

Procesadores de Lenguajes - Entrega Final

David Gutiérrez Marjalizo, David Carrobles Illán,
Alonso Campillos Gómez Cambronero, Clara Dorado Burgos,
Javier Palomino Gómez

Procesadores de Lenguajes, ESI Ciudad Real - Escuela Superior de
Informática de UCLM, Paseo de la Universidad, Ciudad Real, 13071,
España.

Contributing authors: David.Gutierrez7@alu.uclm.es;
David.Carrobles@alu.uclm.es; Alonso.Campillos@alu.uclm.es;
Clara.Dorado@alu.uclm.es; Javier.Palomino1@alu.uclm.es;

Abstract

El presente trabajo aborda el diseño de un lenguaje específico de dominio (DSL) orientado a la generación de currículums vitae dinámicos. La problemática parte de la rigidez de los formatos tradicionales —como documentos de texto o plantillas PDF— que limitan la personalización y no reflejan adecuadamente el perfil técnico y creativo de los profesionales del ámbito de la informática y la ingeniería. Se propone una gramática en notación EBNF que permite definir de forma clara y estructurada la información personal, formación académica, experiencia profesional, habilidades e historial de proyectos. A partir de esta especificación, el procesador del lenguaje analiza, valida y traduce la entrada para generar automáticamente un currículum digital en formato web, ofreciendo una salida flexible, interactiva y fácilmente actualizable. Con ello, se pretende modernizar la manera de presentar la información profesional y destacar las competencias de los candidatos en un entorno altamente competitivo.

Keywords: Procesadores de lenguajes, DSL, EBNF, currículum vitae digital, análisis sintáctico, generación de código, web dinámica

Contents

1 Descripción del problema	3
2 Tablas de Tokens	4
2.1 Palabras reservadas	4
2.2 Símbolos y operadores	5
2.3 Identificadores y números	5
3 EBNF del lenguaje diseñado	6
4 Semántica en lenguaje natural	7
4.1 Definición Léxica y Tipos Básicos	7
4.2 Variables y Estructura Global	8
4.3 Fechas y Tiempo	8
4.4 Datos Personales (DP)	8
4.5 Formación	8
4.6 Idiomas	8
4.7 Experiencia y Habilidades	9
4.8 Portafolio	9
5 Salida esperada	9
6 Construcción del procesador en diagrama T	10
7 Puntuación de cada uno de los miembros del grupo	12

1 Descripción del problema

El **currículum vitae** es, en la actualidad, la **principal herramienta de presentación** en el ámbito académico y profesional. Sin embargo, su elaboración continúa dependiendo casi exclusivamente de **formatos tradicionales** como *documentos de texto, plantillas prediseñadas o archivos PDF*. Estas soluciones, aunque ampliamente utilizadas, presentan **importantes limitaciones**: son **poco flexibles**, ofrecen **escasas posibilidades de personalización** y rara vez logran **reflejar adecuadamente** las capacidades técnicas de perfiles vinculados a la *informática* y la *ingeniería*.

Esta situación genera una **paradoja evidente**: profesionales que dedican años a adquirir competencias en **programación, diseño de sistemas y resolución de problemas**, terminan utilizando medios **rígidos y estáticos** que no hacen justicia a su **potencial**. Como consecuencia, su currículum se convierte en un **documento plano**, que no transmite la **creatividad, la innovación ni el dominio tecnológico** que caracteriza su perfil.

El **problema a resolver** consiste en **trascender los formatos tradicionales** de currículum mediante el diseño de un **lenguaje específico de dominio (DSL)** que permita describir un CV **como si fuera código**. Este lenguaje debe facilitar la definición **clara y estructurada** de:

- **Información personal**
- **Formación académica**
- **Experiencia profesional**
- **Habilidades**
- **Proyectos**

A partir de ese archivo, el sistema procesará la información y generará automáticamente una **representación digital dinámica y atractiva** en forma de **página web**, que explote la **diferenciación** de cada usuario frente a otros candidatos.

