

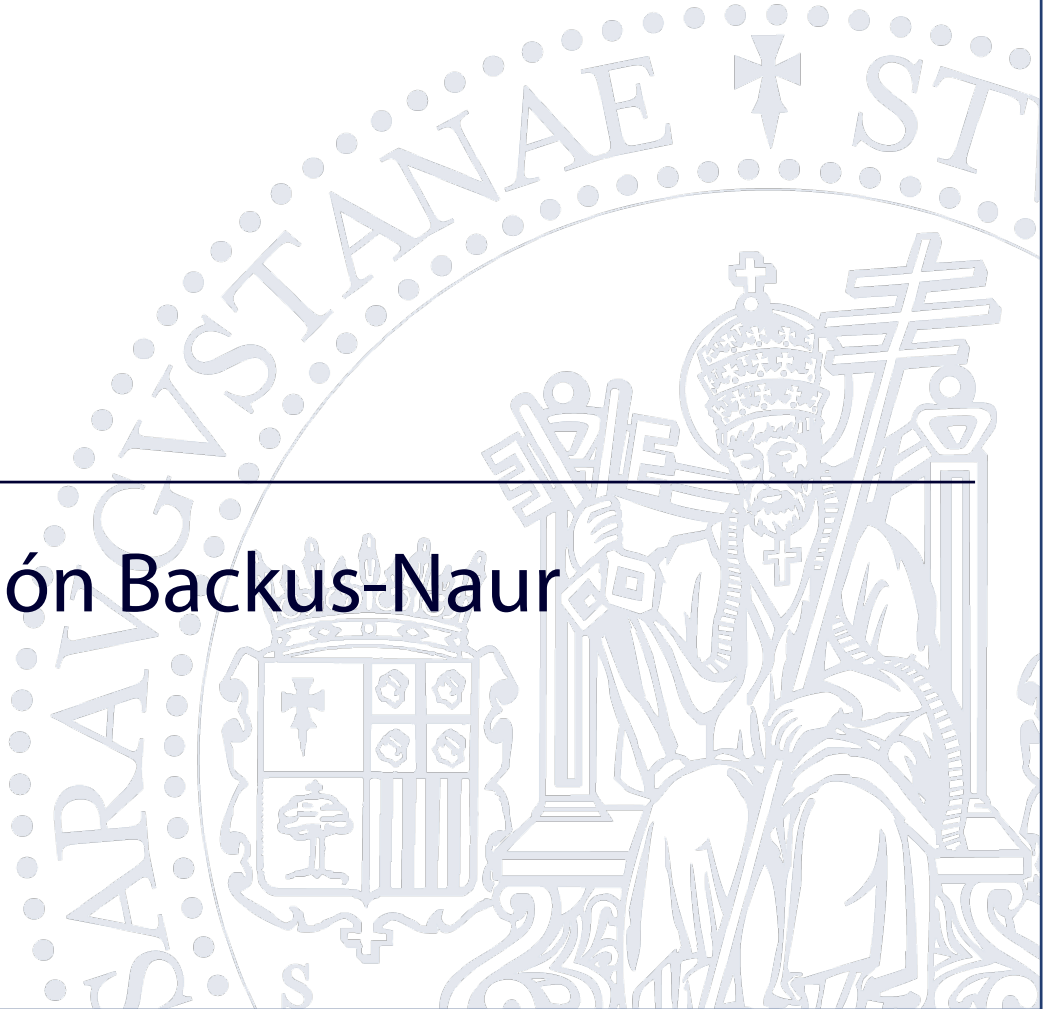
Programación 1

Problemas 1

Notación Backus-Naur



Escuela de
Ingeniería y Arquitectura
Universidad Zaragoza





Notación de Backus-Naur

- Notación BNF (*Backus-Naur form*)
 - Definición de reglas sintácticas para definir lenguajes
 - Descripción de la organización de estructuras de datos secuenciales

