



TRABAJO FIN DE GRADO

Especialidad de Computación

Grado en Ingeniería Informática

SimAS 3.0 descendente predictivo.

Simulador de analizadores sintácticos descendentes predictivos.

Manual de usuario

Autor

D. Antonio Llamas García

Directores

Prof. Dr. Antonio Arauzo Azofra

Prof. Dra. María Luque Rodríguez

Septiembre, 2025



UNIVERSIDAD DE CÓRDOBA

Índice general

1. Introducción	1
1.1. ¿Qué es SimAS 3.0?	1
1.2. Objetivos de SimAS 3.0	1
1.3. Características principales	2
1.3.1. Editor de gramáticas	2
1.3.2. Simulador descendente predictivo	2
1.3.3. Interfaz de usuario	2
1.4. Requisitos del sistema	2
1.4.1. Requisitos mínimos	3
1.4.2. Requisitos recomendados	3
1.5. Organización del manual	3
1.6. Convenciones utilizadas	3
1.7. Notas importantes	4
1.8. Contacto y Soporte	4
2. Instalación y configuración	5
2.1. Requisitos del sistema	5
2.1.1. Requisitos mínimos	5
2.1.2. Requisitos recomendados	5
2.2. Instalación de Java	6
2.2.1. Verificación de Java	6
2.2.2. Descarga e instalación de Java	6
2.3. Descarga de SimAS 3.0	6
2.3.1. Ubicación de los archivos	6
2.3.2. Verificación de la descarga	7

ÍNDICE GENERAL

2.4. Instalación de SimAS 3.0	7
2.4.1. Instalación en macOS	7
2.4.2. Instalación en Windows	7
2.4.3. Instalación en Linux	7
2.5. Primera ejecución	8
2.5.1. Inicio de la aplicación	8
2.5.2. Pantalla de bienvenida	8
2.5.3. Configuración inicial	8
2.6. Configuración de la aplicación	8
2.6.1. Selección de idioma	8
2.6.2. Gestión de archivos	9
2.7. Desinstalación	9
2.7.1. Desinstalación en macOS	9
2.7.2. Desinstalación en Windows	9
2.7.3. Desinstalación en Linux	10
2.8. Solución de problemas	10
2.8.1. Problemas comunes	10
2.8.2. Información para soporte técnico	10
2.9. Actualizaciones futuras	10
2.9.1. Política de versiones	10
2.9.2. Información sobre nuevas versiones	11
2.9.3. Migración entre versiones	11
3. Características de la interfaz	13
3.1. Introducción a la interfaz	13
3.2. Pantalla de bienvenida	13
3.3. Menú principal	14
3.3.1. Elementos del menú principal	14
3.3.2. Botones principales	15
3.4. Sistema de pestañas	16
3.4.1. Características del sistema de pestañas	16
3.4.2. Gestión de pestañas	16

3.5.	Selector de idioma	16
3.5.1.	Idiomas disponibles	17
3.5.2.	Cambio de idioma	17
3.6.	Atajos de teclado	18
3.6.1.	Atajos disponibles	18
3.6.2.	Uso de atajos	19
3.7.	Elementos visuales	19
3.7.1.	Iconos y gráficos	19
3.7.2.	Tipografía y espaciado	20
3.8.	Responsividad de la interfaz	20
3.8.1.	Adaptación a la pantalla	20
3.8.2.	Optimización para diferentes resoluciones	20
3.9.	Accesibilidad	20
3.9.1.	Características de accesibilidad	21
3.9.2.	Compatibilidad con tecnologías asistivas	21
3.10.	Personalización de la interfaz	21
3.10.1.	Opciones de personalización	21
3.10.2.	Limitaciones de personalización	21
3.11.	Integración entre componentes	22
3.11.1.	Flujo de trabajo integrado	22
3.11.2.	Comunicación entre ventanas	22
3.12.	Solución de problemas de interfaz	22
3.12.1.	Problemas comunes	22
3.12.2.	Optimización del rendimiento	23
3.13.	Ventanas secundarias	23
3.13.1.	Creación de ventanas secundarias	23
3.13.2.	Características de las ventanas secundarias	23
3.13.3.	Gestión de múltiples ventanas	24
3.14.	Barra de estado y notificaciones	24
3.14.1.	Elementos de la barra de estado	24
3.14.2.	Sistema de notificaciones	24
3.15.	Menús contextuales	24