2 Tablas de Tokens

2.1 Palabras reservadas

Token	Lexema de ejemplo	Patrón
CV	cv	cv
GVAR	gvar	gvar
LVAR	lvar	lvar
DP	datospersonales	datospersonales
NOMYAPE	nomyape	nomyape
FOTO	foto	foto
FECHA	fecha	fecha
BIO	bio	bio
CONTACTO	contacto	contacto
EMAIL	email	email
TELEFONO	telefono	telefono
REDES	redes	redes
LINKEDIN	linkedin	linkedin
GITHUB	github	github
WEB	web	web
FORMACION	formacion	formacion
OFICIAL	oficial	oficial
TITULO	titulo	titulo
EXPEDIDOR	expedidor	expedidor
DESCRIPCION	descripcion	descripcion
LOGROS	logros	logros
COMPLEMENTARIA	complementaria	complementaria
CERTIFICADO	certificado	certificado
HORAS	horas	horas
IDIOMAS	idiomas	idiomas
IDIOMA	idioma	idioma
NIVEL	nivel	nivel
EXPERIENCIA	experiencia	experiencia
LABORAL	laboral	laboral
PUESTO	puesto	puesto
RESPONSABILIDADES	responsabilidades	responsabilidades
VOLUNTARIADO	voluntariado	voluntariado
ORGANIZACION	organizacion	organizacion
HABILIDADES	habilidades	habilidades
SOFT	soft	soft
HARD	hard	hard
HABILIDAD	abilidad	abilidad
NVHAB	nvhab	nvhab
CATEGORIA	categoria	categoria
PORTAFOLIO	portafolio	portafolio
PROYECTO	projeto	projeto
NOMBRE	nombre	nombre
GRUPO	grupo	grupo
COMPANERO	companero	companero
TECNOLOGIAS	tecnologias	tecnologias
MERITOS	meritos	meritos

2.2 Símbolos y operadores

Token	Lexema de ejemplo	Patrón
LL_A	{	{
LL_C	}	}
PA_A	((
PA_C))
IG	=	=
PYC	;	;
CO	,	,

2.3 Identificadores y números

Token	Lexema de ejemplo	Patrón
COMMENT	/* Hola */	/* . */
TFNO	622756985	[5-9] [0-9] {8}
NUM	123	(0 [1-9] [0-9] *)
PAL	Hola	[A-Za-zÁÉ...Ññ]+
IDENT	"Var1"	" CONJPALYNUM "
MAIL	user@mail.com	[A-Za-z0-9]+ ('.' [A-Za-z0-9]+)* '@' [A-Za-z0-9]+ ('.' [A-Za-z0-9]+)*
RUTA	C:/fotos/img.png	[A-Za-z]+ ([./\:-]+ [A-Za-z]+)+
FECHA_NUM	20/01/2026	('0' [1-9] [12] [0-9] '30' '31') '/' ('0' [1-9] '1' [0-2]) '/' ([12] [0-9] {3})
BOOL	No	Si No
NVI	A1	([ABC] [12]) nativo
NVH	bajo	bajo medio alto
WS		[\t \r \n]+
CONJPALYNUM	Pizza 4 quesos	({PAL} {NUM}) ([,] ? ({PAL} {NUM}) \. ?)*

