

PROCESADORES DE LENGUAJES

TRABAJO PRÁCTICO - PRIMER ENTREGABLE - GRUPO B

- Rubén Márquez Villalta
- Alberto Velasco Mata
- Diego Pedregal Hidalgo
- Antonio Manjavacas Lucas

1. Presentación del problema

Una empresa Castellano Manchega, dirigida por un joven emprendedor de la región apodado el Dr. Doofenshmirtz manchego, ha ideado un robot de cocina que facilita la labor de la elaboración de menús a las familias "ocupadas" del siglo XXI. Su máquina la Gachaneitor 1.0 tiene un funcionamiento similar a otra máquina que ya existe en el mercado (la Thermomix) con la novedad de permitir su programación por cualquier persona. Para su construcción ha contado con la inestimable colaboración de un grupo de brillantes alumnos de la asignatura de "Procesadores de Lenguajes" de la Escuela Superior de Informática.

Estos se encargarán de diseñar un lenguaje que permita definir una o varias recetas y de construir su procesador asociado que como salida deberá generar "algo" que permita simular su funcionamiento (Más Complejo - Más Valorado) o la visualización de las mismas en un lenguaje de marcado (Menos Complejo - Menos Valorado).

El lenguaje deberá tener una zona en la que se definan los ingredientes y cantidades a usar a nivel de menú compuesto por varios platos y una zona de definición de las instrucciones de elaboración de cada plato que forma el menú, que obviamente deberá usar los ingredientes definidos. También será posible definir los ingredientes en la zona de elaboración de cada plato, en este caso solo podrán usarse en ese plato.

Además, cada receta deberá contar con información sobre el autor, el tipo de plato, el tiempo de preparación, cocinado y total, el número de comensales y por supuesto el nombre de cada plato.

2. Representación EBNF

A continuación se detalla la especificación *EBNF* de la gramática utilizada para definir el lenguaje *Gachacode*. Para más detalle consúltese el **Anexo I**.

```
<MENU> ::= "menu" <ID> '{' <NOMBRE> {<INGREDIENTE> ','} <PLATO> {<PLATO>} '}'
<PLATO> ::= "plato" <ID> '{' <CABECERA> {<INGREDIENTE> ','} {<INSTRUCCION>} '}'

<CABECERA> ::= "info" '{' <NOMBRE> <USUARIO> <RACIONES> <TIEMPO_TOTAL> [<CALORIAS>] [<TIPO>] [<COCINA>] '}'
<NOMBRE> ::= "nombre" ':' <CADENA> ';'
<USUARIO> ::= "usuario" ':' <ID> ';'
<RACIONES> ::= "raciones" ':' <NUMERO> ';'
<TIEMPO_TOTAL> ::= "tiempo" ':' <DURACION> ';'
<CALORIAS> ::= "calorias" ':' <NUMERO> ';'
<TIPO> ::= "tipo" ':' <STRING> ';'
<COCINA> ::= "cocina" ':' <STRING> ';'

<INSTRUCCION> ::= (<PROGRAMAR> | <ANADIR> | <CALENTAR> | <REMOVER> | <ACCION_USUARIO> | <SACAR>) ';'
<PROGRAMAR> ::= "programar" '(' <TEMPERATURA> ',' <VELOCIDAD> ',' <TIMER> [',' "inverso"] ')'
<CALENTAR> ::= "calentar" '(' <TEMPERATURA> ',' <TIMER> ')'
<REMOVER> ::= "remover" '(' <VELOCIDAD> ',' <TIMER> [',' "inverso"] ')'
<ACCION_USUARIO> ::= <STRING>
<ANADIR> ::= "anadir" '(' (<INGREDIENTE> | <STRING>) ')'
<SACAR> ::= "sacar" '(' <STRING> ')'
<TEMPERATURA> ::= (<TEMP> | "varoma")
<VELOCIDAD> ::= (<NUMERO> | "cuchara" | "espiga" | "turbo")
<INGREDIENTE> ::= <CANTIDAD> <STRING>
<CANTIDAD> ::= (<NUMERO> ('l' | 'ml' | 'g' | 'cucharada' | 'ud') | "al_gusto")

<ID> ::= ('A' | ... | 'Z' | 'a' | ... | 'z') {('A' | ... | 'Z' | 'a' | ... | 'z' | '0' | ... | '9')}
<STRING> ::= '"' {('A' | ... | 'Z' | 'a' | ... | 'z' | '0' | ... | '9' | <BLANCO>)} '"'

<DURACION> ::= [<NUMERO> 'h'] <NUMERO> 'm'
<TIMER> ::= <DIGITO> <DIGITO> ':' <DIGITO> <DIGITO>
<TEMP> ::= <NUMERO> [0] 'C'
<DIGITO> ::= ('0' | ... | '9')
<NUMERO> ::= <DIGITO> {<DIGITO>}
<BLANCO> ::= (' ' | '\t' | '\n' | '\r')
```