ÍNDICE GENERAL

3.15.1. Menús contextuales de pestañas	25
3.16. Conclusión	25
4. Editor de gramáticas	27
4.1. Introducción al editor	27
4.1.1. Características principales	27
4.1.2. Objetivos del editor	28
4.2. Acceso al editor	28
4.2.1. Métodos de acceso	28
4.3. Interfaz del editor	28
4.3.1. Estado vacío del editor	28
4.3.2. Estado con gramática cargada	30
4.3.3. Barra de herramientas	31
4.4. Asistente guiado de creación	31
4.4.1. Acceso al asistente	31
4.4.2. Proceso de creación	32
4.4.3. Navegación del asistente	32
4.4.4. Paso 1: Datos de la gramática	32
4.4.4.1. Interfaz del paso 1	32
4.4.4.2. Campos del paso 1	33
4.4.4.3. Estado vacío del paso 1	34
4.4.4.4. Requisitos del paso 1	34
4.4.5. Paso 2: Símbolos	35
4.4.5.1. Interfaz del paso 2	35
4.4.5.2. Estado vacío del paso 2	36
4.4.5.3. Elementos del paso 2	37
4.4.5.4. Paneles auxiliares	38
4.4.5.5. Panel de símbolos terminales	38
4.4.5.6. Elementos del panel de símbolos terminales	39
4.4.5.7. Funcionalidades del panel de símbolos terminales	40
4.4.5.8. Panel de símbolos no terminales	40
4.4.5.9. Elementos del panel de símbolos no terminales	41

4.4.5.10. Funcionalidades del panel de símbolos no terminales	42
4.4.5.11. Requisitos del paso 2	42
4.4.6. Paso 3: Producciones	42
4.4.6.1. Interfaz del paso 3	42
4.4.6.2. Estado vacío del paso 3	43
4.4.6.3. Elementos del paso 3	44
4.4.6.4. Panel de producciones	45
4.4.6.5. Elementos del panel de producciones	45
4.4.6.6. Funcionalidades del panel de producciones	46
4.4.6.7. Formato de producciones	46
4.4.6.8. Requisitos del paso 3	46
4.4.7. Paso 4: Símbolo inicial	47
4.4.7.1. Interfaz del paso 4	47
4.4.7.2. Estado vacío del paso 4	48
4.4.7.3. Selección del símbolo inicial	49
4.4.7.4. Finalización del proceso	49
4.5. Gestión de archivos	50
4.5.1. Formato de archivo	50
4.5.2. Operaciones de archivo	50
4.5.3. Ubicación de archivos	50
4.6. Validación de gramáticas	51
4.6.1. Tipos de validación	51
4.6.2. Proceso de validación manual	51
4.6.3. Validación exitosa	51
4.6.4. Validación con errores	52
4.6.5. Tipos de errores detectados	53
4.6.6. Ejemplo de error: símbolo duplicado	54
4.6.7. Corrección de errores	54
4.7. Integración con el simulador	55
4.7.1. Transición al simulador	55
4.7.2. Sincronización de datos	55
4.8. Resolución de problemas	55

