```
<expresion> -> <expresion de asignacion> |
              <expresion> , <expresion de asignacion>
<expresion unaria> <operador asignacion> <expresion de asignacion>
<expresion condicional> ->
                             <expresion O logico> |
                              <expresion O logico> ? <expresion> : <expresion condicional>
<operador asignacion> -> uno de = *= /= %= += -= <<= >>= &= ^= |=
<expresion O logico> -> <expresion Y logico> |
                      <expresion O logico> | | <expresion Y logico>
<expresion Y logico> -> <expresion O inclusivo> |
                      <expresion Y logico> && <expresion O inclusivo>
                             <expresion O excluyente> |
<expresion O inclusivo> ->
                              <expresion O inclusivo> |
                             <expresion O excluyente>
<expresion O excluyente> ->
                             <expresion Y> |
                             <expresion O excluyente> ^ <expresion Y>
<expresion Y> -> <expresion de igualdad> |
                <expresion Y> & <expresion de igualdad>
<expresion de igualdad> -> <expresion relacional> |
                          <expresion de igualdad> == <expresion relacional> |
                          <expresion de igualdad> != <expresion relacional>
<expresion relacional> -> <expresion de corrimiento> |
                        <expresion relacional> < <expresion de corrimiento> |
                         <expresion relacional> > <expresion de corrimiento> |
                         <expresion relacional> <= <expresion de corrimiento> |
                         <expresion relacional> >= <expresion de corrimiento>
<expresion de corrimiento> -> <expresion aditiva> |
                             <expresion de corrimiento> << <expresion aditiva> |
                              <expresion de corrimiento> >> <expresion aditiva>
<expresion aditiva> -> <expresion multiplicativa> |
                      <expresion aditiva> + <expresion multiplicativa> |
                      <expresion aditiva> - <expresion multiplicativa>
```

```
<expresion multiplicativa> * <expresion de conversion> |
                             <expresion multiplicativa> / <expresion de conversion> |
                             <expresion multiplicativa> % <expresion de conversion>
<expresion de conversion> -> <expresion unaria> |
                            (<nombre de tipo>) <expresion de conversion>
<expresion unaria> -> <expresion sufijo> |
                     ++ <expresion unaria> |
                     -- <expresion unaria> |
                     <operador unario> <expresion de conversion> |
                     sizeof <expresion unaria> |
                     sizeof (<nombre de tipo>)
<nombre de tipo> está descripto más adelante, en la seccion Declaraciones.
<operador unario> -> uno de & * + - ~!
<expresion sufijo> -> <expresion primaria> |
                     <expresion sufijo> [<expresion>] |
                      /* arreglo */ <expresion sufijo> (ta de argumentos>?) |
                      /* invocacion */ <expresion sufijo> . <identificador> |
                      <expresion sufijo> -> <identificador> |
                      <expresion sufijo> ++ |
                     <expresion sufijo> --
de argumentos> ->
                            <expresion de asignacion> |
                            ta de argumentos> , <expresion de asignacion>
<expresion primaria> -> <identificador> |
                     <constante> |
                     <constante cadena> |
                     (<expresion>)
```