Un ejemplo de programa escrito en el lenguaje *Gachacode* sería el siguiente:

```
menu merendola {
  nombre: "Merienda estudiantil";

  100g pan;

  plato bocadilloChopped {
    info {
      nombre: "Bocadillo de Chopped";
      usuario: NoamChomsky;
      raciones: 1;
      tiempo: 5m;
      calorias: 44;
      tipo: "Merienda";
      cocina: "Española";
    }

    30g chopped;

    anadir(chopped);
    programar(37°C, espiga, 00:05, inverso);
    sacar("chopped caliente");
    "introducir chopped caliente en pan cortado";
  }
}
```

3. Análisis léxico

Tokens básicos identificados:

TABLA DE TOKENS			
ID	Token	Expresión regular	Lexema
T1	<i>PAREN_IZQ</i>	((
T2	<i>PAREN_DER</i>))
T3	<i>LLAVE_IZQ</i>	{	{
T4	<i>LLAVE_DER</i>	}	}
T5	<i>COMA</i>	,	,
T6	<i>PUNTO_COMA</i>	;	;
T7	<i>DOS_PUNTOS</i>	:	:
T8	<i>DURACION</i>	([0-9]+·h)?·[0-9]+·m	20h02m
T9	<i>TEMP</i>	[0-9]+ · °·C	40 °C
T10	<i>TIMER</i>	[0-9][0-9]·:[0-9][0-9]	03:00
T11	<i>NUMERO</i>	[0-9]+	15
T12	<i>ID</i>	[A-Za-z] · [A-Za-z0-9]*	numero1
T13	<i>STRING</i>	“·([A-Za-z0-9] ' ' \t \n \r)*·”	“carne mechada”
T14	<i>COMMENT_IN</i>	/·*	/*

T15	<i>COMMENT_OUT</i>	<i>*./</i>	<i>*/</i>
T16	<i>COMMENT</i>	<i>/./</i>	<i>//</i>

Tokens identificados como palabras reservadas:

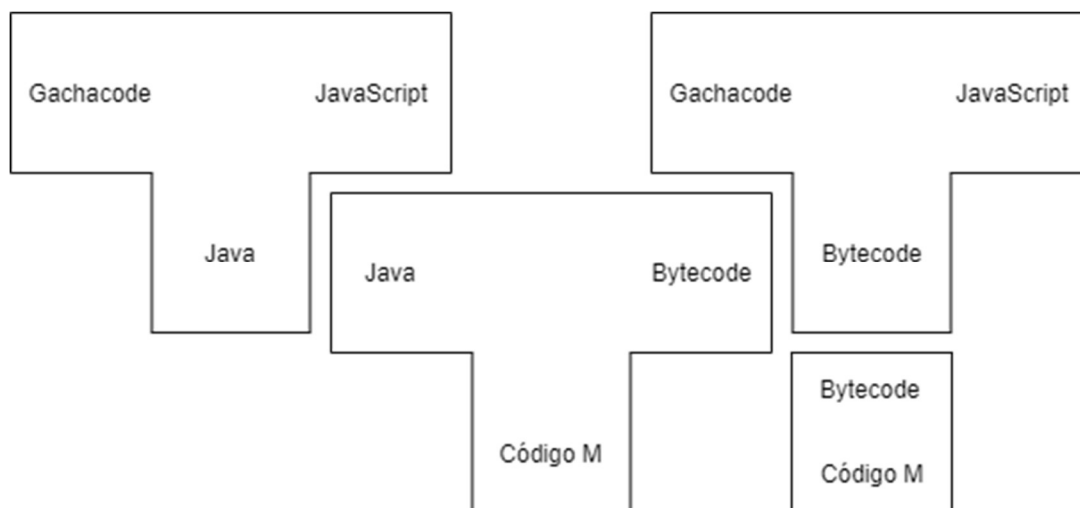
TABLA DE PALABRAS RESERVADAS		
ID	Palabra reservada	Expresión regular
PR1	<i>MENU</i>	<i>m·e·n·u</i>
PR2	<i>PLATO</i>	<i>p·l·a·t·o</i>
PR3	<i>INFO</i>	<i>i·n·f·o</i>
PR4	<i>NOMBRE</i>	<i>n·o·m·b·r·e</i>
PR5	<i>USUARIO</i>	<i>u·s·u·a·r·i·o</i>
PR6	<i>RACIONES</i>	<i>r·a·c·i·o·n·e·s</i>
PR7	<i>TIEMPO</i>	<i>t·i·e·m·p·o</i>
PR7	<i>CALORIAS</i>	<i>c·a·l·o·r·i·a·s</i>
PR8	<i>TIPO</i>	<i>t·i·p·o</i>
PR9	<i>COCINA</i>	<i>c·o·c·i·n·a</i>
PR10	<i>PROGRAMAR</i>	<i>p·r·o·g·r·a·m·a·r</i>
PR11	<i>CALENTAR</i>	<i>c·a·l·e·n·t·a·r</i>
PR12	<i>REMOVER</i>	<i>r·e·m·o·v·e·r</i>
PR13	<i>ANADIR</i>	<i>a·n·a·d·i·r</i>