ÍNDICE GENERAL

4.8.1. Problemas de validación	55
4.8.2. Problemas de archivos	56
4.9. Conclusión	56
5. Simulación del analizador descendente predictivo	57
5.1. Introducción al simulador	57
5.1.1. Características principales	57
5.2. Acceso al simulador	58
5.2.1. Desde el menú principal	58
5.2.2. Desde el editor de gramáticas	58
5.3. Asistente guiado del simulador	59
5.3.1. Navegación del asistente	59
5.3.2. Paso 1: Refactorización y eliminación de recursividad	60
5.3.2.1. Procesos realizados	60
5.3.2.2. Interfaz del paso 1	61
5.3.3. Paso 2: Construcción de conjuntos PRIMERO y SIGUIENTE	62
5.3.3.1. Conjunto PRIMERO	62
5.3.3.2. Conjunto SIGUIENTE	63
5.3.3.3. Interfaz del paso 2	63
5.3.4. Paso 3: Construcción de la tabla predictiva	64
5.3.4.1. Algoritmo de construcción	64
5.3.4.2. Interfaz del paso 3	65
5.3.5. Paso 4: Funciones de error	66
5.3.5.1. Funciones de error del sistema	66
5.3.5.2. Panel de funciones de error	67
5.3.5.3. Gestión de funciones de error	69
5.3.6. Paso 5: Tabla predictiva completa	71
5.3.6.1. Interfaz del paso 5	71
5.3.6.2. Herramientas de edición	73
5.4. Simulador principal	73
5.4.1. Interfaz del simulador	73
5.4.2. Generación de informes	74

5.5.	Simulación interactiva	75
5.5.1.	Interfaz de simulación	75
5.5.2.	Proceso de simulación	77
5.5.3.	Historial de pasos	78
5.5.4.	Navegación en la simulación	79
5.6.	Visualización de resultados	80
5.6.1.	Derivación	80
5.6.2.	Árbol sintáctico	81
5.7.	Múltiples simulaciones	83
5.7.1.	Gestión de simulaciones	83
5.7.2.	Ventajas de múltiples simulaciones	84
5.8.	Integración con el editor	85
5.9.	Resolución de problemas	86
5.9.1.	Problemas comunes	86
5.9.2.	Soluciones recomendadas	86
5.10.	Conclusión	87
6.	Ejemplos prácticos	89
6.1.	Introducción	89
6.2.	Creación de una gramática de contexto libre	89
6.2.1.	Acceso al editor	89
6.2.2.	Paso 1: Datos de la gramática	90
6.2.3.	Paso 2: Símbolos terminales y no terminales	91
6.2.3.1.	Símbolos terminales	92
6.2.3.2.	Símbolos no terminales	93
6.2.4.	Paso 3: Producciones	94
6.2.5.	Paso 4: Símbolo inicial	96
6.2.6.	Validación de la gramática	97
6.2.7.	Vista final del editor	98
6.3.	Simulación del análisis sintáctico descendente	99
6.3.1.	Acceso al simulador	99
6.3.2.	Paso 1: Refactorización de la gramática	100

ÍNDICE GENERAL

6.3.3. Paso 2: Construcción de conjuntos PRIMERO y SIGUIENTE	101
6.3.4. Paso 3: Construcción de la tabla predictiva	102
6.3.5. Paso 4: Funciones de error	103
6.3.6. Paso 5: Tabla predictiva completa	105
6.3.7. Simulador principal	106
6.3.8. Simulación interactiva	107
6.3.9. Derivación de la simulación	108
6.3.10. Árbol de análisis sintáctico	109
6.4. Conclusión del ejemplo	110

Bibliografía	113
---------------------	------------

Índice de figuras

3.1. Menú principal de SimAS 3.0	15
3.2. Selector de idiomas con menú desplegable	18
4.1. Editor de gramáticas en estado vacío	29
4.2. Editor de gramáticas con una gramática cargada	30
4.3. Paso 1: Datos de la gramática	33
4.4. Paso 1: Estado vacío	34
4.5. Paso 2: Símbolos de la gramática	36
4.6. Paso 2: Estado vacío	37
4.7. Panel de símbolos terminales	39
4.8. Panel de símbolos no terminales	41
4.9. Paso 3: Producciones de la gramática	43
4.10. Paso 3: Estado vacío	44
4.11. Panel de producciones	45
4.12. Paso 4: Selección del símbolo inicial	48
4.13. Paso 4: Estado vacío (no se puede finalizar)	49
4.14. Validación exitosa de la gramática	52
4.15. Validación con errores en la gramática	53
4.16. Ejemplo de error: símbolo duplicado	54
5.1. Vista de la gramática original desde el botón "Gramática".	60
5.2. Paso 1: Refactorización y eliminación de recursividad	62
5.3. Paso 2: Construcción de conjuntos PRIMERO y SIGUIENTE	64
5.4. Paso 3: Construcción de la tabla predictiva	66
5.5. Panel auxiliar para crear nuevas funciones de error	68

