




Attribute Grammar Fase de Tipos

 : Campos por defecto.  : Campos añadidos en esta fase  : Campos añadidos en fase anterior

Nodo	Predicados	Reglas Semánticas
Programa → <i>declaraciones</i> :Declaracion*		
DefVariable :Declaracion → <i>nombre</i> :String <i>tipo</i> :Tipo		
DefStruct :Declaracion → <i>nombre</i> :String <i>listaCampos</i> :DefCampo*		
DefFuncion :Declaracion → <i>nombre</i> :String <i>listaParametros</i> :DefVariable* <i>tipo</i> :Tipo <i>listaDeclaraciones</i> :DefVariable* <i>listaSentencias</i> :Sentencia*	<i>listaParametros</i> . <i>i</i> . <i>tipo</i> ∈ <i>TiposSimples</i> <i>tipo</i> ∈ { <i>TipoEntero</i> , <i>TipoReal</i> , <i>TipCoarácter</i> , <i>TipoVoid</i> }	<i>listaSentencias</i> . <i>i</i> . <i>hisFunction</i> = DefFuncion
SenInvocacion :Sentencia, Expresion → <i>nombre</i> :String <i>listaArgumentos</i> :Expresion*	<i>invocacion</i> . <i>listaParametros</i> == <i>listaArgumentos</i> <i>invocacion</i> . <i>listaParametros</i> . <i>i</i> . <i>tipo</i> is <i>listaArgumentos</i> . <i>i</i> . <i>tipo</i>	
SenAsignacion :Sentencia → <i>left</i> :Expresion <i>right</i> :Expresion	<i>left</i> . <i>tipo</i> is <i>right</i> . <i>tipo</i> <i>left</i> . <i>tipo</i> , <i>right</i> . <i>tipo</i> ∈ <i>TiposSimples</i> <i>left</i> . <i>modificable</i> == TRUE	
SenPrint :Sentencia → <i>expresion</i> :Expresion	<i>expresion</i> . <i>tipo</i> ∈ <i>TiposSimples</i>	
SenPrintSp :Sentencia → <i>expresion</i> :Expresion	<i>expresion</i> . <i>tipo</i> ∈ <i>TiposSimples</i>	
SenPrintLn :Sentencia → <i>expresion</i> :Expresion	<i>expresion</i> . <i>tipo</i> ∈ <i>TiposSimples</i>	
SenRead :Sentencia → <i>expresion</i> :Expresion	<i>expresión</i> . <i>modificable</i> == TRUE <i>expresion</i> . <i>tipo</i> ∈ <i>TiposSimples</i>	
SenIf :Sentencia → <i>condicion</i> :Expresion <i>sentenciasIf</i> :Sentencia* <i>sentenciasElse</i> :Sentencia*	<i>condicion</i> . <i>tipo</i> is TipoEntero	<i>sentenciasIf</i> . <i>i</i> . <i>hisFunction</i> = <i>hisFunction</i> <i>sentenciasElse</i> . <i>i</i> . <i>hisFunction</i> = <i>hisFunction</i>
SenWhile :Sentencia → <i>condicion</i> :Expresion <i>sentencias</i> :Sentencia*	<i>condicion</i> . <i>tipo</i> is TipoEntero	<i>sentencias</i> . <i>i</i> . <i>hisFunction</i> = <i>hisFunction</i>
SenReturn :Sentencia → <i>retorno</i> :Expresion	<i>retorno</i> == NULL ⇒ <i>retorno</i> . <i>tipo</i> is TipoVoid <i>retorno</i> ≠ NULL ⇒ <i>retorno</i> . <i>tipo</i> ∈ <i>TiposSimples</i> <i>retorno</i> . <i>tipo</i> is <i>hisFunction</i> . <i>tipo</i>	
ExAritmetica :Expresion → <i>left</i> :Expresion <i>operador</i> :String <i>right</i> :Expresion	<i>left</i> . <i>tipo</i> ∈ <i>TiposNúmero</i> <i>right</i> . <i>tipo</i> ∈ <i>TiposNúmero</i> <i>right</i> . <i>tipo</i> is <i>left</i> . <i>tipo</i>	<i>tipo</i> = <i>left</i> . <i>tipo</i> <i>modificable</i> = FALSE
ExLogica :Expresion → <i>left</i> :Expresion <i>operador</i> :String <i>right</i> :Expresion	<i>left</i> . <i>tipo</i> is Entero <i>right</i> . <i>tipo</i> is Entero	<i>tipo</i> = TipoEntero <i>modificable</i> = FALSE
ExRelacional :Expresion → <i>left</i> :Expresion <i>operador</i> :String <i>right</i> :Expresion	<i>left</i> . <i>tipo</i> ∈ <i>TiposNúmero</i> <i>right</i> . <i>tipo</i> ∈ <i>TiposNúmero</i> <i>right</i> . <i>tipo</i> is <i>left</i> . <i>tipo</i>	<i>tipo</i> = <i>left</i> . <i>tipo</i> <i>modificable</i> = FALSE
ExNot :Expresion → <i>expresion</i> :Expresion	<i>expresion</i> . <i>tipo</i> is TipoEntero	<i>tipo</i> = TipoEntero <i>modificable</i> = FALSE
ExInvocacion :Expresion → <i>nombre</i> :String <i>listaArgumentos</i> :Expresion*	<i>invocacion</i> . <i>listaParametros</i> == <i>listaArgumentos</i> <i>invocacion</i> . <i>listaParametros</i> . <i>i</i> . <i>tipo</i> == <i>listaArgumentos</i> . <i>i</i> . <i>tipo</i>	<i>modificable</i> = FALSE <i>tipo</i> = <i>hisFunction</i> . <i>tipo</i>

