



---

# PRÁCTICA 2: GRAMÁTICAS Y GENERADORES AUTOMÁTICOS

---

PROCESADORES DEL LENGUAJE



DIEGO DÍAZ VIDAL, LUCÍA MONZÓN VILLALBA Y FERNANDO  
CEMBELLÍN SECO  
GRADO EN INGENIERÍA INFORMÁTICA  
CURSO 2025-2026



## Contenido

Introducción .....	3
Objetivos.....	3
Tecnologías utilizadas .....	3
Planteamiento .....	4
Ámbito del proyecto .....	4
Organización .....	4
1. Primera parte: Generación de árboles sintácticos para lenguajes específicos .....	7
1.1. Árbol sintáctico de un CSV .....	7
1.1.1. Nivel básico .....	7
1.1.2. Nivel medio .....	7
1.2. Lenguaje de programación mínimo .....	8
1.2.1. Nivel básico .....	8
1.2.2. Nivel medio .....	9
2. Segunda parte: Traducción e interpretación de manuales de IKEA .....	11
2.1. Introducción y planteamiento .....	11
2.2. Nivel básico .....	12
2.2.1. Definición del lenguaje .....	12
2.2.2. Traducción de los manuales .....	14
2.2.3. Creación del Lexer y el Parser .....	18
2.2.4. Construcción del AST .....	20
2.3. Nivel avanzado .....	34
2.3.1. Desarrollo.....	34
Conclusiones .....	35

## Introducción

El presente documento tiene como finalidad describir, explicar y justificar las decisiones de diseño e implementación de la Práctica 2 de la asignatura, centrada en el uso de gramáticas y generadores automáticos.

La práctica aborda tanto la generación de analizadores léxicos y sintácticos con ANTLR como la construcción de árboles sintácticos abstractos (AST) y su representación legible. Además, se incluyen nociones iniciales de tabla de símbolos y gestión de errores.

Para facilitar la lectura, la memoria se organiza siguiendo la misma estructura que el enunciado de la práctica, diferenciando las dos partes principales y sus subtareas.

## Objetivos

Los objetivos principales de la práctica son los siguientes:

- Comprender el funcionamiento de una herramienta de generación automática de intérpretes de lenguajes.
- Ser capaces de definir y construir analizadores léxicos y sintácticos para lenguajes concretos.
- Aprender a generar el árbol sintáctico abstracto (AST) asociado a una entrada y representarlo de manera legible.
- Iniciarse en el uso de la tabla de símbolos para almacenar información semántica relevante.
- Familiarizarse con el manejo de errores léxicos y sintácticos en un lenguaje definido por una gramática formal.
- Aplicar los conocimientos en el diseño de un lenguaje propio, en este caso relacionado con manuales de montajes de IKEA.

## Tecnologías utilizadas

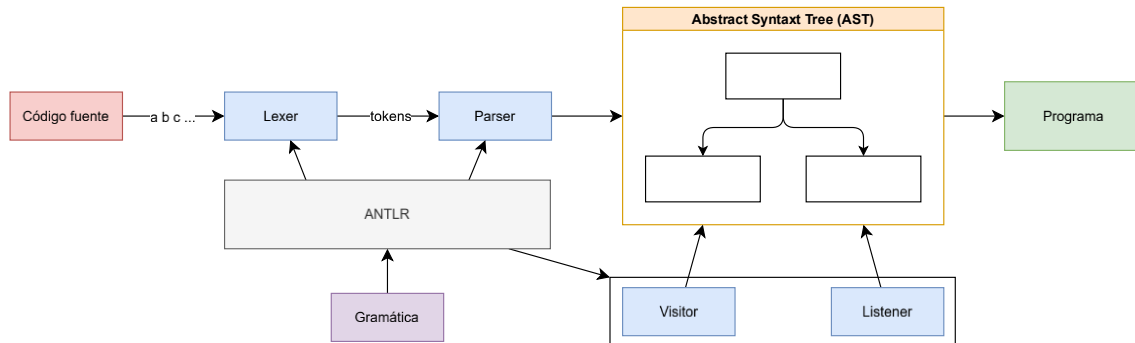
Las tecnologías y herramientas empleadas para la realización de la práctica son las siguientes:

- **Lenguaje:** Java, por su compatibilidad con ANTLR y su robustez para la generación automática de analizadores.
- **IDE:** Visual Studio Code, configurado con extensiones para ANTLR y soporte de ejecución de proyectos Java.
- **Generador:** ANTLR, usado para definir gramáticas léxicas y sintácticas y generar los analizadores y AST correspondientes.
- **Control de versiones:** GitHub, para mantener el historial de cambios y facilitar el trabajo colaborativo.
- **Trabajo colaborativo:** Microsoft OneDrive, con una carpeta compartida para el desarrollo conjunto de la memoria, en Word, y un Excel con un diagrama de Gantt que recoge las tareas, los responsables y los plazos.

## Planteamiento

### Ámbito del proyecto

Con el objetivo de tener una representación visual del proyecto a la que poder acudir para aclarar dudas respecto al ámbito y contexto, se ha desarrollado, con ayuda del vídeo “[Introducción e instalación ANTLR](#)”, del aula virtual de la asignatura, un diagrama que representa la estructura completa de un “Procesador del Lenguaje”.



A continuación, se describe cada elemento del diagrama:

- **Código fuente:** el texto de la entrada escrito en el lenguaje diseñado. Por ejemplo, para la segunda parte, el código fuente son las traducciones normalizadas de los manuales de IKEA.
- **ANTLR:** generador de analizadores léxicos y sintácticos.
- **Gramática:** conjunto de símbolos terminales y no terminales que se generan en base a unas expresiones regulares. Estos símbolos permiten definir las palabras aceptadas por un lenguaje formal.
- **Lexer:** analizador léxico. Recibe el código fuente carácter a carácter y lo transforma en una secuencia de tokens, cada uno asociado a un tipo.
- **Parser:** analizador sintáctico. Recibe la secuencia de tokens generada por el lexer y verifica que se cumplen las reglas de la gramática. También construye un árbol de análisis sintáctico para representar de forma jerárquica la estructura lógica del programa.
- **Visitor:** patrón de diseño que ANTLR genera automáticamente.
- **Listener:** mecanismo generado por ANTLR para recorrer el AST.
- **AST:** árbol de sintaxis abstracta. Representación simplificada y estructural del programa.
- **Programa:** resultado lógico del procesamiento del lenguaje. Abstracción que puede ser analizada o transformada.

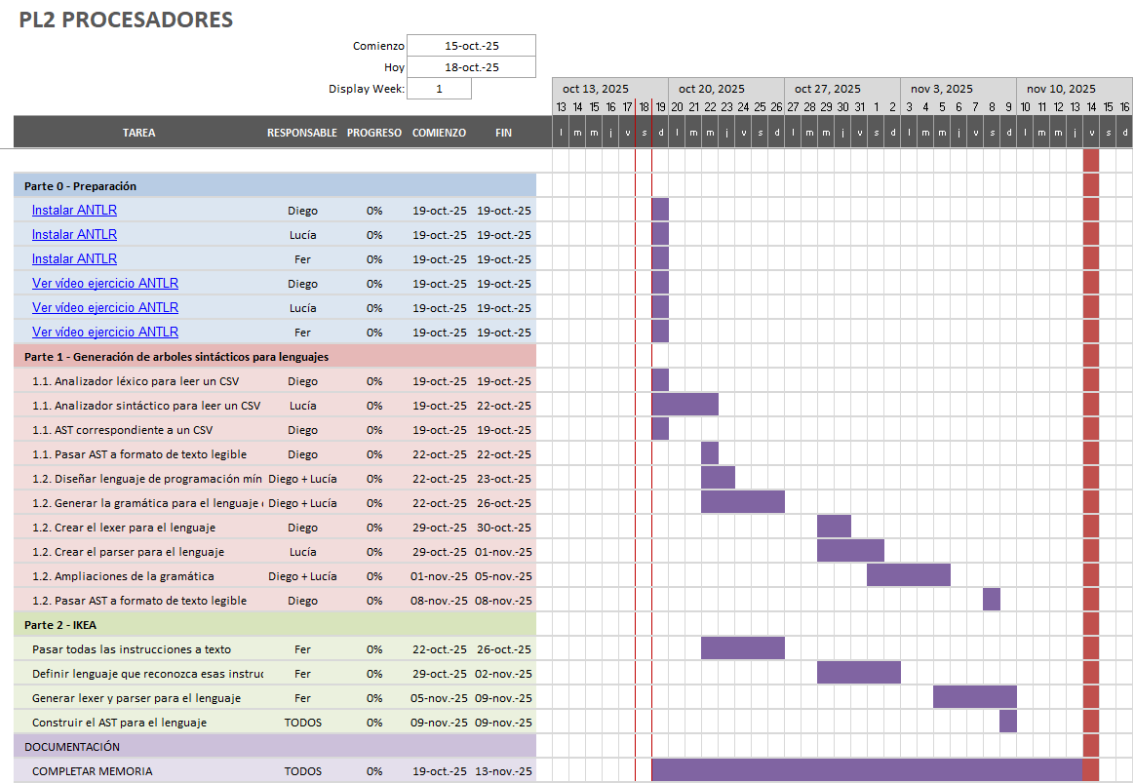
### Organización

Antes de comenzar con la implementación, se realizó una planificación detallada del trabajo con el fin de estructurar las tareas, distribuir responsabilidades y establecer un calendario de ejecución realista.

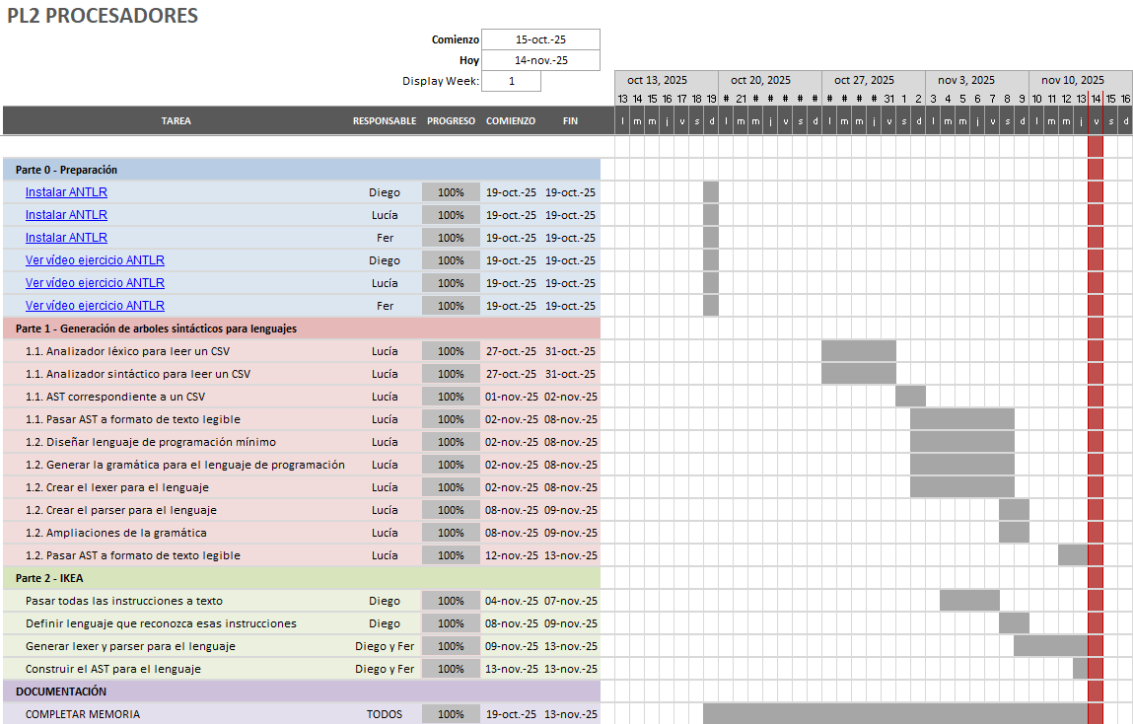
Para ello, se elaboró un diagrama de Gantt que refleja la división de la práctica en fases, las dependencias entre tareas y los plazos estimados para cada una. Este documento ha servido como herramienta de seguimiento y coordinación entre los integrantes del equipo.