5.6.	Panel de funciones de error en estado vacío	69
5.7.	Paso 4: Configuración de funciones de error	70
5.8.	Opción de no usar funciones de error	71
5.9.	Paso 5: Tabla predictiva completa con herramientas de edición	72
5.10.	Interfaz del simulador principal	74
5.11.	Interfaz de simulación en estado vacío	76
5.12.	Interfaz de simulación con cadena de entrada	77
5.13.	Simulación completa con historial de pasos	79
5.14.	Vista de la derivación durante la simulación	81
5.15.	Vista del árbol sintáctico durante la simulación	83
5.16.	Ejemplo de múltiples simulaciones simultáneas	85
6.1.	Acceso al editor desde el menú principal	90
6.2.	Paso 1: Configuración de datos básicos de la gramática	91
6.3.	Paso 2: Definición de símbolos terminales y no terminales	92
6.4.	Panel auxiliar para gestión de símbolos terminales	93
6.5.	Panel auxiliar para gestión de símbolos no terminales	94
6.6.	Paso 3: Definición de reglas de producción	95
6.7.	Panel auxiliar para gestión de producciones	96
6.8.	Paso 4: Selección del símbolo inicial	97
6.9.	Validación exitosa de la gramática	98
6.10.	Vista final del editor con la gramática completa	99
6.11.	Paso 1: Refactorización de la gramática	100
6.12.	Paso 2: Conjuntos PRIMERO y SIGUIENTE	101
6.13.	Paso 3: Tabla predictiva	103
6.14.	Paso 4: Configuración de funciones de error	104
6.15.	Panel auxiliar para añadir funciones de error	105
6.16.	Paso 5: Tabla predictiva completa	106
6.17.	Simulador principal con resumen de configuración	107
6.18.	Introducción de cadena de entrada para simulación	108
6.19.	Derivación paso a paso del análisis sintáctico	109
6.20.	Árbol de análisis sintáctico	110

ÍNDICE DE FIGURAS

Capítulo 1

Introducción

1.1. ¿Qué es SimAS 3.0?

SimAS 3.0 (Simulador de Análisis Sintáctico) es una aplicación educativa desarrollada en Java que permite a estudiantes y profesores de informática comprender y practicar los conceptos fundamentales del análisis sintáctico descendente predictivo [8]. Esta herramienta proporciona un entorno interactivo y visual para el aprendizaje de una de las técnicas más importantes en el diseño de compiladores [1].

La aplicación está diseñada específicamente para facilitar el estudio de los analizadores sintáticos descendentes predictivos, una clase de analizadores que construyen el árbol de análisis sintáctico de arriba hacia abajo, utilizando una tabla predictiva para determinar qué producción aplicar en cada paso del análisis.

1.2. Objetivos de SimAS 3.0

Los principales objetivos de SimAS 3.0 son:

- **Educativo:** proporcionar una herramienta didáctica que facilite la comprensión de los analizadores sintáticos descendentes predictivos.
- **Interactivo:** permitir a los usuarios experimentar con diferentes gramáticas y observar el comportamiento del analizador en tiempo real.
- **Visual:** ofrecer representaciones gráficas claras de los procesos de análisis sintáctico.
- **Práctico:** facilitar la creación, edición y validación de gramáticas de contexto libre.
- **Completo:** cubrir todos los aspectos del análisis descendente predictivo, desde la construcción de la tabla predictiva hasta la simulación del análisis.

1.3. Características principales

SimAS 3.0 ofrece las siguientes características principales:

1.3.1. Editor de gramáticas

- **Asistente guiado:** proceso de creación de gramáticas en 4 pasos claramente definidos.
- **Validación automática:** verificación de la corrección sintáctica y semántica de las gramáticas.
- **Gestión de archivos:** capacidad para guardar, cargar y gestionar múltiples gramáticas.
- **Generación de informes:** creación automática de informes PDF con la información de la gramática.

