

## COMPILADORES E INTÉRPRETES

Proyecto  
Aspectos Léxicos de MINIJAVA  
Segundo Cuatrimestre de 2025

### 1 Introducción

Este documento describe parte de la sintaxis de MINIJAVA. Como se menciona en el documento de especificación del proyecto, MINIJAVA es un lenguaje que puede verse como una simplificación de Java, donde por una parte no se consideran elementos avanzados como Genericidad, Excepciones o Hilos, y por otra parte la estructura de su sintaxis es más estricta. En este documento, se presentan los elementos léxicos correspondientes a la sintaxis del lenguaje, donde en particular se identificarán las palabras reservadas, forma de los identificadores y literales del lenguaje.

### 2 Componentes Léxicos

El primer paso para compilar un programa de MINIJAVA radica en convertir una entrada de caracteres en una entrada de tokens, donde cada token se corresponderá con un lexema de MINIJAVA. Los lexemas de MINIJAVA son palabras clave, nombres de tipos primitivos, literales, símbolos de puntuación, operadores e identificadores. Cada uno de estos elementos se describe en las siguientes secciones.

#### 2.1 Palabras Clave

MINIJAVA tiene las siguientes palabras clave:

<b>class</b>	<b>extends</b>	<b>public</b>	<b>static</b>	<b>void</b>
<b>boolean</b>	<b>char</b>	<b>int</b>	<b>abstract</b>	<b>final</b>
<b>if</b>	<b>else</b>	<b>while</b>	<b>return</b>	<b>var</b>
<b>this</b>	<b>new</b>	<b>null</b>	<b>true</b>	<b>false</b>

Note que varias de las palabras clave originales de Java no están presentes en MINIJAVA. Un compilador de MINIJAVA podría considerar esas palabras clave no presentes como prohibidas en MINIJAVA.

#### 2.2 Identificadores

En MINIJAVA hay dos tipos de identificadores: los identificadores de clases y los identificadores de métodos y variables. Un identificador de clase es una letra mayúscula seguida de cero o más letras (mayúsculas o minúsculas), dígitos y underscores. Un identificador de método y variable es una letra minúscula seguida de cero o más letras (mayúsculas o minúsculas), dígitos y underscores.

En los documentos con la gramática<sup>1</sup> de MINIJAVA los identificadores de clase son denotados como **idClase**, mientras que los identificadores de métodos y variables son denotados como **IdMetVar**. Ningún identificador puede ser una palabra reservada.

---

<sup>1</sup>El documento con la gramática y las reglas de sintaxis de MINIJAVA estará disponible mas adelante cuando deban realizar la Etapa 2

## 2.3 Literales

En MINIJAVA hay cinco tipos de literales, los cuales se corresponden a los tipos primitivos del lenguaje.

### 2.3.1 Literales Enteros

Un literal entero es una secuencia de uno o mas dígitos. El literal entero no puede contener mas de 9 dígitos y su valor corresponde a su interpretación estándar en base 10. En el documento con la gramática de MINIJAVA los literales enteros son denotados como **intLiteral**.

### 2.3.2 Literales Caracteres

Un literal valido caracter tener la forma:

- ‘ $x$ ’ donde  $x$  es cualquier caracter excepto la barra invertida (\) o la comilla simple (’).
- ‘\( $x$ ’ donde  $x$  es cualquier caracter.

En el documento con la gramática de MINIJAVA los literales caracter son denotados como **charLiteral**.

### 2.3.3 Literales String

Un literal string se representa mediante una comilla doble (") seguida de una secuencia de caracteres y finaliza con otra comilla doble ("). Al igual que en Java la secuencia de caracteres no puede ser interrumpida por un salto de linea. También como en Java, es posible que en la secuencia de caracteres aparezca una comilla doble siempre y cuando se la anteceda por el caracter \. Por esto último la comilla doble que marca el final de un string correcto nunca estará antecedita por ese caracter. El valor del literal corresponde a la cadena de caracteres entre las comillas.

En el documento con la gramática de MINIJAVA los literales string son denotados como **stringLiteral**.

### 2.3.4 Literales Booleanos

Los literales booleanos se representan mediante las palabras reservadas **true** y **false**.

### 2.3.5 Literal Nulo

El literal nulo se representa mediante la palabra reservada **null**.

## 2.4 Símbolos

Al igual que en Java, en MINIJAVA cada símbolo valido es identificado univocamente con un nombre de token que lo diferencia del resto de los símbolos. En MINIJAVA identificamos los siguientes:

### 2.4.1 Puntuación

MINIJAVA cuenta con los siguientes símbolos de puntuación, utilizados para estructurar los programas:  
**() {} ; , . :**

### 2.4.2 Operadores

El lenguaje MINIJAVA cuenta con los siguientes operadores que serán utilizados en sus expresiones:

>	<	!	=
==	>=	<=	!=
&&		%	
+	-	*	/
++	--		

## 3 Otros Componentes

En esta seccion se marcan los elementos que no son identificados como tokens del lenguaje. Se dividen los que pueden aparece en un programa valido y los que no

### 3.1 Espacios en Blanco

Los espacios en blanco en MINIJAVA, al igual que en Java, simplemente hacen más fácil la lectura del programa por un humano y **no** son tokens. Un espacio en blanco es un espacio, tab o un salto de línea.

### 3.2 Comentarios

Los comentarios son ignorados por el analizador léxico, anlogamente a los espacios en blanco, simplemente hacen más fácil la lectura del programa por un humano y **no** son tokens. MINIJAVA, al igual que Java, tiene dos estilos de comentarios:

- `/* comentario */` Comentarios multi-línea: Todos los caracteres desde `/*` hasta `*/` son ignorados.
- `// comentario` Comentario simple: Todos los caracteres de desde `//` hasta el final de la línea son ignorados.

### 3.3 Otros caracteres

Cualquier otra entrada que no conforme con las reglas anteriormente presentadas debe generar un error.