3 EBNF del lenguaje diseñado

A continuación se presenta la gramática en EBNF diseñada:

```
(* ===== Simbolo inicial ===== *)
cvs      ::= {global_var} cv {cv} ;

(* ===== Variables ===== *)
global_var   ::= GVAR LL_A variable {variable} LL_C ;
local_var    ::= LVAR LL_A variable {variable} LL_C ;
variable     ::= IDENT IG CONJPALYNUM PYC ;

(* ===== Curriculvbum Vitae ===== *)
cv          ::= CV IDENT LL_A [local_var] datospersonales formacion [idiomas] [experiencia] [ha

(* ===== Datos personales ===== *)
datospersonales ::= DP LL_A nomyape [foto] fecha [bio] contacto LL_C ;
nomyape       ::= NOMYAPE PA_A CONJPALYNUM PA_C
                  |NOMYAPE PA_A IDENT PA_C;
foto          ::= FOTO PA_A RUTA PA_C ;
fecha         ::= FECHA PA_A FECHA_NUM PA_C ;
bio           ::= BIO PA_A CONJPALYNUM PA_C
                  |BIO PA_A IDENT PA_C;
contacto     ::= CONTACTO LL_A email telefono redes LL_C ;
email         ::= EMAIL PA_A MAIL PA_C ;
telefono      ::= TELEFONO PA_A TFNO PA_C ;
redes         ::= REDES LL_A linkedin [github] [web] LL_C
                  |REDES LL_A github [web] LL_C
                  |REDES LL_A web LL_C;
linkedin      ::= LINKEDIN PA_A RUTA PA_C ;
github         ::= GITHUB PA_A RUTA PA_C ;
web           ::= WEB PA_A RUTA PA_C ;

(* ===== Formación ===== *)
formacion    ::= FORMACION LL_A oficial {oficial} {complementaria} LL_C ;
oficial       ::= OFICIAL LL_A titulo expedidor [descripcion] [logros] fecha LL_C ;
titulo        ::= TITULO PA_A CONJPALYNUM PA_C
                  |TITULO PA_A IDENT PA_C;
expedidor     ::= EXPEDIDOR PA_A CONJPALYNUM PA_C
                  |EXPEDIDOR PA_A IDENT PA_C;
descripcion   ::= DESCRIPCION PA_A CONJPALYNUM PA_C
                  |DESCRIPCION PA_A IDENT PA_C;
logros        ::= LOGROS PA_A CONJPALYNUM PA_C
                  |LOGROS PA_A IDENT PA_C;
complementaria ::= COMPLEMENTARIA LL_A titulo [certificado] expedidor [horas] fecha LL_C;
certificado   ::= CERTIFICADO PA_A BOOL PA_C;
horas         ::= HORAS PA_A NUM PA_C;

(* ===== Idiomas ===== *)
idiomas       ::= IDIOMAS LL_A idioma {idioma} LL_C ;
idioma        ::= IDIOMA LL_A nombre nvi [expedidor] LL_C ;
```

```

nvi          ::= NIVEL PA_A NVI PA_C ;

(* ===== Experiencia ===== *)
experiencia  ::= EXPERIENCIA LL_A laboral {laboral} {voluntariado} LL_C
                |EXPERIENCIA LL_A voluntariado {voluntariado} LL_C;
laboral       ::= LABORAL LL_A organizacion puesto horas [responsabilidades] LL_C;
puesto        ::= PUESTO PA_A CONJPALYNUM PA_C
                |PUESTO PA_A IDENT PA_C;
responsabilidades ::= RESPONSABILIDADES PA_A CONJPALYNUM PA_C
                |RESPONSABILIDADES PA_A IDENT PA_C;
voluntariado  ::= VOLUNTARIADO LL_A puesto descripcion horas organizacion LL_C;
organizacion  ::= ORGANIZACION PA_A CONJPALYNUM PA_C
                |ORGANIZACION PA_A IDENT PA_C;

(* ===== Habilidades ===== *)
habilidades   ::= HABILIDADES LL_A soft [hard] LL_C
                |HABILIDADES LL_A hard LL_C ;
soft          ::= SOFT LL_A habilidad {CO habilidad} LL_C;
hard          ::= HARD LL_A categoria {categoria} LL_C;
nvh           ::= NVHAB PA_A NVH PA_C;
habilidad     ::= HABILIDAD PA_A CONJPALYNUM PA_C
                |HABILIDAD PA_A IDENT PA_C;
categoria     ::= CATEGORIA LL_A nombre habilidad nvh {CO habilidad nvh} LL_C;

(* ===== Portafolio ===== *)
portafolio    ::= PORTAFOLIO LL_A proyecto {proyecto} {merito} LL_C
                |PORTAFOLIO LL_A merito {merito} LL_C;
proyecto      ::= PROYECTO LL_A nombre [grupo] descripcion tecnologias [web] LL_C;
nombre        ::= NOMBRE PA_A CONJPALYNUM PA_C
                |NOMBRE PA_A IDENT PA_C;
grupo         ::= GRUPO LL_A companero {companero} LL_C;
companero     ::= COMPANERO LL_A nomyape [github] LL_C;
tecnologias   ::= TECNOLOGIAS PA_A CONJPALYNUM PA_C
                |TECNOLOGIAS PA_A IDENT PA_C;
merito        ::= MERITOS LL_A nombre descripcion LL_C;