1.3.2. Simulador descendente predictivo

- **Proceso paso a paso:** simulación detallada del análisis sintáctico con control manual.
- **Construcción de tablas:** generación automática de las tablas con los conjuntos PRIMERO, SIGUIENTE y la tabla predictiva [3].
- **Recuperación de errores:** implementación de funciones de error para manejar cadenas incorrectas.
- **Visualización clara:** representación gráfica de la pila, entrada y acciones realizadas.

1.3.3. Interfaz de usuario

- **Multilingüe:** soporte para español, inglés, francés, alemán, japonés y portugués.
- **Intuitiva:** diseño moderno y fácil de usar basado en JavaFX [6].
- **Gestión de pestanas:** trabajo simultáneo con múltiples gramáticas y simulaciones.
- **Atajos de teclado:** acceso rápido a las funciones principales.

1.4. Requisitos del sistema

Para ejecutar SimAS 3.0, el sistema debe cumplir los siguientes requisitos:

1.4.1. Requisitos mínimos

- **Sistema operativo:** Windows 10/11, macOS 10.14+, o Linux (Ubuntu 18.04+).
- **Java Runtime Environment:** versión 17 o superior [5].
- **Memoria RAM:** 512 MB disponibles.
- **Espacio en disco:** 100 MB libres.
- **Resolución de pantalla:** 1024x768 píxeles mínimo.

1.4.2. Requisitos recomendados

- **Memoria RAM:** 2 GB o más.
- **Resolución de pantalla:** 1366x768 píxeles o superior.
- **Procesador:** dual-core 2.0 GHz o superior.

1.5. Organización del manual

Este manual de usuario está organizado de la siguiente manera:

- **Capítulo 2:** instalación y configuración del programa, incluyendo requisitos del sistema y procedimientos de instalación.
- **Capítulo 3:** características de la interfaz de usuario, navegación y configuración de idiomas.
- **Capítulo 4:** uso del editor de gramáticas, desde la creación hasta la validación.
- **Capítulo 5:** funcionamiento del simulador descendente predictivo y análisis de resultados.
- **Capítulo 6:** ejemplos prácticos completos que ilustran el uso de la aplicación.

1.6. Convenciones utilizadas

A lo largo de este manual se utilizan las siguientes convenciones:

- **Texto en negrita:** se utiliza para resaltar elementos de la interfaz, nombres de archivos o conceptos importantes.
- **Cursiva:** se emplea para términos técnicos o referencias a otros capítulos.
- **“Texto entre comillas”:** se usa para citar mensajes del sistema o nombres de archivos.

- **Menú >Opción:** indica la secuencia de navegación por los menús de la aplicación.
- **Ctrl+Tecla:** representa las combinaciones de teclas de atajo.

1.7. Notas importantes

Antes de comenzar a utilizar SimAS 3.0, es importante tener en cuenta lo siguiente:

- La aplicación está diseñada para fines educativos y de aprendizaje.
- Se recomienda tener conocimientos básicos de teoría de lenguajes formales y análisis sintáctico.
- Las gramáticas deben estar en forma de Backus-Naur (BNF) [7].
- La aplicación incluye funciones de recuperación de errores, pero es importante entender sus limitaciones.

1.8. Contacto y Soporte

Para obtener soporte técnico o reportar problemas con SimAS 3.0, puede contactar con:

- **Desarrollador:** Antonio Llamas García.
- **Director del proyecto:** Prof. Dr. Antonio Arauzo Azofra & Prof. Dr. María Luque Rodríguez.
- **Institución:** Escuela Politécnica Superior de Córdoba.

Este manual proporciona una guía completa para el uso de SimAS 3.0. Se recomienda leer los capítulos en orden secuencial para obtener el máximo provecho de la aplicación.

Capítulo 2

Instalación y configuración

2.1. Requisitos del sistema

Antes de proceder con la instalación de SimAS 3.0, es importante verificar que el sistema cumple con los requisitos necesarios para el correcto funcionamiento de la aplicación.

2.1.1. Requisitos mínimos

Para ejecutar SimAS 3.0, el sistema debe cumplir los siguientes requisitos mínimos:

- **Sistema operativo:** Windows 10/11, macOS 10.14+, o Linux (Ubuntu 18.04+).
- **Java Runtime Environment:** versión 17 o superior [5].
- **Memoria RAM:** 512 MB disponibles.
- **Espacio en disco:** 100 MB libres.
- **Resolución de pantalla:** 1024x768 píxeles mínimo.

2.1.2. Requisitos recomendados

Para obtener un rendimiento óptimo, se recomienda:

- **Memoria RAM:** 2 GB o más.
- **Resolución de pantalla:** 1366x768 píxeles o superior.
- **Procesador:** dual-core 2.0 GHz o superior.
- **Espacio en disco:** 200 MB libres (para almacenar gramáticas y/o informes).

2.2. Instalación de Java

SimAS 3.0 requiere Java Runtime Environment (JRE) versión 17 o superior [5]. Si no tiene Java instalado o tiene una versión anterior, siga estos pasos:

2.2.1. Verificación de Java

Para verificar si Java está instalado y qué versión tiene:

- **Windows:** abra el símbolo del sistema (cmd) y ejecute: `java -version`.
- **macOS:** abra Terminal y ejecute: `java -version`.
- **Linux:** abra una terminal y ejecute: `java -version`.

2.2.2. Descarga e instalación de Java

Si no tiene Java instalado o tiene una versión anterior a la 17:

1. Visite el sitio web oficial de Java: [5].
2. Descargue Java 17 LTS (Long Term Support) para su sistema operativo.
3. Ejecute el instalador y siga las instrucciones.
4. Reinicie el sistema si es necesario.
5. Verifique la instalación ejecutando `java -version` nuevamente.

Nota: Como alternativa de código abierto, también puede descargar Java desde [2].

2.3. Descarga de SimAS 3.0

2.3.1. Ubicación de los archivos

SimAS 3.0 se distribuye como una aplicación independiente que incluye todas las dependencias necesarias. Los archivos de distribución se encuentran en:

- **Aplicación ejecutable:** `SimAS.app` (macOS) o `SimAS.exe` (Windows).
- **Archivo JAR:** `SimAS.jar` (para ejecución manual).
- **Documentación:** manual de usuario y archivos de ayuda.

2.3.2. Verificación de la descarga

Antes de proceder con la instalación, verifique que:

- Los archivos se descargaron completamente.
- No hay errores de corrupción en los archivos.
- Tiene permisos de lectura y ejecución en los archivos.

2.4. Instalación de SimAS 3.0

2.4.1. Instalación en macOS

Para instalar SimAS 3.0 en macOS:

1. Localice el archivo **SimAS.app** en su carpeta de descargas.
2. Arrastre **SimAS.app** a la carpeta **Aplicaciones**.
3. Opcionalmente, puede crear un alias en el escritorio para acceso rápido.
4. La aplicación estará disponible en el Launchpad y en la carpeta Aplicaciones.

2.4.2. Instalación en Windows

Para instalar SimAS 3.0 en Windows:

1. Localice el archivo **SimAS.exe** en su carpeta de descargas.
2. Haga clic derecho sobre el archivo y seleccione "Ejecutar como administrador".
3. Siga las instrucciones del instalador.
4. La aplicación se instalará en la carpeta **Program Files**.
5. Se creará un acceso directo en el escritorio y en el menú Inicio.

2.4.3. Instalación en Linux

Para instalar SimAS 3.0 en Linux:

1. Abra una terminal y navegue a la carpeta donde descargó SimAS 3.0.
2. Haga el archivo ejecutable: `chmod +x SimAS.jar`.
3. Cree un directorio para la aplicación: `sudo mkdir /opt/simas`.
4. Copie los archivos: `sudo cp -r * /opt/simas/`.
5. Cree un acceso directo en el escritorio o menú de aplicaciones.