PRÁCTICA 2: GRAMÁTICAS Y GENERADORES AUTOMÁTICOS

A continuación, se muestra el estado inicial del diagrama de Gantt, creado al inicio del desarrollo de la práctica.



Durante el desarrollo, se realizaron ajustes en los plazos según la complejidad de cada tarea. A continuación, se muestra el estado final del diagrama, el cual refleja el avance real y las desviaciones respecto a la planificación inicial.



Este contraste permite extraer conclusiones sobre la estimación temporal y el reparto de carga de trabajo, identificando las fases que requirieron más esfuerzo del previsto y

## PRÁCTICA 2: GRAMÁTICAS Y GENERADORES AUTOMÁTICOS

aquellas que se completaron de manera más eficiente. Las conclusiones extraídas son las siguientes:

- Hemos sobrestimado, ligeramente, nuestra capacidad de anticipar trabajo, lo cual ha resultado en retrasos en prácticamente todas las tareas.
- El reparto inicial no se ha respetado, lo cual no es necesariamente negativo, ya que éste se había hecho sin consultar a todos los miembros del equipo.
- El hecho de haber tenido exámenes los días 27 de octubre, 4 de noviembre y 12 de noviembre, junto con varios otros trabajos, ha visto muy mermada nuestra capacidad de llevarlo al día.

La planificación visual mediante el diagrama de Gantt ha resultado especialmente útil para controlar el progreso del proyecto en un plazo limitado, ofreciendo una visión global del estado de completitud en todo momento.

A pesar de los desajustes entre la planificación inicial y la final, el trabajo se ha entregado en tiempo y forma.

# 1. Primera parte: Generación de árboles sintácticos para lenguajes específicos

## 1.1. Árbol sintáctico de un CSV

En esta parte se desarrolla una gramática para reconocer un texto en formato CSV. El objetivo principal es detectar e identificar los elementos del lenguaje; es decir, las cabeceras de las columnas, las filas de datos y, por último, los campos separados por distintos delimitadores (';',',',|'). Una vez realizado esto se procederá a crear un árbol sintáctico legible para el usuario.

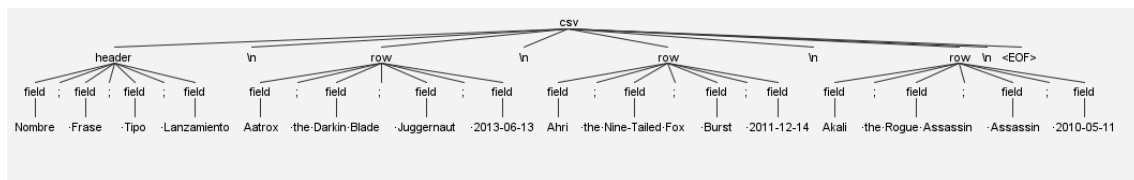
### 1.1.1. Nivel básico

Para ello, se define un lexer (CSVLexer.g4) encargado de identificar los tokens básicos como los separadores, texto, saltos de línea. También, se define un parser (CSVParser.g4) que organiza esos tokens jerárquicamente en un árbol sintáctico (AST). Con estos dos archivos se generan las clases Java automáticamente con ANTLR con ayuda del comando “antlr CSVParser.g4”. A continuación, se tendrán que compilar todas esas clases para poder trabajar con ellas realizando “javac \*.java”.

El fichero se analiza con la herramienta ANTLR y se genera un árbol donde se observa la estructura interna del documento CSV: una cabecera seguida por varias filas, y cada una formada por campos.

Finalmente, el árbol se visualiza mediante la herramienta TestRig (grun) de ANTLR, que permite comprobar gráficamente su correcta descomposición en nodos. De esta manera se demuestra que la gramática reconoce correctamente el archivo CSV, independientemente del separador que se utilice.

Para el archivo de ejemplo “CSV\_01.txt” ejecutando el comando “grun CSV csv -gui CSV\_01.txt” se visualiza esto:



### 1.1.2. Nivel medio

En el nivel medio se amplía lo trabajado anteriormente generando un AST legible por el usuario. EL objetivo ya no es solo construir el árbol interno de ANTLR, sino representarlo de forma clara en texto plano, mostrando únicamente los elementos esenciales.

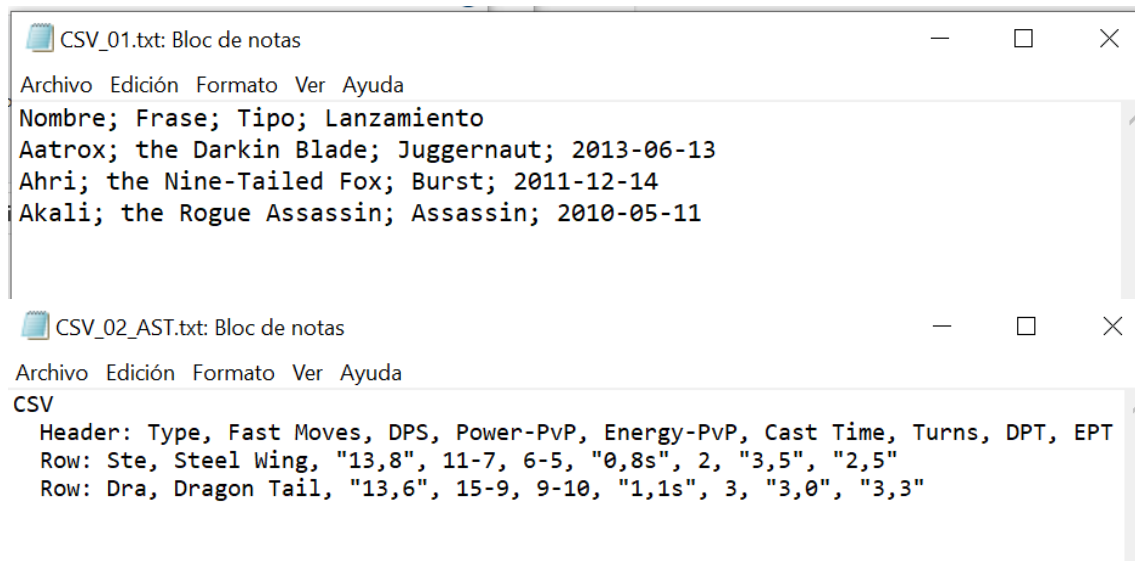
Para ello, se implementa un programa auxiliar en java que se ejecuta con la instrucción ‘java CSVPrinter’ y realiza los siguientes pasos:

1. Lee el archivo CSV
2. Ejecuta el lexer y el parser generados por ANTLR
3. Recorre el árbol sintáctico resultante



4. Guarda en un fichero .txt un resumen limpio del contenido.

El resultado final es un archivo de texto que representa el AST simplificado del CSV, demostrando la correcta comprensión de la estructura del lenguaje y el uso práctico del ANTLR para el análisis sintáctico.



## 1.2. Lenguaje de programación mínimo

En esta parte se realizará prácticamente lo mismo que en el apartado anterior pero esta vez con un lenguaje de programación elemental. Además se incluirán algunas ampliaciones para luego crear el AST legible para el usuario.

### 1.2.1. Nivel básico

El objetivo en este nivel es reconocer las insrucciones fundamentales y construir su árbol sintáctico mediante una gramática ANTLR.

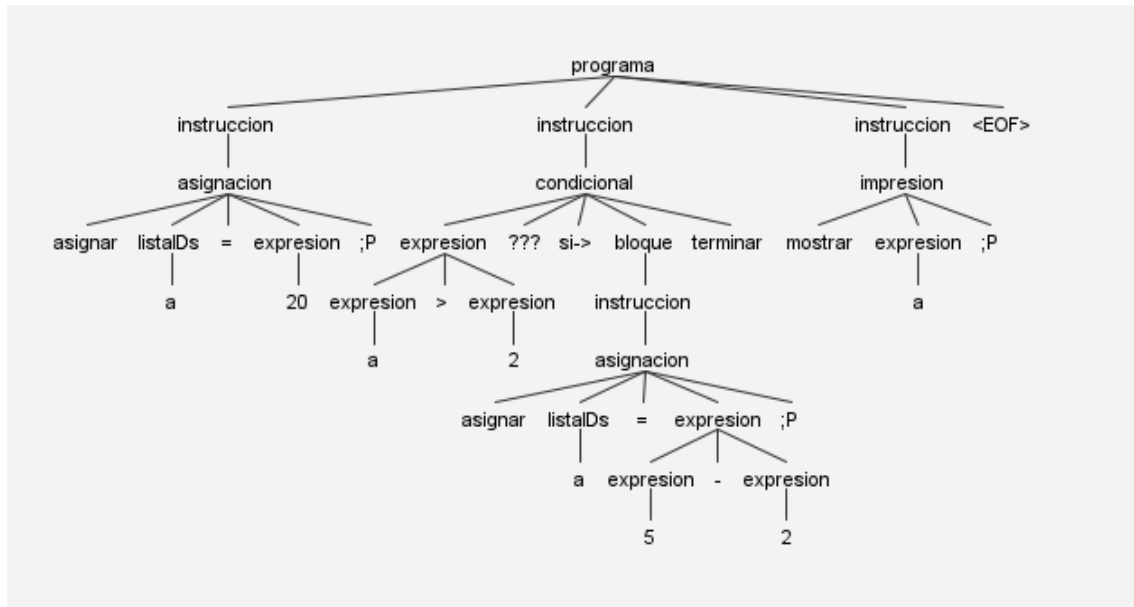
El lenguaje incluye:

- Asignaciones: mediante la palara 'asignar'
- Condicionales: con operadores como '???' 'si->' 'no->' y 'terminar'.
- Instrucciones de salida: mediante la palabra 'mostrar'.
- Operaciones aritméticas y comparativas: '+', '-', '<', '>', '==', etc.
- Comentarios con '#', líneas vacías y final de instrucción con ';P'.

Se crea el lexer (ELexer.g4) y parser (EParser.g4) del lenguaje, definiendo los tokens y las reglas sintácticas necesarias para generar el árbol sintáctico (AST) con ANTLR.