Notación de Backus-Naur

- **Metasímbolos** utilizados:
 - Definición de una regla
<nombre_regla> ::= expresión
 - Sustitución de la expresión
<nombre_regla>
 - Literal
"Prog1f"
 - Alternativa
expresión1 | expresión2
 - Agrupación sin repetición
(expresión)
 - Agrupación con opcionalidad (cero o una veces)
[expresión]
 - Agrupación con repetición (cero, una o más veces)
{ expresión }

Notación Backus-Naur

$::=$	Definición de regla sintáctica
$\langle \quad \rangle$	Delimitadores de nombre de regla sintáctica
$\text{“} \quad \text{”}$	Delimitadores de carácter o secuencia de caracteres literal
$ $	Separador de alternativas
(\quad)	Agrupador sin repetición
$\{ \quad \}$	Agrupador con repetición (0, 1 o más veces)
$[\quad]$	Agrupador opcional (0 o 1 vez)

Ejemplos «lingüísticos»

```
<oración> ::= [<sujeito>] <predicado>
<sujeito> ::= <sintagma_nominal>
<predicado> ::=
    (<verbo_copulativo> <atributo>)
| (<verbo_predicativo>
    [<complemento_directo>]
    [<complemento_indirecto>]
    {<complemento_circunstancial>})
```

Ejemplos

<letraIng>	::=	“A”		“B”		“C”		“D”
		“E”		“F”		“G”		“H”
		“I”		“J”		“K”		“L”
		“M”		“N”		“O”		“P”
		“Q”		“R”		“S”		“T”
		“U”		“V”		“W”		“X”
		“Y”		“Z”		“a”		“b”
		“c”		“d”		“e”		“f”
		“g”		“h”		“i”		“j”
		“k”		“l”		“m”		“n”
		“o”		“p”		“q”		“r”
		“s”		“t”		“u”		“v”
		“w”		“x”		“y”		“z”

Ejemplos

$\langle \text{letraEsp} \rangle ::=$

$\langle \text{letraIng} \rangle \mid \text{“Ñ”} \mid \text{“Á”} \mid \text{“É”}$
 $\mid \text{“Í”} \mid \text{“Ó”} \mid \text{“Ú”} \mid \text{“Ü”} \mid \text{“ñ”}$
 $\mid \text{“á”} \mid \text{“é”} \mid \text{“í”} \mid \text{“ó”} \mid \text{“ú”}$
 $\mid \text{“ü”}$

$\langle \text{palabra} \rangle ::=$

$\langle \text{letraEsp} \rangle \{ \langle \text{letraEsp} \rangle \}$

Problema 1

<vocal>	::=	“a”		“e”		“i”		“o”			
			“u”		“á”		“é”		“í”		“ó”
			“ú”		“ü”						

<consonante>	::=	“b”		“c”		“d”					
			“f”		“g”		“h”		“j”		“k”
			“l”		“m”		“n”		“ñ”		“p”
			“q”		“r”		“s”		“t”		“v”
			“w”		“x”		“y”		“z”		



Problema 1

- Palabras que empiezan por consonante y tienen vocales y consonantes alternadas
 - g, de, pan, caso, hogar, coraza, mirador, zaragozano, ...

Problema 2

<letra_may> ::=							
	“A”	“B”	“C”	“D”			
	“E”	“F”	“G”	“H”	“I”	“J”	
	“K”	“L”	“M”	“N”	“O”	“P”	
	“Q”	“R”	“S”	“T”	“U”	“V”	
	“W”	“X”	“Y”	“Z”	“Ñ”	“Á”	
	“É”	“Í”	“Ó”	“Ú”	“Ü”		
<letra_min> ::=							
	“a”	“b”	“c”	“d”			
	“e”	“f”	“g”	“h”	“i”	“j”	
	“k”	“l”	“m”	“n”	“o”	“p”	
	“q”	“r”	“s”	“t”	“u”	“v”	
	“w”	“x”	“y”	“z”	“ñ”	“á”	
	“é”	“í”	“ó”	“ú”	“ü”		



Problema 2

- Palabras con secuencia correcta de mayúsculas y minúsculas

Problema 2. Solución

```
<palabra> ::=  
    (<letra_min> {<letra_min>})  
| (<letra_may> {<letra_min>})  
| (<letra_may> {<letra_may>})
```



