

<letra>	<b>a ... z A ... Z</b>
<dígito>	<b>0 1 ... 9</b>
<ascii>	<letra> <dígito> * <b>¿ ^ ...</b>
<ident>	<letra>
<resto ident>	<letra> <dígito> <b>lambda</b>
<cadena>	<ascii> <b>lambda</b>
<cte-string>	<b>"</b>
<cte-char>	<b>'</b>
<resto número>	<dígito> <b>lambda</b>
<número>	<dígito>
<resto cte opcional>	<b>. lambda</b>
<cte-num>	<dígito>
<constante>	<dígito> <b>'</b>
<resto lista expr>	<b>, lambda</b>
<invocación función>	<letra>
<resto opcional>	<b>[ lambda</b>
<variable>	<letra>
<factor>	<letra> <dígito> <b>'</b> <b>!</b> <b>(</b> <b>"</b>
<opermul>	<b>* / &amp;&amp;</b>
<resto término>	<b>* / &amp;&amp; lambda</b>
<término>	<letra> <dígito> <b>'</b> <b>!</b> <b>(</b> <b>"</b>
<operador>	<b>+ -   </b>
<resto expresión simple>	<b>+ -    lambda</b>
<operador opcional>	<b>+ - lambda</b>
<expresión simple>	<b>+</b> <b>-</b> <letra> <dígito> <b>'</b> <b>!</b> <b>(</b> <b>"</b>

<op relacional>	!= == < <= >= > =
<resto expresión>	!= == < <= >= > = <b>lambda</b>
<expresión>	+ - <letra> <dígito> , ! ( "
<resto prop out>	<< <b>lambda</b>
<lista expresiones>	+ - <letra> <dígito> , ! ( " <b>lambda</b>
<resto prop in>	>> <b>lambda</b>
<proposición ent/sal>	<b>cin cout</b>
<proposición expresión>	+ - <letra> <dígito> , ! ( " ;
<proposición retorno>	<b>return</b>
<proposición iteración>	<b>while</b>
<else opcional>	<b>else lambda</b>
<proposición selección>	<b>if</b>
<proposición>	<b>if while</b> + - <letra> <dígito> , !



<especificador de tipo>	<b>void char int float</b>
<declaraciones>	<b>void char int float</b>
<unidad traducción>	<b>void char int float lambda</b>