Mediante la ejecución del 'grun' se obtiene una representación gráfica del árbol, en el que se distingue claramente la estructura de cada instrucción del programa: asignación, condiciones y operaciones de salida.

Se puede verificar que el lenguaje puede analizarse correctamente y generar su arbol sintáctico completo:



### 1.2.2. Nivel medio

A continuación, se ampliará la funcionalidad del lenguaje E++ y se añadirá un componente adicional que genere el AST de forma legible.

Ampliaciones de la gramática:

- **Multipliación y división** ('\*' y '/').
- **Bucle** (mientras-> ... finmientras).
- **Asignación múltiple** (asignar a,b,c=0 ;P).
- **Paréntesis anidados** ('((a + b) - c') ).

Estas ampliaciones se implementan directamente en la gramática (tanto en el lexer como en el parser), manteniendo la compatibilidad con el nivel básico y mejorando la expresividad del lenguaje.

Además, se desarrolla un programa en Java que recorre el árbol sintáctico generado y genera un archivo .txt con el AST en formato legible para el usuario.

El archivo resultante (por ejemplo, AST\_prueba.txt) contiene un resumen textual de la estructura del programa, mostrando únicamente las partes esenciales:

- Asignación con variables y expresiones
- Condicionales con sus ramas 'si' y 'no'.
- Bucles 'mientras'
- Instrucciones 'mostrar'

De esta manera, se consigue una salida parecida a esta:

```
Programa E (prueba.txt)
  Asignar: a,b,c = 0
  Mientras (a<3):
    mostrara;P
    asignara=a+1;P
  FinMientras
  Si ((a+(b-c))>=2):
    mostrar"OK";P
  No:
    mostrar"FAIL";P
  Terminar
```

El programa se adapta automáticamente al archivo de entrada, de modo que puede analizar cualquier fichero .txt, generando siempre su correspondiente AST en texto plano, por lo que se cumple el objetivo del apartado.

## 2. Segunda parte: Traducción e interpretación de manuales de IKEA

### 2.1. Introducción y planteamiento

La segunda parte de la práctica plantea el diseño completo de un lenguaje formal capaz de describir, de manera sistemática y rigurosa, los pasos necesarios para montar distintas librerías y estanterías de IKEA a partir de sus manuales gráficos. A diferencia de la primera parte, donde el lenguaje estaba establecido, en esta fase debimos nosotros definir el lenguaje desde cero, implementar su análisis léxico y sintáctico, generar el AST correspondiente y, finalmente, desarrollar un analizador semántico funcional que procese y acumule información del montaje. Se han tratado de seguir las recomendaciones indicadas por el profesor el día de presentación de la práctica: en resumen, tratar de conseguir un balance entre tener un lenguaje simple pero que abarque suficiente información.

La problemática presenta varios retos relevantes. Los manuales de IKEA se caracterizan por ser puramente visuales, carentes de texto y, por tanto, ambiguos en cuanto a terminología o estructura. Además, cada manual utiliza piezas, cantidades y operaciones diferentes, lo que hace imprescindible diseñar un lenguaje flexible pero suficientemente restrictivo como para evitar ambigüedades durante el análisis sintáctico. Todo ello nos ha obligado a tomar ciertas decisiones de diseño que se explicarán a más adelante.

El planteamiento seguido se basó en los siguientes principios:

- **Sistematicidad:** todas las instrucciones del lenguaje debían seguir una estructura uniforme y fácilmente parseable.
- **Generalización:** minimizar el número de verbos, piezas y operadores, de forma que distintas instrucciones de los manuales pudieran describirse mediante los mismos constructos.
- **Modularidad:** separar claramente los componentes del lenguaje (verbos, herramientas, herrajes, piezas, orientaciones) para facilitar el análisis y la extensibilidad.
- **Simplicidad léxica:** evitar la explosión de variantes textuales (por ejemplo, dirigiendo la manipulación de herrajes a través de códigos numéricos únicos y no mediante nombres propios).
- **Correspondencia directa con la semántica real:** cada instrucción del lenguaje debía reflejar de manera explícita acciones que realmente se ejecutan durante el montaje.
- **Procesabilidad:** el lenguaje debía ser lo suficientemente formal como para permitir la generación automática del AST y su posterior análisis semántico.

Para lograr estas metas, se (1) realizó un análisis de los manuales, extrayendo los elementos comunes entre ellos y las acciones esenciales del proceso de montaje. A partir de dicho estudio se (2) elaboró un conjunto reducido pero expresivo de verbos de acción (UNIR, ATORNILLAR, PONER, GIRAR, REPETIR), un sistema normalizado de piezas (PIEZA BASE\_n, PIEZA LATERAL\_n, PIEZA SUPERIOR), y un catálogo unificado de herrajes (ESPIGA, TORNILLO, TUERCA, HERRAJE) asociado a sus respectivos códigos.

Con el lenguaje ya definido, se procedió a traducir manualmente cada uno de los muebles —manteniendo fidelidad a la estructura del montaje original— y a implementar el lexer, parser y AST builder que permitieran reconocer y manipular esas traducciones. Finalmente, se desarrolló un analizador que recorre el AST para obtener información agregada, como la lista de herrajes necesarios y las herramientas empleadas, reproduciendo así un proceso similar al de una tabla de símbolos en un compilador real.

Cabe destacar que la extracción de los elementos del lenguaje (verbos de acción, piezas y herrajes) no se realizó de manera definitiva en la primera iteración, sino que fue fruto de las sucesivas traducciones de los diferentes manuales.

Esta sección documenta de manera detallada cada decisión adoptada en el diseño del lenguaje, el proceso de traducción de los manuales y la implementación de los distintos componentes del analizador.

## 2.2. Nivel básico

### 2.2.1. Definición del lenguaje

Esta parte trata acerca del diseño del lenguaje capaz de describir cualquier manual de montaje de IKEA de manera sistemática, implementando un analizador léxico y sintáctico que pueda reconocer dicho lenguaje y construir su correspondiente AST.

Este proceso no fue lineal: el lenguaje evolucionó de forma iterativa, ajustándose progresivamente tras analizar inconsistencias, ambigüedades o necesidades que iban apareciendo a medida que se traducían los manuales.

A continuación se describen estas iteraciones y las decisiones de diseño más relevantes.

#### 1. Lenguaje semánticamente mezclado.

La versión inicial del lenguaje incluía definiciones como

DESTORNILLADOR <ATORNILLAR TORNILLO <id>> MARTILLO <PONER ESPIGA <id> en PIEZA <tipo>>
---

Aunque intuitivamente parecía útil, ya que expresaba “qué hace cada herramienta”, se detectaron rápidamente dos problemas:

- Se mezcla vocabulario con sintaxis, rompiendo uno de los principios de diseño de los lenguajes formales, ya que las herramientas deben ser tokens, no plantillas de instrucciones.
- El parser no podía interpretar una herramienta que contenía dentro una expresión sintáctica ya que introduce reglas que no corresponden a un terminal.

Finalmente, se simplificó la sección herramientas a símbolos terminales.

#### 2. Verbo PONER sin herramienta.

Mientras se traducían los manuales, se observó que había dos casos en los que se requería introducir un herraje en una pieza: casos en los que era necesaria una herramienta (por fuerza o complejidad) y casos en los que no. Para los casos en los que no se precisaba de una herramienta (como colocar una tuerca o poner una espiga), se optó por añadir el verbo “PONER”. Esto, traducido a nuestro lenguaje, tenía la siguiente apariencia:

PONER 4 ESPIGA 101351 en PIEZA BASE\_1

Lo cual generaba dos problemas también:

- Contradicción de diseño: pues otras instrucciones ya habíaN adoptado el formado uniforme “Con <HERRAMIENTA>, ...”.
- Dificultad para distinguir en el AST si una herramienta había sido utilizada o no.

La solución final a la que se llegó fue introducir una herramienta adicional “MANO”. De esta forma, todas las instrucciones que implican manipular un elemento se expresan con el mismo patrón “Con <HERRAMIENTA>, ...”, mejorando la coherencia.



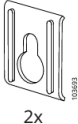
### 3. Utilizar sustantivos sin número.

Es decir, que se empleará “4 TORNILLO” en lugar de “4 TORNILLOS”. Esto mantiene el lenguaje compacto, simplifica la gramática, evita ambigüedades y evita dos tokens casi iguales en el lexer sin perder información relevante.

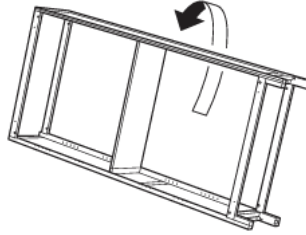
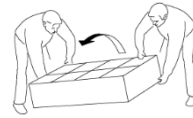
Por último, unos apuntes adicionales sobre el lenguaje:

- Cada paso se identifica por un número. Un paso puede estar dividido en subpasos (2.1, 2.2, etc.) Esto es útil para la acción “REPETIR <paso>”.
- Se ha tomado la decisión de omitir los pasos (son pocos) en los que se utilizan herrajes (tornillos, etc.) sin numerar. Un ejemplo son los pasos 18 y 19 del manual de montaje del mueble “Flysta”. Se podría haber asignado un identificador cualquiera a estos tornillos también.

Finalmente, se llegó a el siguiente diseño final del lenguaje:

- Verbos:
  - UNIR.
  - PONER.
  - ATORNILLAR.
  - GIRAR.
  - REPETIR.
- Piezas:
  - PIEZA BASE\_X. Corresponde con piezas que, usualmente, se colocan de manera horizontal, cuyo objetivo es servir de apoyo para objetos.
  - PIEZA LATERAL\_X. Corresponde con piezas que, usualmente, se colocan de manera vertical, cuyo objetivo es delimitar las dimensiones horizontales del mueble/estantería.
  - PIEZA SUPERIOR. Corresponde con la pieza (o piezas) que delimitan el mueble o estantería por arriba.
- Herrajes:
  - ESPIGA <id>. Un ejemplo de espiga es: 
  - TORNILLO <id>. Un ejemplo de tornillo es: 
  - TUERCA <id>.
  - HERRAJE <id>. Un ejemplo de herraje es: 
- Herramientas:
  - DESTORNILLADOR.
  - MARTILLO.

- MANO.
- Orientaciones:
  - ABAJO.
  - ARRIBA. Un ejemplo de girar hacia ARRIBA es:
  - LATERAL. Un ejemplo de girar LATERAL es:



### 2.2.2. Traducción de los manuales

A continuación se presenta la traducción de los manuales:

- Baggebo.

ITEM: baggebo

1. UNIR PIEZA SUPERIOR y PIEZA LATERAL\_1;  
Con DESTORNILLADOR, ATORNILLAR 2 TORNILLO 10119123 en PIEZA LATERAL\_1
2. UNIR PIEZA BASE y PIEZA LATERAL\_1;  
Con DESTORNILLADOR, ATORNILLAR 6 TORNILLO 10119123 en PIEZA LATERAL\_1
3. UNIR PIEZA LATERAL\_1 y PIEZA BASE;  
Con DESTORNILLADOR, ATORNILLAR 8 TORNILLO 10119123 en PIEZA BASE
4. UNIR PIEZA LATERAL\_2 y PIEZA BASE;  
Con DESTORNILLADOR, ATORNILLAR 4 TORNILLO 10119123 en PIEZA BASE
5. Con DESTORNILLADOR, ATORNILLAR 4 TORNILLO 10003068 en PIEZA LATERAL\_1
- 6.1. GIRAR ARRIBA
- 6.2. Con DESTORNILLADOR, ATORNILLAR 2 TORNILLO 10119123 en PIEZA SUPERIOR con TUERCA 100827
7. REPETIR 6.2

- Billy.

