Problemas de Programación 1
Tema 2. Notación BNF

Palabras

Dada las siguientes reglas expresadas en notación Backus-Naur:

```
<letra> ::= <vocal> | <consonante>
<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

Escribe una regla denominada **<palabra>** que indique cómo construir palabras simplemente como una secuencia de una o más letras.

```
<palabra> ::=
```

Problema 2

Utilizando las reglas **<vocal>** y **<consonante>** definidas anteriormente, escribe una regla denominada **<palabra-alterna>** que permita construir palabras que **empiezan por consonante** y tienen **vocales y consonantes alternadas** como, por ejemplo, g, de, pan, caso, hogar, coraza, mirador, zaragozano, …

```
<palabra-alterna> ::=
```

Números

Problema 3

Escribe las reglas necesarias para expresar cómo construir un número natural (un número entero mayor o igual que 0). Puedes necesitar definir más de una regla.

```
<natural> ::=
```

Escribe ejemplos de naturales que cumplan con la sintaxis de la regla **<natural>** que has definido y escribe también ejemplos de otros elementos que no cumplan con dicha sintaxis.

Problema 4

Añade una regla a la sintaxis del problema anterior para que indique cómo construir números enteros:

```
<entero> ::=
```

Identificadores

Problema 5

Aunque no es exactamente la definición sintáctica que da el estándar de C++, este es el conjunto de reglas que utilizaremos para definir identificadores en C++ en este curso:

```
<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
```

Para aumentar la legibilidad, se han omitido las comillas al definir los símbolos terminales

Según las reglas anteriores, ¿cuáles de las siguientes secuencias de caracteres son identificadores válidos (cumplen con la sintaxis de la regla <identificador>) y cuáles no? Y de válidos, ¿cuáles crees que son aconsejables y cuáles no?

	¿Válido?	Si válido, ¿aconsejable?
C++		
Java		
UMO164G		
spider-man		
pRINCIPIO		
error!		
77E2		
String		
Begin		
o123		
mannana		
mañana		
interés		
cœur		
tasa de cambio		
p'adelante		
hinundaciones		
cero07		
dato_leido		
primer_dato_escrito_en_el_fichero		
primerDato		
dinero$		
_aux		
__aux		
_Aux		
tabla_temperaturas_____auxiliar		
X_		
o_o		

Ficheros con formato SubRip (.srt)

Problema 6

El formato de ficheros SubRip[1] es un formato de ficheros de texto que permite representar el contenido y los tiempos de sincronización de los subtítulos de un vídeo. Un fichero que sigue este formato contiene una secuencia de subtítulos, constando cada uno de ellos de varias líneas en las que se aparecen los siguientes elementos:

- Un contador numérico que identifica cada subtítulo
- Las marcas temporales a las que el subtítulo tiene que aparecer y desaparecer, separada por la secuencia « `-->` »
- El subtítulo propiamente dicho, en una o más líneas
- Una línea en blanco, que separa el subtítulo del que le sigue

A modo de ejemplo, se muestra el contenido de parte de un fichero SubRip:

```
...

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...

...
```

Escribe las reglas sintácticas que permitan expresar el contenido de un fichero SubRip. Utiliza una regla denominada **<fin-línea>** que supondremos **ya definida** para referirnos al carácter que finaliza una línea y una regla **<carácter>**, también **ya definida**, que hace referencia a cualquier carácter distinto al de final de línea

```
<fichero-subrip> ::=
...
```

[1] Wikipedia contributors. (2021, September 20). «SubRip». *Wikipedia, The Free Encyclopedia*. Accedido el 6-9-2023. https://en.wikipedia.org/wiki/SubRip#SubRip_file_format