

# Gramática abstracta

## CATEGORIES

sentencia, expresion, tipo, definicion

## NODES

**programa** -> definiciones:definicion\*;

**definicionFuncion**:definicion -> nombre:string params:definicionVariable\* tipo  
variablesLocales:definicionVariable\* sentencias:sentencia\*;

**definicionVariable**:definicion -> nombre:string tipo;

**definicionStruct**:definicion -> nombre:string campos:campo\*;

**campo**:definicion -> nombre:string tipo;

**tipoEntero**:tipo -> ;

**tipoReal**:tipo -> ;

**tipoChar**:tipo -> ;

**tipoArray**:tipo -> longitud:int tipo;

**tipoStruct**:tipo -> nombre:string campos:campo\*;

**tipoVoid**:tipo -> ;

**print**:sentencia -> expresion\* tipo\_print:string;

**read**:sentencia -> expresion;

**asignacion**:sentencia -> izquierda:expresion derecha:expresion;

**if**:sentencia -> condicion:expresion verdadero:sentencia\* falso:sentencia\*;

**while**:sentencia -> condicion:expresion sentencia\*;

**invocacion**:sentencia -> nombre:string params:expresion\*;

**return**:sentencia -> expresion\*;

**constanteEntero:**expresion -> valor:int;

**constanteReal:**expresion -> valor:double;

**constanteChar:**expresion -> valor:string;

**variable:**expresion -> nombre:string;

**expresionAritmetica:**expresion -> izq:expresion operador:string der:expresion;

**expresionLogica:**expresion -> izq:expresion operador:string der:expresion;

**comparacion:**expresion -> izq:expresion operador:string der:expresion;

**expresionUnaria:**expresion -> expresion operador:string;

**conversion:**expresion -> nuevoTipo:tipo expresion;

**invocacionExpresion:**expresion -> nombre:string params:expresion\*;

**accesoArray:**expresion -> array:expresion indice:expresion;

**accesoCampo:**expresion -> struct:expresion campo:expresion;