ITEM: billy

- 1.1. Con MANO, PONER 12 ESPIGA 101351 en PIEZA BASE\_1
- 1.2. Con MANO, PONER 4 ESPIGA 101351 en PIEZA BASE\_2
2. Con DESTORNILLADOR, ATORNILLAR 12 TORNILLO 118331 en PIEZA LATERAL\_1
- 3.1. UNIR PIEZA BASE\_1 y PIEZA LATERAL\_1
- 3.2. UNIR PIEZA BASE\_2 y PIEZA LATERAL\_1
4. Con DESTORNILLADOR, ATORNILLAR 4 TORNILLO 119081 en PIEZA BASE\_1
5. UNIR PIEZA BASE\_1 y PIEZA LATERAL\_1
6. REPETIR 5

7. Con DESTORNILLADOR, ATORNILLAR 6 TORNILLO 119081 en PIEZA LATERAL\_1  
9. UNIR PIEZA LATERAL\_2 y PIEZA LATERAL\_1  
11. Con DESTORNILLADOR, ATORNILLAR 18 TORNILLO 101201 en PIEZA LATERAL\_2  
12.1. GIRAR ARIIBA  
12.2. Con DESTORNILLADOR, ATORNILLAR 2 TORNILLO 100823 en PIEZA BASE\_1  
12.3. Con DESTORNILLADOR, ATORNILLAR 2 TORNILLO 109041 en PIEZA BASE\_1 con TUERCA 100823  
13. Con MANO, PONER 16 HERRAJE 131372 en PIEZA LATERAL\_1

- Flysta.

ITEM:flysta

1. UNIR PIEZA BASE\_1 y PIEZA LATERAL\_1;  
Con DESTORNILLADOR, ATORNILLAR 2 TORNILLO 104321 en PIEZA LATERAL\_1  
2. UNIR PIEZA BASE\_1 y PIEZA LATERAL\_2;  
Con MARTILLO, PONER 2 ESPIGA 101345 en PIEZA BASE\_1  
3. UNIR PIEZA LATERAL\_2 y PIEZA BASE\_2;  
Con MARTILLO, PONER 2 ESPIGA 101345 en PIEZA BASE\_2  
4. Con MARTILLO, PONER 2 ESPIGA 101339 en PIEZA BASE\_2  
5. UNIR PIEZA LATERAL\_3 y PIEZA BASE\_2  
6. UNIR PIEZA LATERAL\_1 y PIEZA BASE\_3;  
Con MARTILLO, PONER 2 ESPIGA 101345 en PIEZA BASE\_3  
7. Con MARTILLO, PONER 2 ESPIGA 101339 en PIEZA BASE\_3  
8. UNIR PIEZA LATERAL\_4 y PIEZA BASE\_3  
9. UNIR PIEZA LATERAL\_1 y PIEZA BASE\_4;  
Con MARTILLO, PONER 2 ESPIGA 101345 en PIEZA BASE\_4  
10. Con MARTILLO, PONER 2 ESPIGA 101339 en PIEZA BASE\_4  
11. UNIR PIEZA LATERAL\_5 y PIEZA BASE\_4  
12.1. Con MARTILLO, PONER 2 ESPIGA 101345 en PIEZA BASE\_2  
12.2. Con MARTILLO, PONER 2 ESPIGA 101345 en PIEZA BASE\_3  
12.3. Con MARTILLO, PONER 2 ESPIGA 101345 en PIEZA BASE\_4  
12.4. Con MARTILLO, PONER 2 ESPIGA 101345 en PIEZA LATERAL\_5  
13. UNIR PIEZA LATERAL\_6 y PIEZA BASE\_1  
14. UNIR PIEZA SUPERIOR y PIEZA LATERAL\_6;  
Con DESTORNILLADOR, ATORNILLAR 6 TORNILLO 104321 en PIEZA SUPERIOR  
15. Con MANO, PONER 8 HERRAJE 153550 en PIEZA LATERAL\_6  
16. Con DESTORNILLADOR, ATORNILLAR 2 TORNILLO 100372 en PIEZA LATERAL\_1  
19. Con MANO, PONER 2 HERRAJE 103693 en PIEZA LATERAL\_1

- Hemnes.

ITEM:hemnes

1. Con DESTORNILLADOR, ATORNILLAR 8 TORNILLO 118331 en PIEZA LATERAL\_1  
2. Con MANO, PONER 6 ESPIGA 101350 en PIEZA LATERAL\_2



3. Con DESTORNILLADOR, ATORNILLAR 2 TORNILLO 100212 en PIEZA LATERAL\_2
4. Con DESTORNILLADOR, ATORNILLAR 2 TORNILLO 100211 en PIEZA LATERAL\_1
5. Con MANO, PONER 4 HERRAJE 102728 en PIEZA LATERAL\_1
6. UNIR PIEZA LATERAL\_3 y PIEZA LATERAL\_1;  
Con MANO, PONER 2 ESPIGA 101350 en PIEZA LATERAL\_3
7. REPETIR 6
8. REPETIR 6
9. Con MANO, PONER 4 HERRAJE 113434 en PIEZA LATERAL\_3
10. Con DESTORNILLADOR, ATORNILLAR 4 TORNILLO 118331 en PIEZA LATERAL\_2
11. UNIR PIEZA LATERAL\_4 y PIEZA LATERAL\_2;  
Con DESTORNILLADOR, ATORNILLAR 2 TORNILLO 113434 en PIEZA LATERAL\_4
12. UNIR PIEZA LATERAL\_5 y PIEZA LATERAL\_1;  
Con MANO, PONER 3 ESPIGA 101350 en PIEZA LATERAL\_5
13. Con MANO, PONER 6 HERRAJE 113434 en PIEZA LATERAL\_5
- 14.1. GIRAR LATERAL
- 14.2. UNIR PIEZA SUPERIOR y PIEZA LATERAL\_1
- 14.3. Con MANO, PONER 2 ESPIGA 101350 en PIEZA SUPERIOR
15. Con DESTORNILLADOR, ATORNILLAR 7 TORNILLO 100211 en PIEZA SUPERIOR
- 16.1. GIRAR LATERAL
- 16.2. UNIR PIEZA LATERAL\_6 y PIEZA LATERAL\_7
17. REPETIR 16.2
18. REPETIR 16.2
19. REPETIR 16.2
- 20.1. GIRAR LATERAL
- 20.2. Con DESTORNILLADOR, ATORNILLAR 3 TORNILLO 109560 en PIEZA LATERAL\_6
21. REPETIR 20.2
22. GIRAR ARRIBA
23. Con DESTORNILLADOR, ATORNILLAR 2 TORNILLO 109560 en PIEZA SUPERIOR con TUERCA 100823
24. Con MANO, PONER 4 HERRAJE 104171 en PIEZA LATERAL\_1
25. UNIR PIEZA BASE y PIEZA LATERAL\_1
26. Con MANO, PONER 8 HERRAJE 104171 en PIEZA LATERAL\_1

- Kallax.

ITEM:kallax

1. UNIR PIEZA LATERAL\_1 y PIEZA BASE\_1;  
Con DESTORNILLADOR, ATORNILLAR 2 TORNILLO 104321 en PIEZA BASE\_1
2. UNIR PIEZA LATERAL\_2 y PIEZA BASE\_1;  
Con MARTILLO, PONER 1 ESPIGA 101339 en PIEZA LATERAL\_2
3. UNIR PIEZA BASE\_2 y PIEZA LATERAL\_1;  
Con MARTILLO, PONER 2 ESPIGA 101339 en PIEZA BASE\_2
4. REPETIR 3
5. UNIR PIEZA LATERAL\_3 y PIEZA BASE\_2
6. UNIR PIEZA BASE\_3 y PIEZA LATERAL\_1;  
Con MARTILLO, PONER 2 ESPIGA 101339 en PIEZA BASE\_3

7. Con MARTILLO, PONER 2 ESPIGA 101339 en PIEZA BASE\_3  
8. UNIR PIEZA LATERAL\_4 y PIEZA BASE\_3  
9. UNIR PIEZA BASE\_4 y PIEZA LATERAL\_1;  
    Con MARTILLO, PONER 2 ESPIGA 101339 en PIEZA BASE\_4  
10. Con MARTILLO, PONER 2 ESPIGA 101339 en PIEZA BASE\_4  
11. UNIR PIEZA LATERAL\_5 Y PIEZA BASE\_4  
12. Con MARTILLO, PONER 8 ESPIGA 101339 en PIEZA BASE  
13. UNIR PIEZA LATERAL\_6 Y PIEZA BASE  
14. UNIR PIEZA SUPERIOR y PIEZA LATERAL\_1  
15. Con DESTORNILLADOR, ATORNILLAR 6 TORNILLO 104322 en PIEZA SUPERIOR  
16. Con MANO, PONER 2 HERRAJE 100372 en PIEZA SUPERIOR  
19. Con MANO, PONER 2 HERRAJE 115754 en PIEZA LATERAL\_1

- Knagglig.

ITEM:knagglig

1. UNIR PIEZA LATERAL\_1 y PIEZA BASE\_1  
2. Con DESTORNILLADOR, ATORNILLAR 2 TORNILLO 108363 en PIEZA LATERAL\_1  
3.1. GIRAR ARRIBA  
3.2. UNIR PIEZA LATERAL\_2 y PIEZA LATERAL\_1  
4. Con DESTORNILLADOR, ATORNILLAR 2 TORNILLO 108363 en PIEZA LATERAL\_1  
5. UNIR PIEZA LATERAL\_3 y PIEZA BASE\_1  
6. Con DESTORNILLADOR, ATORNILLAR 4 TORNILLO 108560 en PIEZA LATERAL\_3  
7. REPETIR 6  
8. REPETIR 7

- Lack.

ITEM:lack

1.1. UNIR PIEZA LATERAL\_1 y PIEZA BASE  
1.2. Con DESTORNILLADOR, ATORNILLAR 12 TORNILLO 104323 en PIEZA BASE  
2. Con DESTORNILLADOR, ATORNILLAR 2 TORNILLO 104325 en PIEZA BASE

- Smagora.

