

## Tarea 1

### (solución)

Lenguajes de Programación

Tomada de [Sethi,1992]

**Nota:** La solución puede tener muchas respuestas. La que les doy a continuación es una de ellas.

1. Proporcione las gramáticas en BNF, BNFE y Diagrama (o Esquema) de Sintaxis para describir la sintaxis de cada uno de los siguientes casos:

- a. Cadenas de longitud 1 o mayores definidas sobre el conjunto de símbolos terminales {**blanco**, **tab**, **EOL**}.

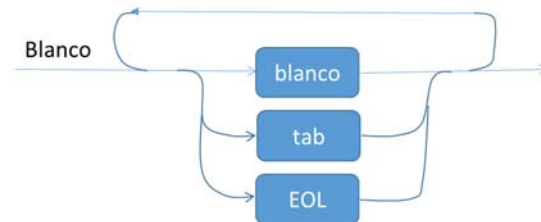
BNF

```
<blancos> ::= blanco <blancos_1>
            | tab <blancos_1>
            | EOL <blancos_1>
<blancos_1> ::= <ε>
            | <blancos>
```

BNFE

```
Blancos ::= (blanco|tab|EOL) {blanco|tab|EOL}
```

DS



- b. Secuencia de letras o dígitos, iniciando con una letra (especie de identificadores).

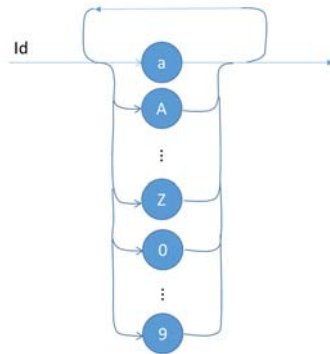
BNF

```
<id> ::= <letra> <id1>
<id2> ::= <ε>
            | <letra> <id2>
            | <digito> <id2>
<letra> ::= a|A|b|B|...|z|Z
<digito> ::= 0|1|2|3|4|5|6|7|8|9
```

BNFE

```
Id ::= (a|A|b|B|...|z|Z) { a|A|b|B|...|z|Z|0|...|9 }
```

DS



- c. Números reales en los cuales, tanto la parte entera como la fraccionaria, puedan estar vacías, pero no ambas a la vez. Así, la gramática debe permitir 31., 3.1, y .13, pero no un punto decimal solo.

BNF

```

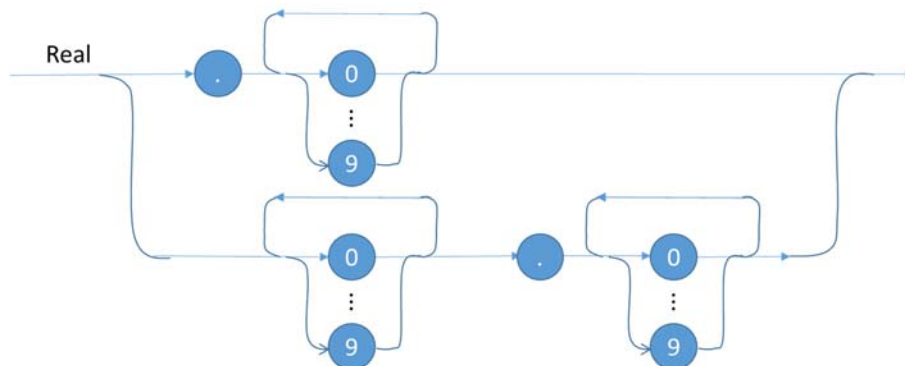
<real> ::= . <dígitos>
        | <dígitos> .
        | <dígitos> . <dígitos>
<dígitos> ::= <numero> <digitos1>
<digitos1> ::= <ε>
              | <numero> <digitos1>
<numero> ::= 1 | 2 | ... | 9
  
```

BNFE

```

Real ::= . (0 | 1 | ... | 9) { 0 | 1 | ... | 9 }
        | (0 | 1 | ... | 9) { 0 | 1 | ... | 9 } . { 0 | 1 | ... | 9 }
  
```

DS



2. La siguiente gramática BNFE se basa en la sintaxis de los enunciados del lenguaje de programación Modula-2:

```

S ::= []
    | id := expr
    | if expr then SL { elsif expr then SL } [ else SL ] end
    | loop SL end
    | while expr do SL end
SL ::= S { ; S }

```

El componente léxico id representa una variable, mientras que expr representa una expresión. Obsérvese que [] representa la cadena vacía.

a. Escriba una versión BNF de esta gramática

```

<S> ::= <ε>
        | id := expr
        | if expr then <SL> <elsifST> <elseST> end
        | loop <SL> end
        | while expr do <SL> end
<SL> ::= <S> <S1>
<S1> ::= <ε>
        | <;S>
<elsifST> ::= <elsifST1> <elsifST>
<elseifST1> ::= <ε>
                | else expr then <SL>
<elseST> ::= <ε>
            | else <SL>

```

b. Escriba un esquema de sintaxis para esta gramática.

