

*Universidad de San Carlos de Guatemala USAC.  
División de Ciencias de la Ingeniería.  
Centro Universitario de Occidente CUNOC.  
Compiladores 2*



**Estudiante**

*Elvis Lizandro Aguilar Tax*

**Carnet**

*201930304*

**Manual de Tecnico**

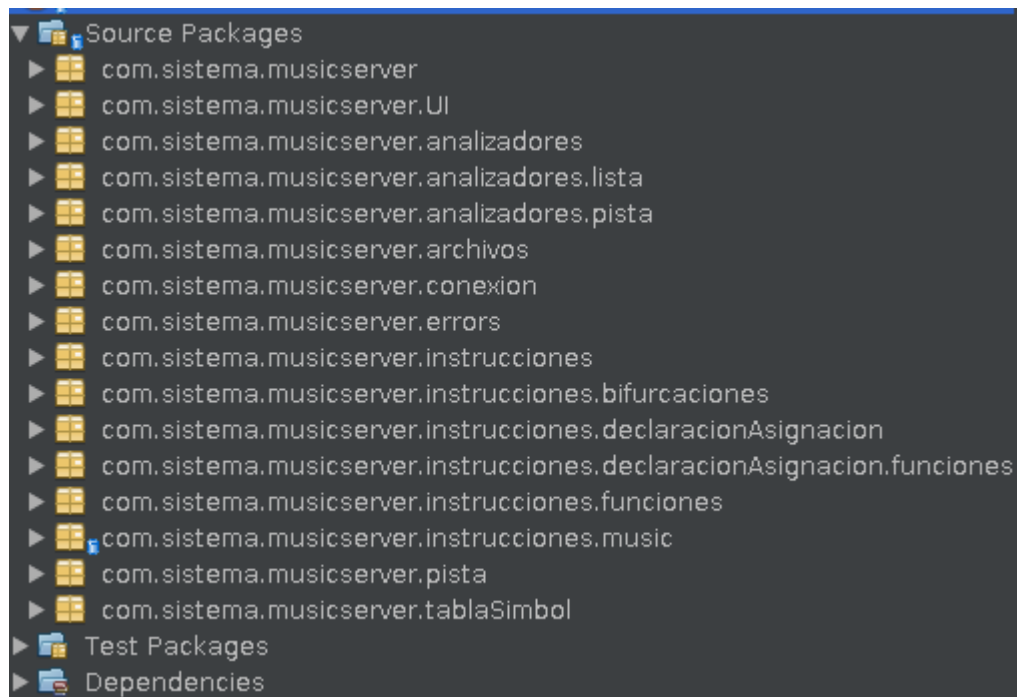
## **Herramientas de Desarrollo**

- Lenguaje :
  - java**  
versión: 1.8 — 11.0.2
  - kotlin**  
version: 1.8 –11.0.2
- Analizador Lexico
  - **Jflex**
    - version: 1.8.2
    - documentacion oficial: <https://www.jflex.de/>
- Analizador sintáctico
  - Jcup
    - versión: 0.11b
    - documentacion oficial: <http://www2.cs.tum.edu/projects/cup/>
- IDE
  - Netbeans version 12.0

## **Sistema operativo**

- OS: Manjaro Linux X86\_64

## **Organización del proyecto (Paquetes)**



el paquete principal es: `com.sistema.muscservice`

en el mismo esta la clase principal para que el programa se inicializa, a sí mismo:

**UI:** encargado de todo las funciones, del frontend

**analizadores:** en esta carpeta se encuentra los analizadores, léxicos y sintácticos para el funcionamiento correcto del analizadores de el lenguaje