ITEM:smagora

1. Con DESTORNILLADOR, ATORNILLAR 16 TORNILLO 118331 en PIEZA LATERAL\_1  
2.1. UNIR PIEZA SUPERIOR y PIEZA LATERAL\_1;  
    Con MANO, PONER 4 ESPIGA 101345 en PIEZA SUPERIOR  
2.2. UNIR PIEZA BASE\_1 y PIEZA LATERAL\_1;  
    Con MANO, PONER 4 ESPIGA 101345 en PIEZA BASE\_1  
3.1. UNIR PIEZA BASE\_2 y PIEZA LATERAL\_1;  
    Con MANO, PONER 4 ESPIGA 101345 en PIEZA BASE\_2  
3.2. UNIR PIEZA BASE\_3 y PIEZA LATERAL\_1;

```
Con MANO, PONER 4 ESPIGA 101345 en PIEZA BASE_3
4. Con MANO, PONER 8 HERRAJE 119252 en PIEZA BASE
5. UNIR PIEZA LATERAL_2 y PIEZA LATERAL_1
6. GIRAR LATERAL
7. Con MANO, PONER 8 HERRAJE 119252 en PIEZA BASE
8. Con MANO, PONER 2 HERRAJE 107716 en PIEZA LATERAL_1
9. Con MANO, PONER 2 HERRAJE 118331 en PIEZA BASE_1
10. Con MANO, PONER 2 ESPIGA 101345 en PIEZA LATERAL_3
11. UNIR PIEZA LATERAL_3 y PIEZA SUPERIOR
12. Con MANO, PONER 2 HERRAJE 119252 en PIEZA LATERAL_3
```

- Vesken:

```
ITEM:vesken

1.1. Con MANO, PONER 4 HERRAJE
1.2. UNIR PIEZA LATERAL_1 y PIEZA BASE 1
2. Con DESTORNILLADOR, ATORNILLAR 2 TORNILLO 108363 en PIEZA
LATERAL_1
3.1. GIRAR ARRIBA
3.2. Unir PIEZA LATERAL 2 y PIEZA LATERAL_1
4. Con DESTORNILLADOR, ATORNILLAR 2 TORNILLO 108363 en PIEZA
LATERAL_1
5. UNIR PIEZA LATERAL_3 y PIEZA BASE_1
6. Con DESTORNILLADOR, ATORNILLAR 4 TORNILLO 108560 en PIEZA
LATERAL_3
7. REPETIR 6
8. REPETIR 7
```

### 2.2.3. Creación del Lexer y el Parser

El lexer fue diseñado con tres objetivos:

- Ser sensible a mayúsculas y minúsculas. Los manuales escritos por nosotros podían tener distintas formas de escribir una misma palabra: UNIR, unir, UniR, etc. Por ello, se definieron tokens con expresiones regulares explícitas:

```
UNIR : [Uu][Nn][Ii][Rr];
```

- Reconocer estructuras típicas de los manuales. Como “PIEZA LATERAL\_1”.
- Limitar identificadores. Siempre en mayúsculas y con guion bajo.

El lexer es deliberadamente estricto para evitar ambigüedades y garantizar que cualquier error en la traducción se detecte y corrija inmediatamente. Esta tarea se ve facilitada por los **claros mensajes de error generados**, como se muestra en la siguiente captura:

```

traducciones > baggebo.txt
1  ITEM:baggebo
2
3
4  1. UNIR PIEZA SUPERIOR y PIEZA LATERAL_1;
5    Con DESTORNILLADOR, ATORNILLAR 2 TORNILLO 10119123 en PIEZA LATERAL_1
6  2. UNIR PIEZA BASE y PIEZA LATERAL_1;
7    Con DESTORNILLADOR, ATORNILLAR 6 TORNILLO 10119123 en PIEZA LATERAL_1
8  3. UNIR PIEZA LATERAL_1 y PIEZA BASE;
9    Con DESTORNILLADOR, ATORNILLAR 8 TORNILLOS 10119123 en PIEZA BASE
10 4. UNIR PIEZA LATERAL_2 y PIEZA BASE;
11   Con DESTORNILLADOR, ATORNILLAR 4 TORNILLO 10119123 en PIEZA BASE
12 5. Con DESTORNILLADOR, ATORNILLAR 4 TORNILLO 10003068 en PIEZA LATERAL_1
13 6.1. GIRAR ARRIBA
14 6.2. Con DESTORNILLADOR, ATORNILLAR 2 TORNILLO 10119123 en PIEZA SUPERIOR con TUERCA 100827
15 7. REPETIR 6.2

```

GIT LENS POSTGRES SQL QUERY RESULTS SQLTOOLS PROBLEMS 86 TERMINAL CHAT DEBUG CONSOLE OUTPUT PORTS

PS C:\Users\diego\DIEGO\UAH\Curso 2025-2026\1 Cuatrimestre\PDL\Laboratorio\uah\_procesadores\PL2- Gramáticas complete.jar" IkeaMain ..\traducciones\baggebo.txt  
line 9:37 mismatched input 'TORNILLOS' expecting TORNILLO

Por otro lado, el parser se diseñó siguiendo un enfoque **descendente predictivo LL(1)** para garantizar:

- Reglas claras.
- No ambigüedad.
- Facilidad de construcción del AST.

Las decisiones clave en su desarrollo fueron:

- Separación entre instrucción y componentes internos. Cada verbo tiene su propia regla:

```

unirInstr
: UNIR PIEZA IDENT Y PIEZA IDENT
;

ponerInstr
: CON herramienta COMMA PONER INT tipoHerraje INT EN PIEZA IDENT
;

conHerramientaAtornillarInstr
: CON herramienta COMMA atornillarInstr
;

atornillarInstr
: ATORNILLAR INT TORNILLO INT
  (EN PIEZA IDENT)?
  (CON TUERCA INT)?
;

girarInstr
: GIRAR orientacion
;

repetirInstr
: REPETIR stepLabel
;

```

- Soporte para subpasos:

```

step
: stepLabel DOT instructionList
;

stepLabel
: INT (DOT INT)*
;
    
```

- El encabezado “ITEM” es opcional. Esto permite generar las instrucciones para un mueble cuyo nombre es desconocido.

A continuación se muestran los resultados de la ejecución del código Main, el cual, entre otras cosas, comprueba que la traducción de un manual es aceptada por nuestro lenguaje:

```

> java -cp ".;generated;..\lib\antlr-4.13.2-complete.jar" IkeaMain ..\traducciones\baggebo.txt
Parseo correcto.

AST construido correctamente.
ITEM: null
Número de pasos: 8
Árbol guardado en: C:\Users\diego\DIEGO\UAH\Curso 2025-2026\1 Cuatrimestre\PDL\Laboratorio\uah_procesadores\PL2- Gramáticas y Generadores Automáti
cos\src\ikea\src\parse_tree_html\null_parse_tree.html
Árbol de parseo generado en: parse_tree_html\null_parse_tree.html
AST HTML guardado en: C:\Users\diego\DIEGO\UAH\Curso 2025-2026\1 Cuatrimestre\PDL\Laboratorio\uah_procesadores\PL2- Gramáticas y Generadores Autom
áticos\src\ikea\src\ast_html\null_ast.html
AST visual generado en: ast_html\null_ast.html
    
```

#### 2.2.4. Construcción del AST

Una vez definido el lenguaje y generados el lexer y el parser, el siguiente paso consistió en construir un Árbol de Sintaxis Abstracta (AST) que representase de manera estructurada la semántica del manual de montaje.

A diferencia del árbol de análisis sintáctico (parse tree), que refleja fielmente todas las reglas de la gramática, el AST debe capturar únicamente los elementos conceptuales del montaje: pasos, instrucciones, piezas, cantidades, herramientas, etc.

- Diseño de la jerarquía de nodos:  
El diseño del AST se orientó hacia la simplicidad y extensibilidad, definiendo una jerarquía clara:
  - AstNode como superclase abstracta.
  - InstructionNode como clase base para todas las instrucciones.
  - Varias subclases especializadas, como UnirInstrNode, PonerInstrNode o ConHerramientaAtornillarInstrNode.
- Uso del patrón Visitor:  
Para construir el AST se utilizó el patrón Visitor, que ANTLR soporta generando automáticamente la clase IkeaParserBaseVisitor. Implementamos una subclase llamada IkeaAstBuilder, que:
  1. Visita cada nodo relevante del parse tree.
  2. Extrae los valores semánticos (números, identificadores, tipos).
  3. Construye instancias de las clases AST.
  4. Las enlaza formando una estructura completa y coherente.
- Validación del AST.  
Para garantizar que el AST construido era correcto, se realizaron dos tipos de validación:
 

dsd

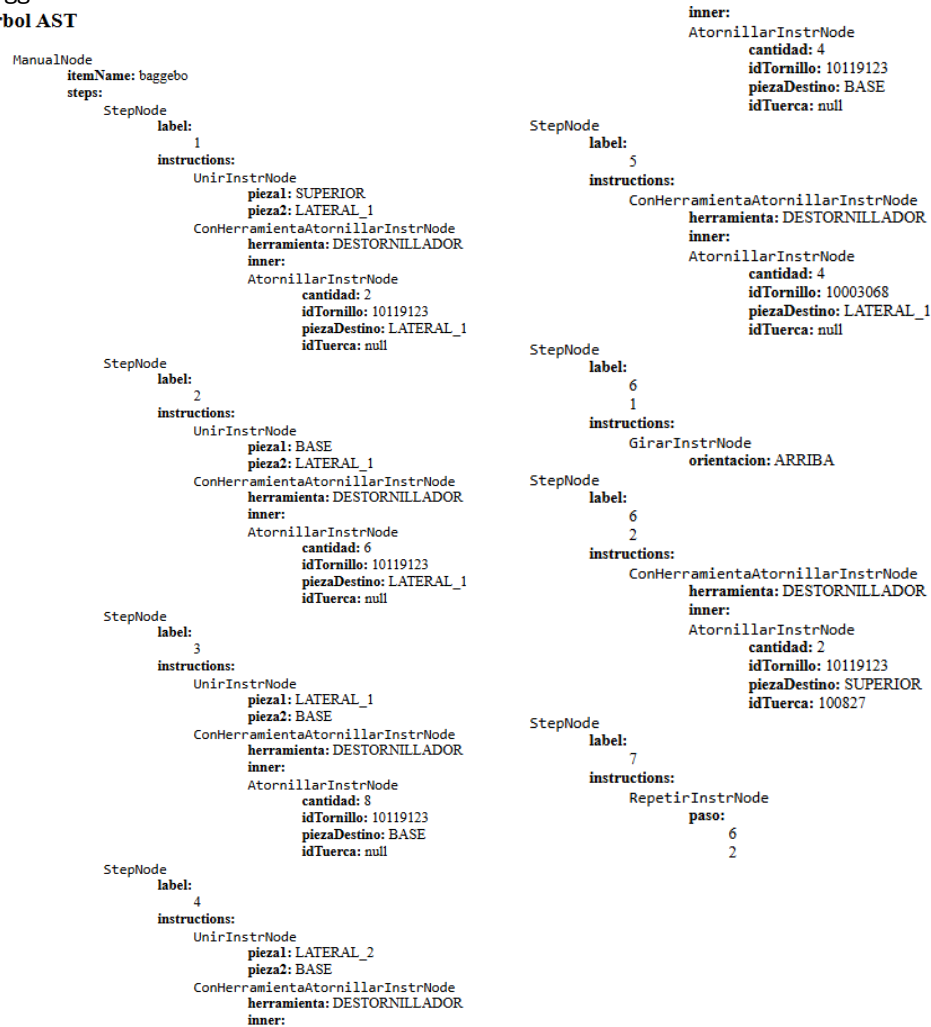
## PRÁCTICA 2: GRAMÁTICAS Y GENERADORES AUTOMÁTICOS

1. Pruebas automáticas: al ejecutar el parser sobre cada traducción se verifica que no se producen errores de recorrido, todos los pasos están representados y los nodos reflejan fielmente la traducción textual.
2. Generación visual: se desarrolló un generador de HTML que muestra de forma expandida el tree, facilitando la comprobación manual del resultado.