	<i>tipo</i> ∈ <i>TiposSimples</i>	
ExIndice :Expresion → <i>left</i> :Expresion <i>indice</i> :Expresion	<i>left.tipo</i> is TipoArray <i>indice.tipo</i> is TipoEntero	<i>tipo</i> = <i>left.tipo.tipobase</i> <i>modificable</i> = <i>left.modificable</i>
ExCampo :Expresion → <i>struct</i> :Expresion <i>campo</i> :Expresion	<i>struct.modificable</i> == TRUE <i>campo.modificable</i> == TRUE <i>struct.tipo</i> is TipoStruct <i>campo</i> ⊂ <i>struct.tipo.definicion.campos</i>	<i>campo.declaracion</i> = <i>struct.tipo.definicion.listaCampos</i> [<i>nombre</i> = <i>campo.nombre</i>] <i>tipo</i> = <i>campo.Tipo</i> <i>modificable</i> = <i>campo.modificable</i>
ExCast :Expresion → <i>to</i> :Tipo <i>from</i> :Expresion	<i>to.tipo</i> , <i>from.tipo</i> ∈ <i>TiposSimples</i> <i>to.tipo</i> is <i>from.tipo</i>	<i>tipo</i> = <i>to.tipo</i> <i>modificable</i> = FALSE
LitEntero :Expresion, Literal → <i>valor</i> :String		<i>tipo</i> = TipoEntero <i>modificable</i> = FALSE
LitReal :Expresion, Literal → <i>valor</i> :String		<i>tipo</i> = TipoREAL <i>modificable</i> = FALSE
LitCaracter :Expresion, Literal → <i>valor</i> :String		<i>tipo</i> = TipoCaracter <i>modificable</i> = FALSE
Variable :Expresion → <i>nombre</i> :String		<i>tipo</i> = <i>declaracion.tipo</i> <i>modificable</i> = TRUE
TipoEntero :Tipo → λ		
TipoReal :Tipo → λ		
TipoCaracter :Tipo → λ		
TipoVariable :Tipo → <i>nombre</i> :String		
TipoArray :Tipo → <i>tipo</i> :Tipo <i>tamaño</i> :String	<i>tamaño</i> > 0	
TipoVoid :Tipo → λ		

Recordatorio de operadores (para cortar y pegar): ⇒ ⇔ ≠ ∅ ∈ ∉ ∪ ∩ ⊂ ⊄ ∑ ∃ ∀

TiposSimples = {*TipoEntero*, *TipoReal*, *TipoCaracter*}

TiposNúmero = {*TipoEntero*, *TipoReal*}

Atributos (De esta Fase y los de la fase de identificación)

Categoría Sintáctica	Nombre del atributo	Tipo Java	Heredado/Sintetizado	Descripción
Sentencia	<i>hisFunction</i>	DefFuncion	Heredado	Función en la que se encuentra la Sentencia.
Expresion	<i>tipo</i>	Tipo	Sintetizado	Tipo absoluto de una expresión.
Expresion	<i>modificable</i>	Boolean	Sintetizado	Modificabilidad de una expresion.
Categoría Sintáctica	Nombre del atributo	Tipo Java	Heredado/Sintetizado	Descripción
SenInvocacion, ExInvocacion	<i>invocacion</i>	DefFuncion	Sintetizado	Función a invocar por la invocación.
TipoStruct	<i>definicion</i>	DefStruct	Sintetizado	DefStruct de una DefVariable de TipoStruct
Variable	<i>declaracion</i>	DefVariable	Sintetizado	DefVariable de una Variable