```

4 Semántica en lenguaje natural

4.1 Definición Léxica y Tipos Básicos

- **LET:** Representa cualquier letra del alfabeto (mayúsculas A-Z y minúsculas a-z).
- **DIG:** Un dígito numérico del 1 al 9.
- **NUM:** Un número entero formado por un dígito inicial o 0, seguido de más dígitos.
- **PALABRA:** Secuencia de letras (LET) consecutivas.
- **CARACTER_ESPECIAL:** Símbolos de puntuación y caracteres especiales permitidos en el texto.
- **TXT:** Cadena de texto compuesta por palabras y caracteres especiales separados por espacios.

- **VAL:** Unidad de información genérica que puede ser texto (**TXT**) o número (**NUM**).
- **RTA:** Cadena específica para direcciones de archivos o URLs, permitiendo caracteres como puntos, barras y guiones.
- **MAIL:** Cadena que valida el formato estándar de correo electrónico (`usuario@dominio.extensión`).

4.2 Variables y Estructura Global

- **CVS:** Nodo raíz del lenguaje. Permite definir una lista opcional de variables globales seguida de uno o varios currículums.
- **GVAR (Global Variables):** Bloque para definir variables que pueden ser utilizadas en múltiples currículums (ej. constantes para nombres de empresas recurrentes).
- **CV:** Define un currículum individual. Contiene variables locales y las secciones de datos.
- **LVAR (Local Variables):** Bloque de variables específicas para un CV concreto.
- **ID:** Identificador único del currículum.

4.3 Fechas y Tiempo

El lenguaje maneja fechas explícitas para validar la consistencia temporal:

- **DIA:** Validado para aceptar rangos lógicos (01-31).
- **MES:** Validado para los 12 meses del año (01-12).
- **AÑO:** Estructura de cuatro dígitos que comienza por 1 o 2 (ej. 1990, 2024).

4.4 Datos Personales (DP)

Bloque obligatorio con la información del candidato:

- **NOMYAPE:** Nombre completo del usuario.
- **FOTO:** Ruta a la imagen de perfil.
- **BIO:** Breve descripción biográfica.
- **CONTACTO:** Incluye email, un teléfono de exactamente 9 dígitos y redes sociales opcionales.

4.5 Formación

Se divide en dos categorías:

- **OFICIAL:** Educación formal. Requiere título, expedidor y una fecha de finalización o la cadena literal "`En_curso`".
- **COMPLEMENTARIA:** Cursos extra. Incluye un campo **CERTIFICADO** que acepta explícitamente los valores "`Si`" o "`No`".

4.6 Idiomas

Define las competencias lingüísticas:

- **NIVEL:** Restringido a los niveles del Marco Común Europeo (A1 a C2) o la palabra clave "nativo".

4.7 Experiencia y Habilidades

- **EXPERIENCIA:** Puede ser **LABORAL** (remunerada, con responsabilidades) o **VOLUNTARIADO** (centrada en la organización y descripción).
- **HABILIDADES:** Se clasifican en **SOFT** (blandas) y **HARD** (técnicas).
- **NVHAB:** Nivel de habilidad técnica restringido a "bajo", "medio" o "alto".

4.8 Portafolio

Agrupa la demostración práctica de conocimientos:

- **PROYECTO:** Trabajos realizados, permitiendo detallar el grupo de compañeros (con sus enlaces a GitHub) y las tecnologías usadas.
- **MERITOS:** Reconocimientos puntuales que no constituyen un proyecto completo.

5 Salida esperada

La salida esperada será una **página web** que genere automáticamente un currículum vitae digital. Este CV será visualmente atractivo, interactivo y adaptado al perfil de cada usuario. Además, será fácilmente actualizable a partir de la descripción en el lenguaje diseñado. Esta página web estará escrita en el lenguaje HTML que será el lenguaje de salida del procesador del lenguaje que implementaremos, por lo que una vez obtenida la salida en HTML cada web podrá usarla de forma que se adapte al perfil de cada usuario.