A continuación, se muestra el AST para la traducción de cada uno de los manuales:

- **Baggebo:**

**Árbol AST**



- **Billy:**

## Árbol AST

```
ManualNode
  itemName: billy
  steps:
    StepNode
      label: 1
      1
      instructions:
        PonerInstrNode
          cantidad: 12
          tipoHerraje: ESPIGA
          idHerraje: 101351
          piezaDestino: BASE_1
          herramienta: MANO
    StepNode
      label: 1
      2
      instructions:
        PonerInstrNode
          cantidad: 4
          tipoHerraje: ESPIGA
          idHerraje: 101351
          piezaDestino: BASE_2
          herramienta: MANO
    StepNode
      label: 2
      instructions:
        ConHerramientaAtornillarInstrNode
          herramienta: DESTORNILLADOR
          inner:
            AtornillarInstrNode
              cantidad: 12
              idTornillo: 118331
              piezaDestino: LATERAL_1
              idTuerca: null
    StepNode
      label: 3
      1
      instructions:
        UnirInstrNode
          pieza1: BASE_1
          pieza2: LATERAL_1
    StepNode
      label: 3
      2
      instructions:
        UnirInstrNode
          pieza1: BASE_2
          pieza2: LATERAL_1
    StepNode
      label: 4
      instructions:
        ConHerramientaAtornillarInstrNode
```

```
ConHerramientaAtornillarInstrNode
  herramienta: DESTORNILLADOR
  inner:
    AtornillarInstrNode
      cantidad: 4
      idTornillo: 119081
      piezaDestino: BASE_1
      idTuerca: null
StepNode
  label: 5
  instructions:
    UnirInstrNode
      pieza1: BASE_1
      pieza2: LATERAL_1
StepNode
  label: 6
  instructions:
    RepetirInstrNode
      paso: 5
StepNode
  label: 7
  instructions:
    ConHerramientaAtornillarInstrNode
      herramienta: DESTORNILLADOR
      inner:
        AtornillarInstrNode
          cantidad: 6
          idTornillo: 119081
          piezaDestino: LATERAL_1
          idTuerca: null
StepNode
  label: 9
  instructions:
    UnirInstrNode
      pieza1: LATERAL_2
      pieza2: LATERAL_1
StepNode
  label: 11
  instructions:
    ConHerramientaAtornillarInstrNode
      herramienta: DESTORNILLADOR
      inner:
        AtornillarInstrNode
          cantidad: 18
          idTornillo: 101201
          piezaDestino: LATERAL_2
          idTuerca: null
StepNode
  label: 12
  1
  instructions:
    GirarInstrNode
      orientacion: ARRIBA
```

## PRÁCTICA 2: GRAMÁTICAS Y GENERADORES AUTOMÁTICOS

```
StepNode
  label:
    12
    2
  instructions:
    ConHerramientaAtornillarInstrNode
      herramienta: DESTORNILLADOR
      inner:
        AtornillarInstrNode
          cantidad: 2
          idTornillo: 100823
          piezaDestino: BASE_1
          idTuerca: null
```

```
StepNode
  label:
    12
    3
  instructions:
    ConHerramientaAtornillarInstrNode
      herramienta: DESTORNILLADOR
      inner:
        AtornillarInstrNode
          cantidad: 2
          idTornillo: 109041
          piezaDestino: BASE_1
          idTuerca: 100823
```

```
StepNode
  label:
    13
  instructions:
    PonerInstrNode
      cantidad: 16
      tipoHerraje: HERRAJE
      idHerraje: 131372
      piezaDestino: LATERAL_1
      herramienta: MANO
```

- Flysta:



### Árbol AST

```
ManualNode
  itemName: flysta
  steps:
    StepNode
      label: 1
      instructions:
        UnirInstrNode
          pieza1: BASE_1
          pieza2: LATERAL_1
        ConHerramientaAtornillarInstrNode
          herramienta: DESTORNILLADOR
          inner:
            AtornillarInstrNode
              cantidad: 2
              idTornillo: 104321
              piezaDestino: LATERAL_1
              idTuerca: null
    StepNode
      label: 2
      instructions:
        UnirInstrNode
          pieza1: BASE_1
          pieza2: LATERAL_2
        PonerInstrNode
          cantidad: 2
          tipoHerraje: ESPIGA
          idHerraje: 101345
          piezaDestino: BASE_1
          herramienta: MARTILLO
    StepNode
      label: 3
      instructions:
        UnirInstrNode
          pieza1: LATERAL_2
          pieza2: BASE_2
        PonerInstrNode
          cantidad: 2
          tipoHerraje: ESPIGA
          idHerraje: 101345
          piezaDestino: BASE_2
          herramienta: MARTILLO
    StepNode
      label: 4
      instructions:
        PonerInstrNode
          cantidad: 2
          tipoHerraje: ESPIGA
          idHerraje: 101339
          piezaDestino: BASE_2
          herramienta: MARTILLO
    StepNode
      label: 5
      instructions:
```

```
instructions:
  UnirInstrNode
    pieza1: LATERAL_3
    pieza2: BASE_2
StepNode
  label: 6
  instructions:
    UnirInstrNode
      pieza1: LATERAL_1
      pieza2: BASE_3
    PonerInstrNode
      cantidad: 2
      tipoHerraje: ESPIGA
      idHerraje: 101345
      piezaDestino: BASE_3
      herramienta: MARTILLO
StepNode
  label: 7
  instructions:
    PonerInstrNode
      cantidad: 2
      tipoHerraje: ESPIGA
      idHerraje: 101339
      piezaDestino: BASE_3
      herramienta: MARTILLO
StepNode
  label: 8
  instructions:
    UnirInstrNode
      pieza1: LATERAL_4
      pieza2: BASE_3
StepNode
  label: 9
  instructions:
    UnirInstrNode
      pieza1: LATERAL_1
      pieza2: BASE_4
    PonerInstrNode
      cantidad: 2
      tipoHerraje: ESPIGA
      idHerraje: 101345
      piezaDestino: BASE_4
      herramienta: MARTILLO
StepNode
  label: 10
  instructions:
    PonerInstrNode
      cantidad: 2
      tipoHerraje: ESPIGA
      idHerraje: 101339
      piezaDestino: BASE_4
      herramienta: MARTILLO
StepNode
  label: 11
  instructions:
```

```

    instructions:
      UnirInstrNode
        pieza1: LATERAL_5
        pieza2: BASE_4
  StepNode
    label:
      12
    1
    instructions:
      PonerInstrNode
        cantidad: 2
        tipoHerraje: ESPIGA
        idHerraje: 101345
        piezaDestino: BASE_2
        herramienta: MARTILLO
  StepNode
    label:
      12
    2
    instructions:
      PonerInstrNode
        cantidad: 2
        tipoHerraje: ESPIGA
        idHerraje: 101345
        piezaDestino: BASE_3
        herramienta: MARTILLO
  StepNode
    label:
      12
    3
    instructions:
      PonerInstrNode
        cantidad: 2
        tipoHerraje: ESPIGA
        idHerraje: 101345
        piezaDestino: BASE_4
        herramienta: MARTILLO
  StepNode
    label:
      12
    4
    instructions:
      PonerInstrNode
        cantidad: 2
        tipoHerraje: ESPIGA
        idHerraje: 101345
        piezaDestino: LATERAL_5
        herramienta: MARTILLO
  StepNode
    label:
      13
    instructions:
      UnirInstrNode
  StepNode
    label:
      13
    instructions:
      UnirInstrNode
        pieza1: LATERAL_6
        pieza2: BASE_1
  StepNode
    label:
      14
    instructions:
      UnirInstrNode
        pieza1: SUPERIOR
        pieza2: LATERAL_6
      ConHerramientaAtornillarInstrNode
        herramienta: DESTORNILLADOR
        inner:
          AtornillarInstrNode
            cantidad: 6
            idTornillo: 104321
            piezaDestino: SUPERIOR
            idTuerca: null
  StepNode
    label:
      15
    instructions:
      PonerInstrNode
        cantidad: 8
        tipoHerraje: HERRAJE
        idHerraje: 153550
        piezaDestino: LATERAL_6
        herramienta: MANO
  StepNode
    label:
      16
    instructions:
      ConHerramientaAtornillarInstrNode
        herramienta: DESTORNILLADOR
        inner:
          AtornillarInstrNode
            cantidad: 2
            idTornillo: 100372
            piezaDestino: LATERAL_1
            idTuerca: null
  StepNode
    label:
      19
    instructions:
      PonerInstrNode
        cantidad: 2
        tipoHerraje: HERRAJE
        idHerraje: 103693
        piezaDestino: LATERAL_1
        herramienta: MANO

```

- Hemnes:

### Árbol AST

```
ManualNode
  itemName: hemnes
  steps:
    StepNode
      label: 1
      instructions:
        ConHerramientaAtornillarInstrNode
          herramienta: DESTORNILLADOR
          inner:
            AtornillarInstrNode
              cantidad: 8
              idTornillo: 118331
              piezaDestino: LATERAL_1
              idTuerca: null
    StepNode
      label: 2
      instructions:
        PonerInstrNode
          cantidad: 6
          tipoHerraje: ESPIGA
          idHerraje: 101350
          piezaDestino: LATERAL_2
          herramienta: MANO
    StepNode
      label: 3
      instructions:
        ConHerramientaAtornillarInstrNode
          herramienta: DESTORNILLADOR
          inner:
            AtornillarInstrNode
              cantidad: 2
              idTornillo: 100212
              piezaDestino: LATERAL_2
              idTuerca: null
    StepNode
      label: 4
      instructions:
        ConHerramientaAtornillarInstrNode
          herramienta: DESTORNILLADOR
          inner:
            AtornillarInstrNode
              cantidad: 2
              idTornillo: 100211
              piezaDestino: LATERAL_1
              idTuerca: null
    StepNode
      label: 5
      instructions:
        PonerInstrNode
          cantidad: 4
          tipoHerraje: HERRAJE
          idHerraje: 102728
          piezaDestino: LATERAL_1
```

```

      piezaDestino: LATERAL_1
      herramienta: MANO
    StepNode
      label: 6
      instructions:
        UnirInstrNode
          pieza1: LATERAL_3
          pieza2: LATERAL_1
        PonerInstrNode
          cantidad: 2
          tipoHerraje: ESPIGA
          idHerraje: 101350
          piezaDestino: LATERAL_3
          herramienta: MANO
    StepNode
      label: 7
      instructions:
        RepetirInstrNode
          paso: 6
    StepNode
      label: 8
      instructions:
        RepetirInstrNode
          paso: 6
    StepNode
      label: 9
      instructions:
        PonerInstrNode
          cantidad: 4
          tipoHerraje: HERRAJE
          idHerraje: 113434
          piezaDestino: LATERAL_3
          herramienta: MANO
    StepNode
      label: 10
      instructions:
        ConHerramientaAtornillarInstrNode
          herramienta: DESTORNILLADOR
          inner:
            AtornillarInstrNode
              cantidad: 4
              idTornillo: 118331
              piezaDestino: LATERAL_2
              idTuerca: null
    StepNode
      label: 11
      instructions:
        UnirInstrNode
          pieza1: LATERAL_4
          pieza2: LATERAL_2
        ConHerramientaAtornillarInstrNode
          herramienta: DESTORNILLADOR
          inner:

```

```

                                herramienta: DESTORNILLADOR
                                inner:
                                AtornillarInstrNode
                                cantidad: 7
                                idTornillo: 100211
                                piezaDestino: SUPERIOR
                                idTuerca: null

StepNode
  label:
    16
    1
  instructions:
    GirarInstrNode
    orientacion: LATERAL

StepNode
  label:
    16
    2
  instructions:
    UnirInstrNode
    pieza1: LATERAL_6
    pieza2: LATERAL_7

StepNode
  label:
    17
  instructions:
    RepetirInstrNode
    paso:
      16
      2

StepNode
  label:
    18
  instructions:
    RepetirInstrNode
    paso:
      16
      2

StepNode
  label:
    19
  instructions:
    RepetirInstrNode
    paso:
      16
      2

StepNode
  label:
    20
    1
  instructions:
    GirarInstrNode
    orientacion: LATERAL

StepNode
  label:
    20
    2
  instructions:
    ConHerramientaAtornillarInstrNode
    herramienta: DESTORNILLADOR
    inner:
      herramienta: DESTORNILLADOR
      inner:
      AtornillarInstrNode
      cantidad: 7
      idTornillo: 100211
      piezaDestino: SUPERIOR
      idTuerca: null

inner:
  AtornillarInstrNode
  cantidad: 2
  idTornillo: 113434
  piezaDestino: LATERAL_4
  idTuerca: null

StepNode
  label:
    12
  instructions:
    UnirInstrNode
    pieza1: LATERAL_5
    pieza2: LATERAL_1
    PonerInstrNode
    cantidad: 3
    tipoHerraje: ESPIGA
    idHerraje: 101350
    piezaDestino: LATERAL_5
    herramienta: MANO

StepNode
  label:
    13
  instructions:
    PonerInstrNode
    cantidad: 6
    tipoHerraje: HERRAJE
    idHerraje: 113434
    piezaDestino: LATERAL_5
    herramienta: MANO

StepNode
  label:
    14
    1
  instructions:
    GirarInstrNode
    orientacion: LATERAL

StepNode
  label:
    14
    2
  instructions:
    UnirInstrNode
    pieza1: SUPERIOR
    pieza2: LATERAL_1

StepNode
  label:
    14
    3
  instructions:
    PonerInstrNode
    cantidad: 2
    tipoHerraje: ESPIGA
    idHerraje: 101350
    piezaDestino: SUPERIOR
    herramienta: MANO

StepNode
  label:
    15
  instructions:
    ConHerramientaAtornillarInstrNode
    herramienta: DESTORNILLADOR

```

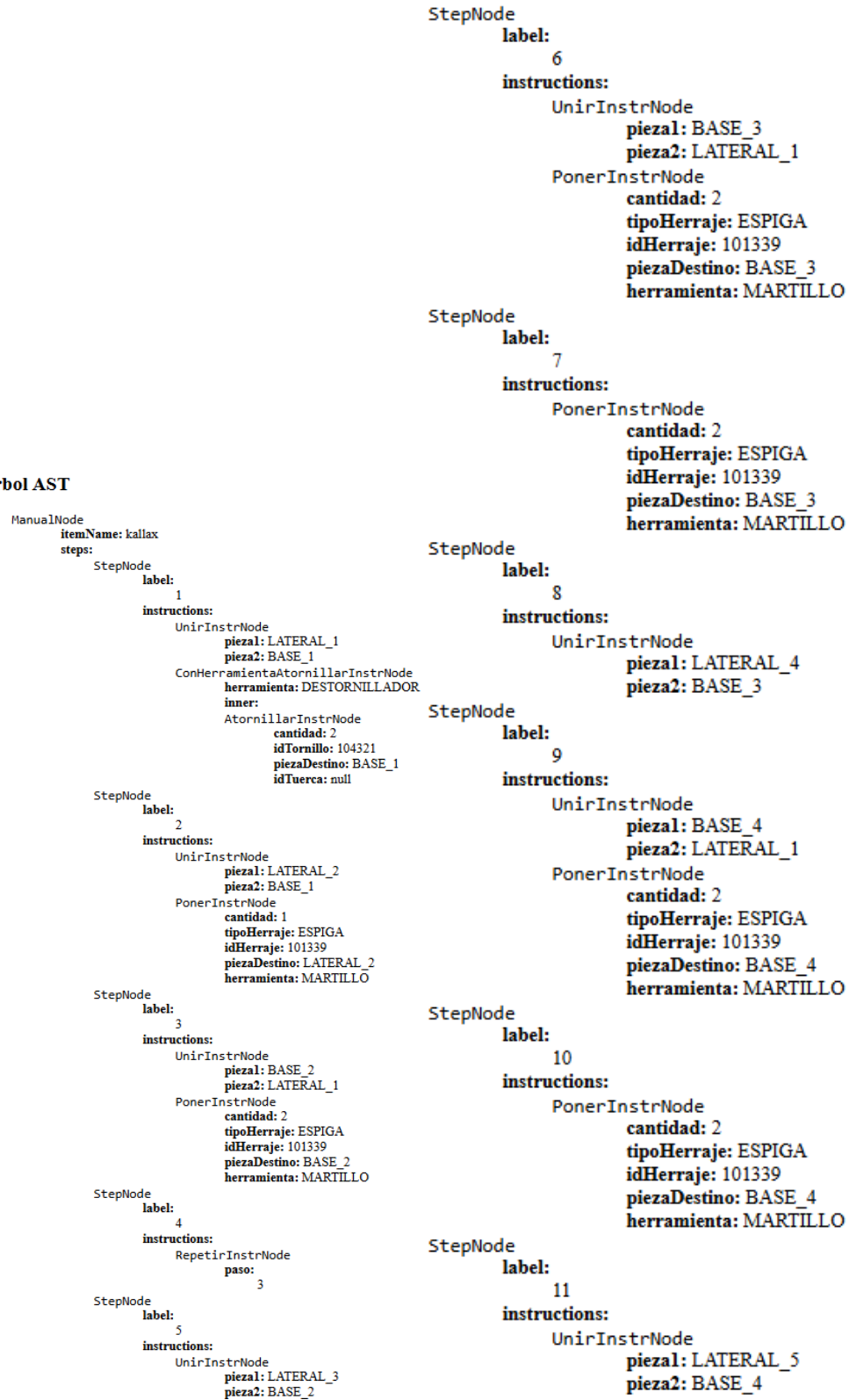
```

        AtornillarInstrNode
            cantidad: 3
            idTornillo: 109560
            piezaDestino: LATERAL_6
            idTuerca: null
StepNode
    label:
        21
    instructions:
        RepetirInstrNode
            paso:
                20
                2
StepNode
    label:
        22
    instructions:
        GirarInstrNode
            orientacion: ARRIBA
StepNode
    label:
        23
    instructions:
        ConHerramientaAtornillarInstrNode
            herramienta: DESTORNILLADOR
            inner:
                AtornillarInstrNode
                    cantidad: 2
                    idTornillo: 109560
                    piezaDestino: SUPERIOR
                    idTuerca: 100823
StepNode
    label:
        24
    instructions:
        PonerInstrNode
            cantidad: 4
            tipoHerraje: HERRAJE
            idHerraje: 104171
            piezaDestino: LATERAL_1
            herramienta: MANO
StepNode
    label:
        25
    instructions:
        UnirInstrNode
            pieza1: BASE
            pieza2: LATERAL_1
StepNode
    label:
        26
    instructions:
        PonerInstrNode
            cantidad: 8
            tipoHerraje: HERRAJE
            idHerraje: 104171
            piezaDestino: LATERAL_1
            herramienta: MANO

```

- Kallax:

### Árbol AST



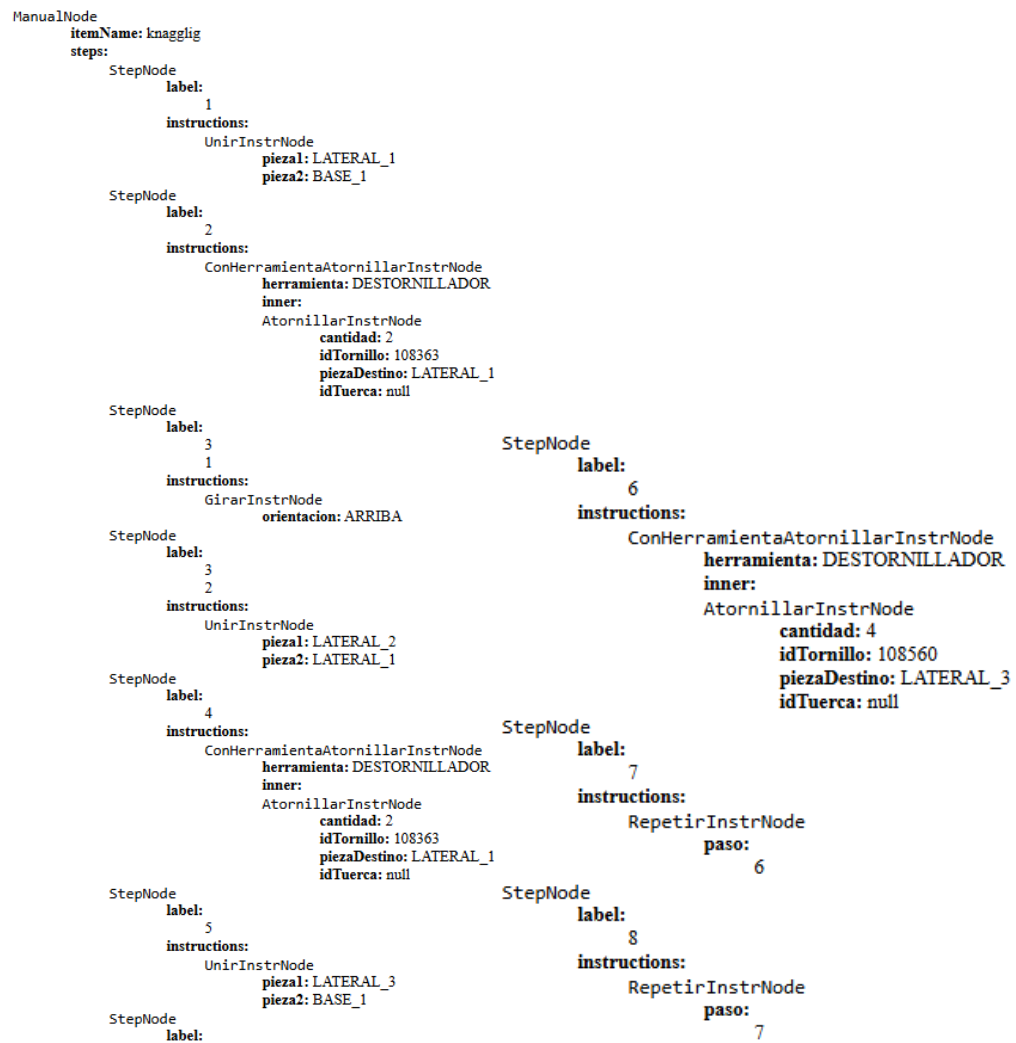
```

        instructions:
            UnirInstrNode
                pieza1: LATERAL_5
                pieza2: BASE_4
StepNode
    label:
        12
    instructions:
        PonerInstrNode
            cantidad: 8
            tipoHerraje: ESPIGA
            idHerraje: 101339
            piezaDestino: BASE
            herramienta: MARTILLO
StepNode
    label:
        13
    instructions:
        UnirInstrNode
            pieza1: LATERAL_6
            pieza2: BASE
StepNode
    label:
        14
    instructions:
        UnirInstrNode
            pieza1: SUPERIOR
            pieza2: LATERAL_1
StepNode
    label:
        15
    instructions:
        ConHerramientaAtornillarInstrNode
            herramienta: DESTORNILLADOR
            inner:
                AtornillarInstrNode
                    cantidad: 6
                    idTornillo: 104322
                    piezaDestino: SUPERIOR
                    idTuerca: null
StepNode
    label:
        16
    instructions:
        PonerInstrNode
            cantidad: 2
            tipoHerraje: HERRAJE
            idHerraje: 100372
            piezaDestino: SUPERIOR
            herramienta: MANO
StepNode
    label:
        19
    instructions:
        PonerInstrNode
            cantidad: 2
            tipoHerraje: HERRAJE
            idHerraje: 115754
            piezaDestino: LATERAL_1
            herramienta: MANO

