

UNIVERSIDAD DE SAN CARLOS DE GUATEMALA
CENTRO UNIVERSITARIO DE OCCIDENTE
DIVISIÓN DE CIENCIAS DE LA INGENIERÍA
ORGANIZACIÓN DE LENGUAJES Y COMPILADORES 1



MANUAL TÉCNICO

Tecnologías utilizadas:

- Angular: Como lenguaje de programación
- Peggys: Como analizador sintáctico, léxico y semántico
- Visual studio: Como ide utilizado.

Gramáticas utilizadas:

Utilizamos una gramática en la cual podamos representar todo el funcionamiento del programa mini C al ser una gramática bastante compleja no se pudo mostrar todo lo realizado en este documento.

statements

```
= stmts:(statement _)* {  
    return stmts.map([stmt] => stmt);  
}
```

statement

```
= importStmt  
/ varDecl  
/ structDecl  
/ assignment  
/ printStmt  
/ funcDecl  
/ structDefDecl  
/ ifStmt  
/ forStmt  
/ returnStmt  
/ exprStmt  
/ block
```

importStmt

```
= "#import" __ name:identifier _ {  
    return { type: "Import", name };  
}
```

block

```
= "{" _ body:statements _ "}" {  
    return { type: "Block", body };  
}
```

returnStmt

```
= "return" _ value:expression? _ ";" {  
    return { type: "Return", value };  
}
```

```

varDecl
    = type:dataType __ name:identifier __? "=" __? value:expression __?
";" {
    return { type: "VarDecl", varType: type, name, value };
}

dataType
    = "int" / "float" / "string" / "char" / "bool" / structType

structType
    = "struct" __ name:identifier {
        return `struct_${name}`;
    }

structDecl
    = type:structType __ name:identifier __ "=" __ "{" __
values:expressionList _ "}" _ ";" {
    return {
        type: "StructDecl",
        structType: type.substring(7),
        name,
        values
    };
}
/ type:structType __ name:identifier _ ";" {
    return {
        type: "StructDecl",
        structType: type.substring(7),
        name,
        values: null
    };
}

assignment
    = name:identifier "." field:identifier _ "=" _ value:expression _
";" {
    return { type: "Assignment", name, field, value };
}
/ name:identifier _ "=" _ value:expression _ ";" {
    return { type: "Assignment", name, value };
}

printStmt

```

```

= "print" _ "(" _ val:expression _ ")" _ ";" {
    return { type: "Print", value: val };
}

```

funcDecl

```

= ret:("void" / "int" / "float" / "string" / "bool" / "char" /
structType) _ name:identifier _ "(" _ params:parameterList? _ ")" _
"{" _ __? body:statements __? "}" {
    return {
        type: "Function",
        returnType: ret,
        name,
        params: params || [],
        body
    };
}

```

parameterList

```

= head:parameter tail:(_ "," _ parameter)* {
    return [head, ...tail.map(t => t[3])];
}

```

parameter

```

= type:dataType _ name:identifier _ "*" {
    return { type: "Parameter", varType: type, name, byReference:
true };
}
/ type:dataType _ name:identifier {
    return { type: "Parameter", varType: type, name, byReference:
false };
}

```

structDefDecl

```

= "struct" _ name:identifier _ "{" _ __ fields:structField* __ "}" _
";" {
    return { type: "Struct", name, fields };
}

```

structField

```

= type:dataType __ name:identifier _ ";" {
    return { type: "Field", varType: type, name };
}

```

Diagrama de clases:

