

Procesadores de Lenguajes - Entrega 1

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 EBNF del lenguaje diseñado	4
3 Semántica en lenguaje natural	6
3.1 Elementos generales	6
3.2 Símbolo inicial	6
3.3 Datos Personales (DP)	6
3.4 Formación	7
3.5 Idiomas	7
3.6 Experiencia	7
3.7 Habilidades	7
3.8 Portafolio	8
4 Salida esperada	8
5 Construcción del procesador en diagrama T	9
6 Puntuación de cada uno de los miembros del grupo	11

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 EBNF del lenguaje diseñado

A continuación se presenta la gramática en EBNF diseñada:

```
(* ===== Símbolo inicial ===== *)
CV         ::= "{" DP FORMACION [IDIOMAS] [EXPERIENCIA]
[HABILIDADES] [PORTAFOLIO] "}" ;

(* ===== Elementos básicos ===== *)
VAL        ::= TXT | NUM;
TXT        ::= (PALABRA | CARACTER_ESPECIAL) { " " (PALABRA
| CARACTER_ESPECIAL) } ;
PALABRA   ::= LET {LET} ;
LET        ::= "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" ;
NUM       ::= ("0" | DIG {DIG | "0"}) ;
DIG        ::= "1" | "2" | "3" | "4" | "5" | "6" | "7" | "8" | "9" ;
RUTA      ::= LET {LET | NUM | CARACTER_RUTA} ;
CARACTER_RUTA ::= "/" | ":" | "-" | ";" | "." | ","
CARACTER_ESPECIAL ::= "!" | "\"" | "#" | "$" | "%" | "&" | ","
| "(" | ")" | "*" | "+" | "," | "-" | "." | "/" | ":" | ";"
| "<" | "=" | ">" | "?" | "@" | "[" | "\\" | "]" | "^" | ""
| `` | {" | "}" | "}" | "~" ;

(* ===== Datos personales ===== *)
DP          ::= "datospersonales{", NOMBREAPE [FOTO] FECHA
[BIO] CONTACTO, "}" ;
NOMBREAPE   ::= "nomyape(", TXT, ")" ;
FOTO        ::= "foto(", RUTA, ")" ;
FECHA      ::= "fecha(", (DIG | 0) (DIG | 0), "/", (DIG
| 0) (DIG | 0), "/",
(DIG | 0) (DIG | 0) (DIG | 0), ")" ;
BIO         ::= "bio(", VAL {VAL}, ")" ;
CONTACTO    ::= "contacto{", EMAIL TELEFONO [REDES], "}" ;
EMAIL       ::= "email(", PALABRA {PALABRA | NUM | '.' |
'_'}, "@", PALABRA {'.' PALABRA} '.' PALABRA, ")" ;
TELEFONO    ::= "telefono(", NUM, ")" ;
REDES       ::= "redes{", [LINKEDIN] [GITHUB] [WEB], "}" ;
LINKEDIN    ::= "linkedin(", TXT, ")" ;
GITHUB      ::= "github(", TXT ")" ;
WEB         ::= "web(", TXT, ".", TXT ")" ;

(* ===== Formación ===== *)
```

```

FORMACION      ::= "formacion\"", OFICIAL {OFICIAL}
{COMPLEMENTARIA}, ")" ;
OFICIAL        ::= "oficial\"", TITULO (FECHA | "En_curso")
EXPEDIDOR [DESCRIPCION] [LOGROS], ")" ;
TITULO         ::= "titulo\"", TXT, ")" ;
EXPEDIDOR     ::= "expedidor\"", TXT, ")" ;
DESCRIPCION    ::= "descripcion\"", VAL {VAL}, ")" ;
LOGROS         ::= "logros\"", VAL {VAL}, ")" ;
COMPLEMENTARIA ::= "complementaria\"", TITULO [CERTIFICADO]
(FECHA | "En_curso") EXPEDIDOR [HORAS], ")" ;
CERTIFICADO    ::= "certificado\"", VAL {VAL}, ")" ;
HORAS          ::= "horas\"", NUM ")" ;

(* ===== Idiomas ===== *)
IDIOMAS        ::= "idiomas\"", {IDIOMA}, ")" ;
IDIOMA         ::= "idioma\"", PALABRA, NIVEL, [EXPEDIDOR],
[CERTIFICADO], ")" ;
NIVEL          ::= "nivel\"", ("A" | "B" | "C"), ("1" | "2") |
"nativo", ")" ;

(* ===== Experiencia ===== *)
EXPERIENCIA   ::= "experiencia\"", {LABORAL}, {VOLUNTARIADO},
)" ;
LABORAL        ::= "laboral\"", PUESTO, HORAS,
[RESPONSABILIDADES], ")" ;
RESPONSABILIDADES ::= "responsabilidades\"", VAL, {VAL}, ")" ;
VOLUNTARIADO   ::= "voluntariado\"", PUESTO, DESCRIPCION, HORAS,
ORGANIZACION, ")" ;
ORGANIZACION   ::= "organizacion\"", VAL {VAL}, ")" ;

(* ===== Habilidades ===== *)
HABILIDADES   ::= "habilidades\"", {SOFT}, {HARD}, ")" ;
SOFT           ::= "soft\"", HABILIDAD, NIVEL, ")" ;
HARD           ::= "hard\"", HABILIDAD, CATEGORIA, DESCRIPCION,
NIVEL, ")" ;
HABILIDAD      ::= "habilidad\"", VAL {VAL}, ")" ;
CATEGORIA      ::= "categoria\"", TXT, ")" ;

(* ===== Portafolio ===== *)
PORTAFOLIO    ::= "portafolio\"", {PROYECTO} {MERITOS}, ")" ;
PROYECTO       ::= "proyecto\"", NOMBRE [GRUPO] DESCRIPCION
CATEGORIA TECNOLOGIAS [WEB], ")" ;
NOMBRE         ::= "nombre\"", VAL {VAL}, ")" ;
GRUPO          ::= "grupo\"", COMPAÑERO {COMPAÑERO}, ")" ;
COMPAÑERO      ::= "compañero\"", NOMYAPE [GITHUB], ")" ;

```

```

CATEGORIA      ::= "categoria(", TXT, ")" ;
TECNOLOGIAS   ::= "tecnologias{", VAL {VAL}, "}" ;
MERITOS        ::= "meritos{", NOMBRE DESCRIPCION, "}" ;
WEB            ::= "web(", TXT, ".", TXT, ")" ;

```

3 Semántica en lenguaje natural

3.1 Elementos generales

- **VAL**: valor textual o numérico que puede usarse como unidad básica de información en títulos, descripciones, nombres, etc.
- **TXT**: cadena formada por palabras y/o caracteres especiales, separadas opcionalmente por espacios.
- **PALABRA**: secuencia de letras (mayúsculas o minúsculas).
- **NUM**: número entero positivo compuesto por dígitos (sin ceros iniciales, salvo en posiciones posteriores).
- **LET**: cualquier letra del alfabeto en mayúscula o minúscula.
- **DIG**: cualquier dígito entre 1 y 9.
- **RTA**: secuencia de letras, números o caracteres de ruta (/,\,.,-,_,:, espacio). Se utiliza para identificar rutas de archivos.
- **CARACTER_ESPECIAL**: conjunto de símbolos de puntuación y caracteres no alfabéticos que pueden formar parte de textos.
- **CARACTER_RUTA**: representa cualquier carácter que puede aparecer en una ruta de archivo o URL, además de letras y números

3.2 Símbolo inicial

- **CV**: representa un currículum completo. Está delimitado por llaves {} e incluye:
 - Datos personales obligatorios (**DP**).
 - Formación obligatoria (**FORMACION**).
 - Opcionalmente: idiomas (**IDIOMAS**), experiencia (**EXPERIENCIA**), habilidades (**HABILIDADES**) y portafolio (**PORTAFOLIO**).

3.3 Datos Personales (DP)

- **DP**: agrupa la información básica de identificación del usuario.
- **NOMYAPE**: nombre y apellidos de la persona.
- **FOTO**: ruta a un archivo de imagen que representa la fotografía del usuario.
- **FECHA**: fecha de nacimiento en formato DD/MM/AAAA.
- **BIO**: breve texto descriptivo de la persona (perfil profesional o resumen personal).
- **CONTACTO**: información de contacto obligatoria.
- **EMAIL**: dirección de correo electrónico con formato estándar.
- **TELEFONO**: número de teléfono compuesto por dígitos.
- **REDES**: enlaces opcionales a perfiles en línea.

- **LINKEDIN, GITHUB, WEB**: direcciones específicas a redes sociales o páginas web.

3.4 Formación

- **FORMACION**: conjunto de estudios oficiales y complementarios.
- **OFICIAL**: formación reglada, con título, fecha (o estado en curso), entidad expedidora, y opcionalmente descripción y logros.
- **COMPLEMENTARIA**: cursos, talleres o capacitaciones no oficiales, que incluyen título, fecha o estado, entidad, horas de duración y posible certificado.
- **EXPEDIDOR**: institución o entidad que emite la formación.
- **DESCRIPCION**: texto explicativo del contenido de la formación.
- **LOGROS**: resultados, reconocimientos o habilidades adquiridas.
- **CERTIFICADO**: constancia oficial del curso.
- **HORAS**: carga horaria dedicada a la actividad formativa.
- **TITULO**: denominación del título académico o curso principal.

3.5 Idiomas

- **IDIOMAS**: bloque que lista las lenguas que el usuario domina.
- **IDIOMA**: nombre del idioma (por ejemplo, inglés, francés).
- **NIVEL**: grado de competencia lingüística, expresado como A1, A2, B1, B2, C1, C2 o la palabra **nativo**.
- **EXPEDIDOR**: institución que certifica el nivel (opcional).
- **CERTIFICADO**: acreditación formal del dominio del idioma (opcional).

3.6 Experiencia

- **EXPERIENCIA**: bloque que recoge la trayectoria práctica del usuario.
- **LABORAL**: experiencia profesional remunerada, con puesto, carga horaria y responsabilidades principales.
- **PUESTO**: nombre o título del cargo desempeñado.
- **RESPONSABILIDADES**: lista de tareas y funciones realizadas en el puesto.
- **VOLUNTARIADO**: experiencia no remunerada, con puesto, descripción de la actividad, carga horaria y entidad organizadora.
- **ORGANIZACION**: institución o grupo responsable del voluntariado.

3.7 Habilidades

- **HABILIDADES**: bloque que agrupa las competencias del usuario.
- **SOFT**: habilidades blandas, como liderazgo, trabajo en equipo o comunicación, con nivel asociado.
- **HARD**: habilidades técnicas, vinculadas a conocimientos concretos, con categoría, descripción y nivel.
- **HABILIDAD**: denominación de la competencia.
- **CATEGORIA**: clasificación de la habilidad técnica (por ejemplo, programación, diseño).

- **DESCRIPCION:** detalle adicional de la habilidad.
- **NIVEL:** grado de dominio de la habilidad (básico, intermedio, avanzado).

3.8 Portafolio

- **PORATAFOLIO:** bloque que reúne proyectos y méritos destacados.
- **PROYECTO:** experiencia práctica concreta, con nombre, descripción, categoría, tecnologías empleadas, grupo de trabajo y enlace web opcional.
- **NOMBRE:** título o denominación del proyecto.
- **GRUPO:** conjunto de compañeros participantes en el proyecto.
- **COMPAÑERO:** persona colaboradora, identificada por nombre y opcionalmente perfil de GitHub.
- **TECNOLOGIAS:** herramientas, lenguajes o frameworks utilizados.
- **MERITOS:** logros, premios o reconocimientos vinculados a la trayectoria.

4 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 sera 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.

5 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:

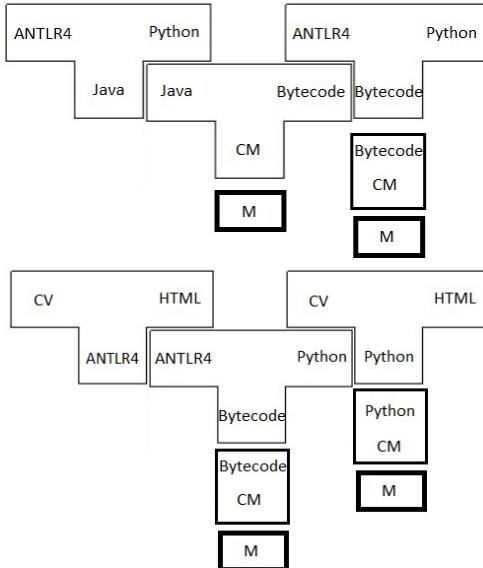


Diagrama de Python

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.

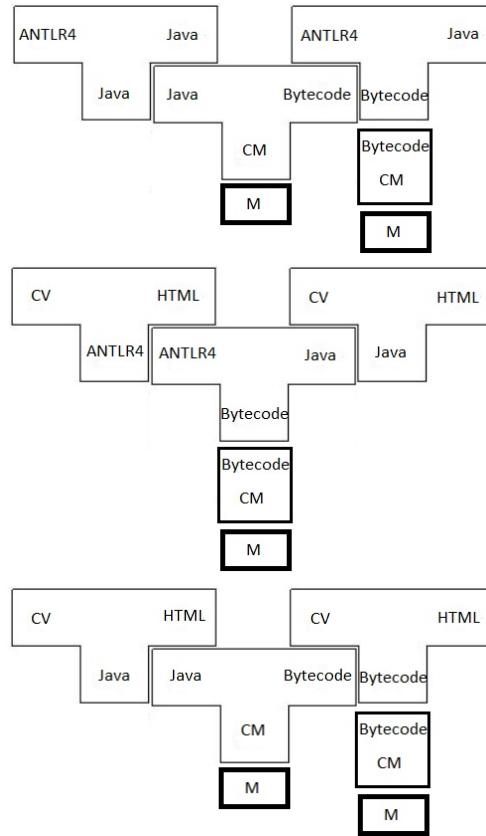


Diagrama de Java

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.

6 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