```

- Knagglig:

### Árbol AST



- Lack:



## Árbol AST

```

ManualNode
  itemName: lack
  steps:
    StepNode
      label:
        1
        1
      instructions:
        UnirInstrNode
          pieza1: LATERAL_1
          pieza2: BASE
    StepNode
      label:
        1
        2
      instructions:
        ConHerramientaAtornillarInstrNode
          herramienta: DESTORNILLADOR
          inner:
            AtornillarInstrNode
              cantidad: 12
              idTornillo: 104323
              piezaDestino: BASE
              idTuerca: null
    StepNode
      label:
        2
      instructions:
        ConHerramientaAtornillarInstrNode
          herramienta: DESTORNILLADOR
          inner:
            AtornillarInstrNode
              cantidad: 2
              idTornillo: 104325
              piezaDestino: BASE
              idTuerca: null

```

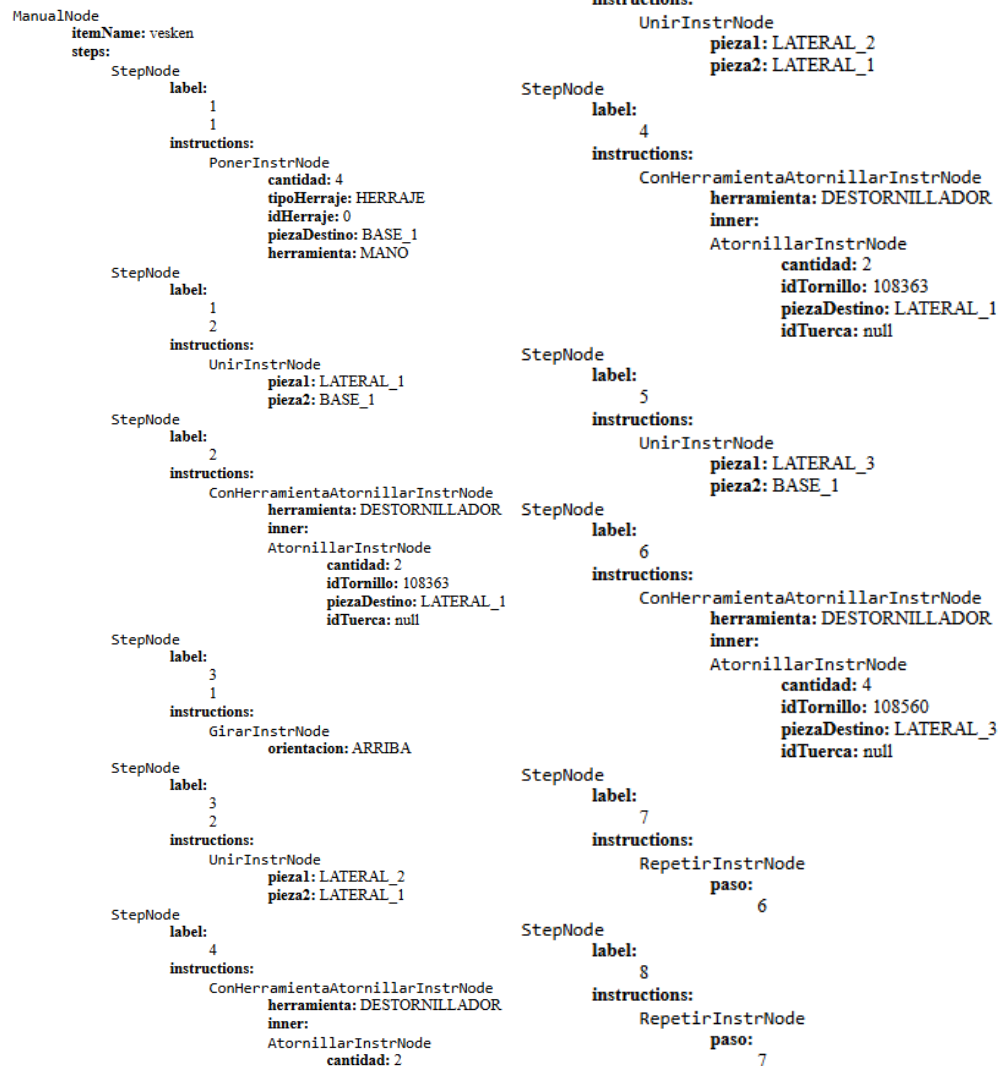
- Smagora:

Árbol AST

```
ManualNode
  itemName: smagora
  steps:
    StepNode
      label: 1
      instructions:
        ConHerramientaAtornillarInstrNode
          herramienta: DESTORNILLADOR
          inner:
            AtornillarInstrNode
              cantidad: 16
              idTornillo: 118331
              piezaDestino: LATERAL_1
              idTuerca: null
    StepNode
      label: 2
      instructions:
        UnirInstrNode
          pieza1: SUPERIOR
          pieza2: LATERAL_1
        PonerInstrNode
          cantidad: 4
          tipoHerraje: ESPIGA
          idHerraje: 101345
          piezaDestino: SUPERIOR
          herramienta: MANO
    StepNode
      label: 2
      instructions:
        UnirInstrNode
          pieza1: BASE_1
          pieza2: LATERAL_1
        PonerInstrNode
          cantidad: 4
          tipoHerraje: ESPIGA
          idHerraje: 101345
          piezaDestino: BASE_1
          herramienta: MANO
    StepNode
      label: 3
      instructions:
        UnirInstrNode
          pieza1: BASE_2
          pieza2: LATERAL_1
        PonerInstrNode
          cantidad: 4
          tipoHerraje: ESPIGA
          idHerraje: 101345
          piezaDestino: BASE_2
          herramienta: MANO
```

- Vesken:

```
StepNode
  label: 3
  instructions:
    UnirInstrNode
      pieza1: BASE_3
      pieza2: LATERAL_1
    PonerInstrNode
      cantidad: 4
      tipoHerraje: ESPIGA
      idHerraje: 101345
      piezaDestino: BASE_3
      herramienta: MANO
StepNode
  label: 4
  instructions:
    PonerInstrNode
      cantidad: 8
      tipoHerraje: HERRAJE
      idHerraje: 119252
      piezaDestino: BASE
      herramienta: MANO
StepNode
  label: 5
  instructions:
    UnirInstrNode
      pieza1: LATERAL_2
      pieza2: LATERAL_1
StepNode
  label: 6
  instructions:
    GirarInstrNode
      orientacion: LATERAL
StepNode
  label: 7
  instructions:
    PonerInstrNode
      cantidad: 8
      tipoHerraje: HERRAJE
      idHerraje: 119252
      piezaDestino: BASE
      herramienta: MANO
StepNode
  label: 8
  instructions:
    PonerInstrNode
      cantidad: 2
      tipoHerraje: HERRAJE
      idHerraje: 107716
      piezaDestino: LATERAL_1
      herramienta: MANO
StepNode
  label: 9
  instructions:
```

**Árbol AST**

## 2.3. Nivel avanzado

Finalmente, la práctica incluye un nivel adicional orientado a aplicar los conocimientos de análisis semántico y tablas de símbolos sobre el AST previamente construido.

Además, se pide que el analizador pueda leer los pasos y escribir la lista de herrajes y herramientas, con sus cantidades.

En este apartado se implementó un analizador capaz de interpretar una secuencia completa de montaje y extraer información útil para el usuario.

### 2.3.1. Desarrollo

Para cumplir con los requisitos, se implementó la clase `IkeaAnalyzer`, que realiza el siguiente flujo:

1. Interpretación del AST: para que el analizador no vuelva a leer el parse tree y tranaje exclusivamente sobre el AST, ya que contiene una estructura semántica depurada.
2. Índice de pasos: para poder procesar correctamente las instrucciones del tipo “REPETIR”.

3. En cuanto a la gestión de la tabla de símbolos: se implementaron dos estructuras principales: una de tipo Map<String, Integer> herrajes y otra de tipo Set<String> herramientas, las cuales recogen los distintos herrajes y herramientas (con su cantidad) utilizados a lo largo del manual. Ambos se actualizan mientras se recorren los pasos del manual.

A continuación, se muestran los resultados tras ejecutar el analizador sobre el manual del mueble Billy:

```
> java -cp ".;generated;..\lib\antlr-4.13.2-complete.jar" IkeaAnalyzer ..\traducciones\billy.txt
ITEM: billy
Pasos totales: 15

Herrajes necesarios:
  ESPIGA 101351: 16 uds
  HERRAJE 131372: 16 uds
  TORNILLO 100823: 2 uds
  TORNILLO 101201: 18 uds
  TORNILLO 109041: 2 uds
  TORNILLO 118331: 12 uds
  TORNILLO 119081: 10 uds
  TUERCA 100823: 2 uds

Herramientas a utilizar:
  DESTORNILLADOR
  MANO
```

Esto funciona para todos los manuales traducidos.

## Conclusiones

Con la realización de la práctica hemos aprendido a crear analizadores léxicos y sintácticos que reconozcan un determinado lenguaje, así como a crear el AST asociado al mismo. Esto se ha vuelto especialmente didáctico en la segunda parte de la práctica, debido a la libertad que teníamos, tanto para crear el lenguaje y su gramática como para traducir los manuales.

La experiencia de desarrollo ha permitido poner en práctica los conceptos vistos en clase (expresiones regulares, tabla de símbolos, gestión de errores, gramáticas, etc.) de manera integrada y visual, reforzando nuestra comprensión de la materia, tanto de un punto de vista del proceso de construcción de un lenguaje formal como de la creación de unas reglas léxicas y sintácticas para trabajar con el mismo.