> → Var\_Int | Var\_Float | Var\_Char

$$\text{id} \rightarrow [a-zA-Z]^+([a-zA-Z0-9\_])^*$$

String  $\rightarrow$  [Printable Characters]\*

$$\text{Var\_Int} \rightarrow [0-9]^+$$
$$\text{Var\_Float} \rightarrow [0-9]+(.[0-9]+)^*$$

Var\_Char  $\rightarrow$  '[a-zA-z]'

Num  $\rightarrow [0-9]^+$

## **Codificación:**

Se implementaron las reglas creadas para !Jitters usando *pyparsing*. Se trabajó en conjunto utilizando Collab, pero después se agregaron los avances al gitHub.

¿Funciona? SIIII, SI FUNCIONA HASTA LO QUE SABEMOS. :)

¿Faltó una parte? DEL PRIMER AVANCE, NO.

¿Corrieron suficientes test-cases? Corrimos los suficientes para poder identificar si había algún error, sin embargo, no fueron test cases exhaustivos.

¿Tienen todavía ambigüedades en la gramática? Creemos que no, porque no hay manera de generar dos cosas iguales de distintas maneras. Con las pruebas que hicimos no vimos que hubiera alguna ambigüedad.

¿Pueden marcar errores particulares o solo uno genérico (syntax error)? Nos marca errores particulares pero no en el lugar correcto, es decir sabe exactamente en dónde está el error pero no nos dice correctamente lo que debería de haber, nadamas nos dice que esperaba que no haya nada o que estuviera lo que le seguía a la regla donde esta el error.