6 Construcción del procesador en diagrama T

Para nuestra implementación hemos decidido usar Python, pero además de los diagramas necesarios para esto, hemos pensado también otros en caso de que en el futuro decidamos usar Java.

Vamos a empezar con los diagramas para Python:

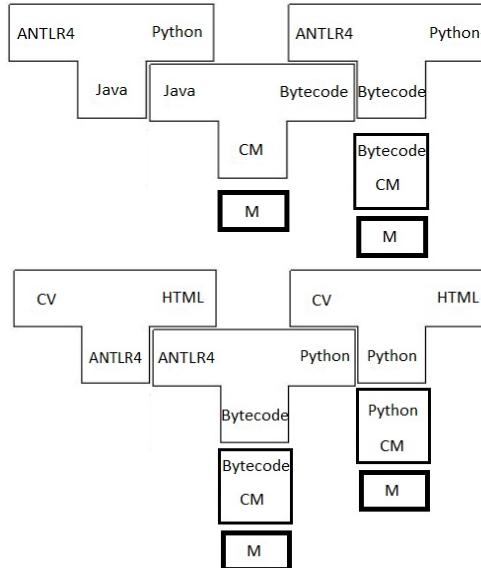


Diagrama de Python

Empezamos desde el diagrama de arriba, como vamos a usar el lenguaje ANTLR4 para definir nuestro lenguaje, que esta implementado en Java, y de esta manera nos permite definir las fases de análisis y síntesis ya que por debajo de ANTLR4 corre Java. Además conocemos el compilador javac que es el diagrama que nos encontramos en medio, tras este paso obtenemos un nuevo procesador que nos permite pasar de ANTLR4 a python usando bytecode.

En un segundo paso vamos ya a pasar el CV a HTML usando ANTLR4 y el procesador que acabamos de obtener del paso anterior, esto nos permite hacer la implementación de CV a HTML en Python, que es un lenguaje interpretado.

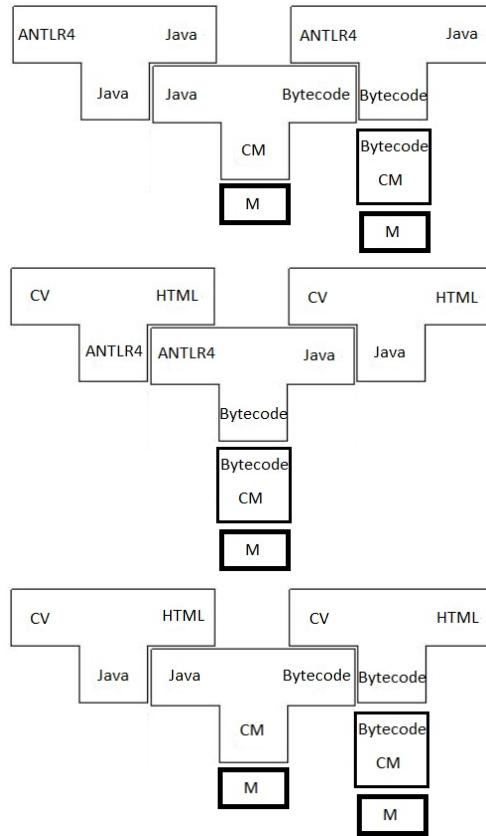


Diagrama de Java

En este diagrama T, encontramos la definición de nuestra transformación de lenguajes si nuestro sistema tuviese que usar Java. Al igual que en el anterior, podemos observar como en la primera construcción ANTLR4 se traduce a Java, que sería nuestro lenguaje empleado en este caso. En el primer paso usamos javac como en el caso anterior. Así obtenemos el compilador necesario para el siguiente paso. En el tercer paso usamos de nuevo javac para procesar la salida del paso 2 y obtener una implementación en bytecode que pueda correr en la máquina.

7 Puntuación de cada uno de los miembros del grupo

- David Gutiérrez Marjalizo: 5
- David Carrobles Illán: 5
- Alonso Campillos Gómez Cambronero: 5
- Clara Dorado Burgos: 5
- Javier Palomino Gómez: 5