Problema 3. Sintaxis en notación NBF para un número entero positivo

```
<dígito> ::= "0" | "1" | "2" | "3"  
           | "4" | "5" | "6" | "7" | "8"  
           | "9"
```

```
<positivo>  
    ::= <dígito> {<dígito>}
```

Problema 3. Sintaxis en notación NBF para un número entero positivo

```
<dígitoNoNulo> ::= "1" | "2" | "3"  
                  | "4" | "5" | "6" | "7" | "8"  
                  | "9"  
  
<dígito> ::= "0" | <dígitoNoNulo>  
  
<positivo> ::= "0"  
              | (<dígitoNoNulo> {<dígito>})
```

Problema 4. Secuencias binarias

```
<bit> ::= "0" | "1"  
<secuencia_binaria>  
    ::= <bit> {<bit>}
```

Problema 5. Identificadores

```
<identificador> ::=  
    ( <letra> | “_” ) { <letra> | <dígito> | “_” }  
<letra> ::= <mayúscula> | <minúscula>  
<mayúscula> ::= “A” | “B” | “C” | “D” | “E” | “F” | “G” | “H”  
    | “I” | “J” | “K” | “L” | “M” | “N” | “O” | “P” | “Q” | “R” | “S”  
    | “T” | “U” | “V” | “W” | “X” | “Y” | “Z”  
<minúscula> ::= “a” | “b” | “c” | “d” | “e” | “f” | “g” | “h”  
    | “i” | “j” | “k” | “l” | “m” | “n” | “o” | “p” | “q” | “r” | “s”  
    | “t” | “u” | “v” | “w” | “x” | “y” | “z”  
<dígito> ::= “0” | “1” | “2” | “3” | “4” | “5” | “6” | “7”  
    | “8” | “9”
```


Problema 5. Identificadores

```
<identificador> ::=  
    ( <letra> | _ ) { <letra> | <dígito> | _ }  
<letra> ::= <mayúscula> | <minúscula>  
<mayúscula> ::= A | B | C | D | E | F | G | H  
    | I | J | K | L | M | N | O | P | Q | R | S  
    | T | U | V | W | X | Y | Z  
<minúscula> ::= a | b | c | d | e | f | g | h  
    | i | j | k | l | m | n | o | p | q | r | s  
    | t | u | v | w | x | y | z  
<dígito> ::= 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7  
    | 8 | 9
```

Problema 5. Identificadores (¿válidos, no válidos, aconsejables?)

C++

java

UM0164G

spider-man

pRINCIPIO

error!

77E2

String

begin

o123

mannana

mañana

interés

cœur

tasa de cambio

p'adelante

hinundaciones

cero07

Problema 5. Identificadores

(¿válidos, no válidos, aconsejables?)

```
dato_leido  
primer_dato_escrito_en_el_fichero  
primerDato  
dinero$  
_aux  
__aux  
_Aux  
tabla_temperaturas_____auxiliar  
X_  
o_o
```



Problema 6. Estructuras de datos

Ficheros con formato SubRip (.srt)

...

104

00:08:02,997 --> 00:08:05,563

Sí, bueno, creía que la Mano del Rey...

105

00:08:05,663 --> 00:08:07,541

...era bienvenido en las
reuniones del Pequeño Consejo.

106

00:08:07,575 --> 00:08:09,577

Nuestro padre es la Mano del Rey.

107

00:08:09,611 --> 00:08:12,647

Sí, pero en su ausencia...

...

Problema 6. Estructuras de datos

Ficheros con formato SubRip (.srt)

```
<ficheroSubRip> ::= {<subtítulo>}  
<subtítulo> ::= <número> finLínea  
                <tiempo> “ --> ” <tiempo> finLínea  
                {<texto> finLínea}  
                finLínea  
<número> ::= <dígito> {<dígito>}  
<tiempo> ::= <dígito> <dígito> “:”  
              <dígito> <dígito> “:” <dígito> <dígito> “,”  
              <dígito> <dígito> <dígito>  
<texto> ::= <carácter> {<carácter>}  
<carácter> ::= carácter distinto a finLínea
```