PR14	SACAR	s·a·c·a·r
PR15	VAROMA	v·a·r·o·m·a
PR16	VELOCIDAD	c·u·c·h·a·r·a e·s·p·i·g·a t·u·r·b·o
PR17	INVERSO	i·n·v·e·r·s·o
PR18	MEDIDA	l m · l g c·u·c·h·a·r·a·d·a u·d
PR19	AL_GUSTO	a·l·_·g·u·s·t·o

4. Diagramas de T

Por último, utilizaremos diagramas de T para representar la construcción del compilador para nuestro lenguaje *Gachacode*.

Para dicha construcción, utilizaremos la técnica de construcción de compiladores que emplea un lenguaje de alto nivel. En nuestro caso, consistirá en el lenguaje de programación Java. Como lenguaje de salida, el compilador genera código en el lenguaje de programación JavaScript, el cual se utilizará para para la visualización de la simulación de la receta introducida en código *Gachacode*.



5. Puntuaciones

Rubén Márquez Villalta	5
Antonio Manjavacas Lucas	5
Alberto Velasco Mata	5
Diego Pedregal Hidalgo	5

Anexo I - Gramática en notación EBNF del lenguaje *Gachacode*

```
<MENU> ::= "menu" <ID> '{' <NOMBRE> {<INGREDIENTE> ';' }
<PLATO> {<PLATO>} '}'
<PLATO> ::= "plato" <ID> '{' <CABECERA> {<INGREDIENTE> ';' }
{<INSTRUCCION>} '}'

<CABECERA> ::= "info" '{' <NOMBRE> <USUARIO> <RACIONES>
<TIEMPO_TOTAL> [<CALORIAS>] [<TIPO>] [<COCINA>] '}'
<NOMBRE> ::= "nombre" ':' <ID> ';'
<USUARIO> ::= "usuario" ':' <STRING> ';'
<RACIONES> ::= "raciones" ':' <NUMERO> ';'
<TIEMPO_TOTAL> ::= "tiempo" ':' <DURACION> ';'
<CALORIAS> ::= "calorias" ':' <NUMERO> ';'
<TIPO> ::= "tipo" ':' <STRING> ';'
<COCINA> ::= "cocina" ':' <STRING> ';'

<INSTRUCCION> ::= (<PROGRAMAR> | <ANADIR> | <CALENTAR> | <REMOVER>
| <ACCION_USUARIO> | <SACAR>) ';'
<PROGRAMAR> ::= "programar" '(' <TEMPERATURA> ',' <VELOCIDAD>
',' <TIMER> [',' "inverso"] ')'
<CALENTAR> ::= "calentar" '(' <TEMPERATURA> ',' <TIMER> ')'
<REMOVER> ::= "remover" '(' <VELOCIDAD> ',' <TIMER> [','
"inverso"] ')'
<ACCION_USUARIO> ::= <STRING>
<ANADIR> ::= "anadir" '(' (<INGREDIENTE> | <STRING>) ')'
<SACAR> ::= "sacar" '(' <STRING> ')'
<TEMPERATURA> ::= (<TEMP> | "varoma")
<VELOCIDAD> ::= (<NUMERO> | "cuchara" | "espiga" | "turbo")
<INGREDIENTE> ::= <CANTIDAD> <STRING>
<CANTIDAD> ::= (<NUMERO> ('l' | 'ml' | 'g' | 'cucharada' | 'ud')
| "al_gusto")

<ID> ::= ('A' | ... | 'Z' | 'a' | ... | 'z') {('A' | ... |
'Z' | 'a' | ... | 'z' | '0' | ... | '9')}
<STRING> ::= '"' {('A' | ... | 'Z' | 'a' | ... | 'z' | '0' | ... |
'9' | <BLANCO>)} '"'

<DURACION> ::= [<NUMERO> 'h'] <NUMERO> 'm'
<TIMER> ::= <DIGITO> <DIGITO> ':' <DIGITO> <DIGITO>
<TEMP> ::= <NUMERO> '°C'
<DIGITO> ::= ('0' | ... | '9')
<NUMERO> ::= <DIGITO> {<DIGITO>}
<BLANCO> ::= (' ' | '\t' | '\n' | '\r')
```