**Archivos:** el de archivos, se encarga de la persistencia de los datos binarios

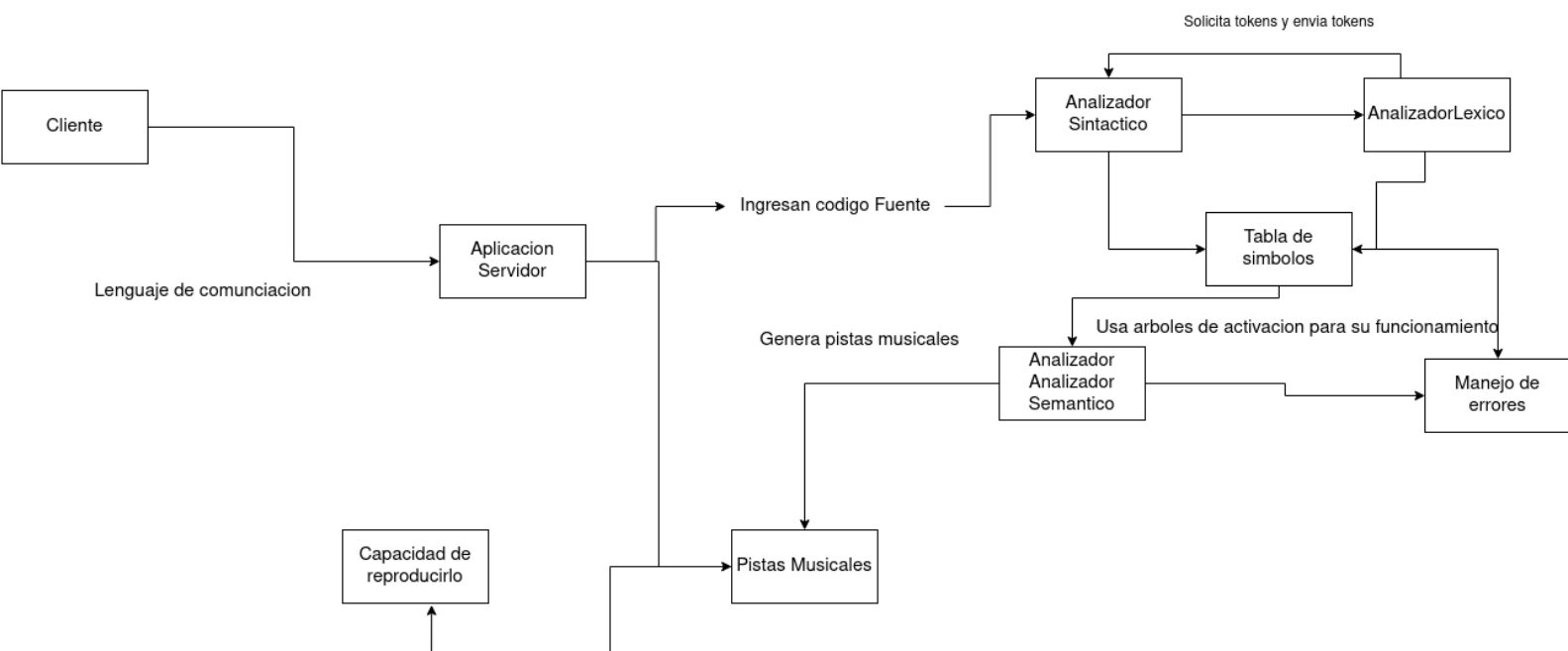
**conexión:** este se encarga de la conexión con sockets para el cliente

**errores:** este es el encargado de manejar y reportar los errores.

**Instrucciones:** encargado de las clases que representan los nodos de activación, para el analisis semántico, cada sub-carpeta maneja su propio nodo y funciones, como el de bifurcaciones.

**pista:** encargado de manejar la pista, es decir este controla todas las otras clases y en este se va almacenar las funciones, nodos, tabla de símbolos.

**tabla symbol:** carpeta para el manejo de la tabla de símbolos



## Analisis Lexico

Salto de línea y espacios en blanco, que bien es cierto, son ignorados, se deben de indicar los patrones necesarios para que el analizador léxico sepa que se debe de ignorar, las expresiones regulares son las siguientes

**LineTerminator** = `\r\n\r\n`

**WhiteSpace** = `{LineTerminator} | [ \t\f]`

Lo siguiente que debemos ignorar son los comentarios, de los cuales se muestran ejemplos a continuación:

**InputCharacter** = `[^\r\n]`

**CommentContent** = `([^\-|\+|\"->"]|\\-|>|\\-*[w<,;\\"])*`

**EndOfLineComment** = `"!!" {InputCharacter}* {LineTerminator}?`

**BlockComment** = `"<!--" {CommentContent} -* "-->"`

**Comment** = `{EndOfLineComment} | {BlockComment}`

Posteriormente se utilizan los siguientes patrones para números enteros y números con coma flotante o decimales:

**DECIMAL** = ([0-9]+[.]([0-9]+))

**NUM\_ENTERO** = [0-9]+

algunos caracteres que se usaron:

**LLAVEA** = "{"

**LLAVEC** = "}"

**PARENTESISA** = "("

**PARENTESISB** = ")"

**COMA** = ","

**PUNTOCOMA** = ";"

**IGUAL** = "="

**CORCHETA** = "["

**CORCHETC** = "]"

**COMILLAS** = "\""

**DOPUNTO** = ":"

**DIGONALB** = "\_"

Operadores relacionales

**EQUALS** = ({IGUAL}{IGUAL})

**NOTEQUALS** = "!="

**MENORQ** = "<"

**MAYORQ** = ">"

**MAYOROI** = ">="

**MENOROI** = "<="

**ISNULO** = "!!"

Operadores lógicos

**NOT** = "!"

**OR** = "||"

**AND** = "&&"

**NAND** = "!&&"

**NOR** = "!!||"

**XOR** = "&|"

operadores aritméticos

**POR** = "\*"

**DIVISION** = "/"

**MENOS** = "-"

**MAS = "+"**  
**MODULO = "%"**  
**POTENCIA = "^"**

palabras reservadas del lenguaje

**PISTA = "pista"|"Pista"**  
**KEEP = "keep"|"Keep"**  
**EXTIENDE = "extiende" | "Extiende"**  
**ENTERO = "entero" | "Entero"**  
**DOBLE = "doble"|"Doble"**  
**BOOLEAN = "boolean" | "Boolean"**  
**CARACTER = "caracter" | "Caracter"**  
**VERDADERO = "verdadero" | "Verdadero" | "true" | "True"**  
**FALSO = "falso" | "Falso" | "false" | "False"**  
**CONT\_CARACTER = "" "##"? [^] ""**  
**CADENA = "cadena"|"Cadena"**  
**VAR = "var"|"Var"**  
**ARREGLO = "arreglo"|"Arreglo"**  
**SI = "si"|"Si"**  
**SINO = "sino"|"Sino"**  
**SWITCH = "switch"|"Switch"**  
**CASO = "caso"|"Caso"**  
**DEFAULT = "default" | "Default"**  
**SALIR = "salir" | "Salir"**  
**PARA = "para"|"Para"**  
**MIENTRAS = "mientras"|"Mientras"**  
**HACER = "Hacer"|"hacer"**  
**CONTINUAR="continuar"|"Continuar"**  
**RETORNA = "retorna"|"Retorna"**  
**REPRODUCIR = "reproducir"|"Reproducir"**  
**ESPERAR = "esperar"|"Esperar"**  
**ORDENAR = "ordenar"|"Ordenar"**  
**SUMARIZAR = "sumarizar"|"Sumarizar"**  
**LONGITUD = "longitud"|"Longitud"**  
**MENSAJE = "mensaje"|"Mensaje"**  
**PRINCIPAL = "principal"|"Principal"**  
**ASCENDENTE = "ascendente"|"Ascendente"**  
**DESCENDENTE = "descendente"|"Descendente"**  
**PARES = "pares"|"Pares"**  
**IMPARES = "impares"|"Impares"**  
**PRIMOS = "primos"|"Primos"**  
  
**NOTA\_DO = "Do"**

NOTA\_DO\_S = "Do#"  
NOTA\_RE = "Re"  
NOTA\_RE\_S = "Re#"  
NOTA\_MI = "Mi"  
NOTA\_FA = "Fa"  
NOTA\_FA\_S = "Fa#"  
NOTA\_SOL = "Sol"  
NOTA\_SOL\_S = "Sol#"  
NOTA\_LA = "La"  
NOTA\_LA\_S = "La#"

## **Analizador sintactico**

### **Gramatica**

presidencia 👍

precedence left OR, XOR, NOR;  
precedence left AND, NAND;  
precedence right NOT;  
precedence left EQUALS, NOTEQUALS, MAYORQ, MAYOROI, MENORQ, MENOROI;  
precedence right ISNULO;  
precedence left MENOS, MAS;  
precedence left POR, DIVISION, MODULO;  
precedence left POTENCIA;

inicio ::= def\_pista  
|  
;

sentencia\_global ::= def\_variable  
| asignacion\_var  
| def\_funciones:fun  
| def\_fun\_principal:funPrin  
| def\_fun\_arreglo  
;

/\*Gramatica de operacion aritmetica\*/  
operation ::= operation:opLeft MAS operation:opRight  
| operation:opLeft MENOS operation:opRight

| operation:opLeft POR operation:opRight  
  
| operation:opLeft DIVISION operation:opRight  
  
| operation:opLeft MODULO operation:opRight  
  
| operation:opLeft POTENCIA operation:opRight  
| operation:opLeft MAYORQ operation:opRight  
  
| operation:opLeft MENORQ operation:opRight  
| operation:opLeft MAYOROI operation:opRight  
| operation:opLeft MENOROI operation:opRight  
| operation:opLeft NOTEQUALS operation:opRight  
| operation:opLeft EQUALS operation:opRight  
| ISNULO operation:opRight  
| NOT operation:opRight  
| operation:opLeft OR operation:opRight  
| operation:opLeft AND operation:opRight  
| operation:opLeft NAND operation:opRight  
| operation:opLeft NOR operation:opRight  
| operation:opLeft XOR operation:opRight  
| terminal\_casos:nodoOp  
| PARENTESISA operation:nodoOp PARENTESIS  
;