# Documentación tokenizador.

Integrantes:

Carlos Antonio Minero Dubón 00058016

Raul Antonio Granados Hernandez 00138816

Marcelo Alexander Martínez Montenegro 00044616

José Alexander García Valladares 00062816

### Generalidades:

El tokenizador fue desarrollado en JavaScript con notación ECMAScript 6, solamente se hace uso de la consola, el lenguaje objetivo el cual se tokeniza es **Kotlin.** Los símbolos que toma en cuenta el tokenizador son los siguientes:

#### Palabras reservadas:

Int

Float

Char

String

return

fun

if

else

do

while

for

void

:

println

print

#### Operadores:

+

-

/

\*

#### Operadores de comparación:

>

<

#### Identificadores:

Ejemplos:

* variable1
* \_variable2
* variable\_constante2

#### Símbolos de apertura:

(

{

[

#### Símbolos de cierre:

)

}

]

#### Formato de comentarios:

Comentario de una línea:

// Cualquier\_comentario\_aqui

Comentarios de bloque

/\*Comentario\_de\_bloque\_aqui\*/

#### Otros símbolos contemplados:

var | var : Asignadores

; : Fin de instrucción

= : Símbolo de asignación

#### Consideraciones del tokenizador:

* Las cadenas de texto no deben contener espacios vacíos.
* Los comentarios de línea y de bloque no deben contener espacios.
* El código a tokenizar debe estar en una sola línea.
* Los bloques de código deben estar separados por espacios.
* Puede visualizar el tokenizador directamente desde: <https://ragranados.github.io/TokenizadorTEO/>

#### 

#### Símbolos no reconocidos:

$

|

^

~

°

¬

!

#

%

&

¡

¨

´

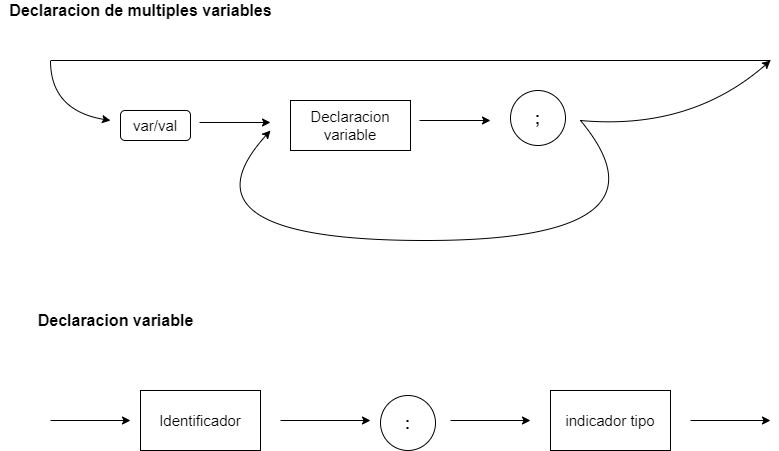
`

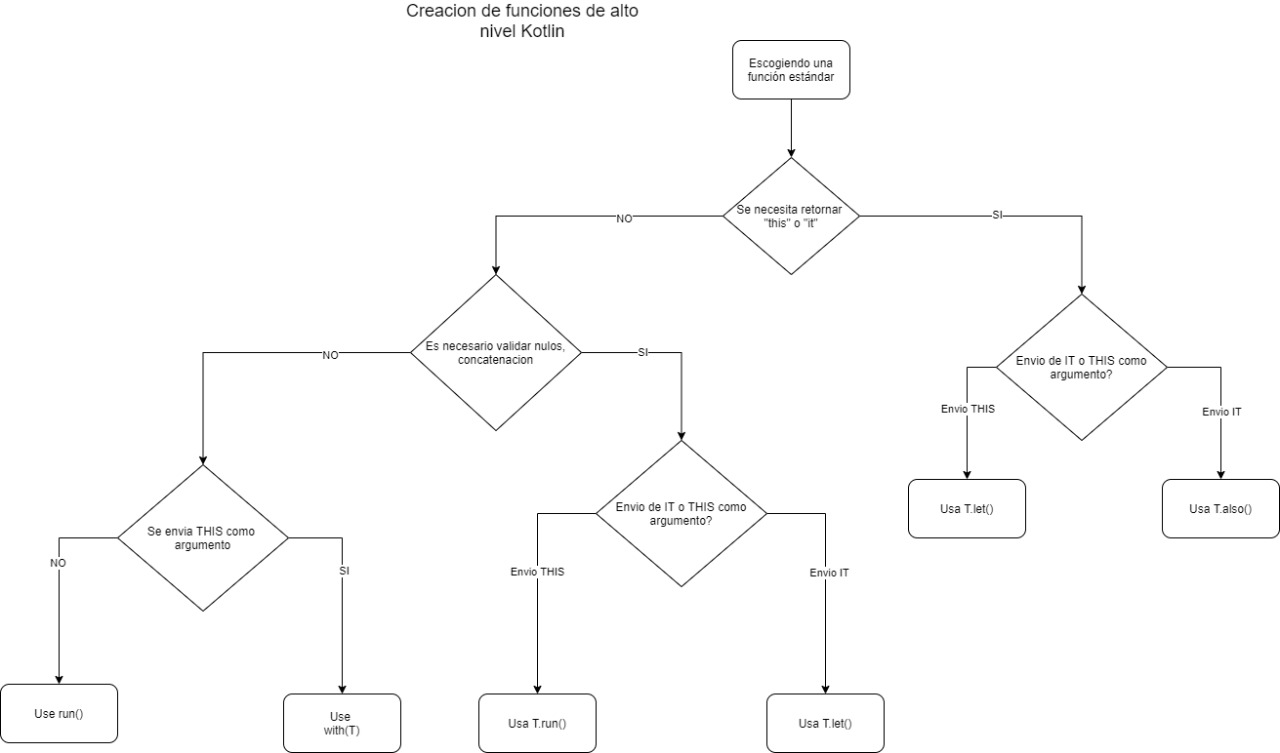
#### Ejemplos de códigos a tokenizar:

"fun main(){ val x:String=2; val y=\"Hello\"; //Comentario println(y+x); }"

"fun main(){ var x:Char=\'A\'; val y=\"BC\"; //Comentario println(x+y); }"

**Diagramas de sintaxis:**





Para la creación de los diagramas se hizo uso de la herramienta Draw.io.