2.5. Primera ejecución

2.5.1. Inicio de la aplicación

Para ejecutar SimAS 3.0 por primera vez:

1. **macOS**: haga doble clic en `SimAS.app` o búsqelo en Launchpad.
2. **Windows**: haga doble clic en el acceso directo del escritorio o búsqelo en el menú Inicio.
3. **Linux**: ejecute `java -jar /opt/simas/SimAS.jar` o use el acceso directo creado.

2.5.2. Pantalla de bienvenida

La primera vez que ejecute SimAS 3.0, verá:

- La pantalla de bienvenida con el logo de la aplicación.
- Información sobre la versión y el desarrollador.
- Un indicador de progreso mientras se cargan los componentes.

2.5.3. Configuración inicial

Después de la primera ejecución:

1. La aplicación se abrirá en el menú principal.
2. Se establecerán los valores predeterminados para la interfaz.
3. El idioma se configurará en español por defecto.
4. La aplicación estará lista para crear y gestionar gramáticas.

2.6. Configuración de la aplicación

2.6.1. Selección de idioma

SimAS 3.0 soporta múltiples idiomas. Para cambiar el idioma:

1. Localice el selector de idioma en la esquina superior derecha.
2. Haga clic en el menú desplegable.
3. Seleccione el idioma deseado de la lista disponible.
4. La interfaz se actualizará inmediatamente al nuevo idioma.

2.6.2. Gestión de archivos

SimAS 3.0 permite guardar y cargar archivos en cualquier ubicación de su sistema:

- **Gramáticas:** se guardan en formato XML [9] en la ubicación que elija el usuario.
- **Informes PDF:** se generan en la ubicación seleccionada por el usuario usando iText [4].
- **Archivos de configuración:** se almacenan temporalmente en memoria durante la sesión.
- **Archivos de ayuda:** incluidos en la aplicación, no requieren instalación adicional.

2.7. Desinstalación

2.7.1. Desinstalación en macOS

Para desinstalar SimAS 3.0 en macOS:

1. Abra la carpeta **Aplicaciones**.
2. Arrastre **SimAS.app** a la papelera.
3. No es necesario eliminar carpetas adicionales, ya que SimAS 3.0 no crea archivos de configuración permanentes.

2.7.2. Desinstalación en Windows

Para desinstalar SimAS 3.0 en Windows:

1. Abra el Panel de control.
2. Seleccione "Programas y características".
3. Busque "SimAS 3.0" en la lista.
4. Haga clic en "Desinstalar" y siga las instrucciones.
5. No es necesario eliminar carpetas adicionales, ya que SimAS 3.0 no crea archivos de configuración permanentes.

2.7.3. Desinstalación en Linux

Para desinstalar SimAS 3.0 en Linux:

1. Elimine la carpeta de instalación: `sudo rm -rf /opt/simas`.
2. Elimine los accesos directos creados.
3. No es necesario eliminar carpetas adicionales, ya que SimAS 3.0 no crea archivos de configuración permanentes.

2.8. Solución de problemas

2.8.1. Problemas comunes

Si experimenta problemas durante la instalación o ejecución:

- **Error "Java no encontrado"**: verifique que Java 17+ esté instalado correctamente.
- **Aplicación no se abre**: compruebe los permisos de ejecución del archivo.
- **Error de memoria**: cierre otras aplicaciones para liberar memoria RAM.
- **Interfaz no se muestra correctamente**: verifique la resolución de pantalla.

2.8.2. Información para soporte técnico

Si necesita ayuda técnica, proporcione la siguiente información:

1. Sistema operativo y versión.
2. Versión de Java instalada (ejecute `java -version`).
3. Descripción detallada del problema.
4. Pasos para reproducir el error.

2.9. Actualizaciones futuras

2.9.1. Política de versiones

SimAS 3.0 es una versión estable y completa. Las futuras mejoras y nuevas funcionalidades se desarrollarán como nuevas versiones principales:

- **SimAS 4.0:** futuras mejoras y nuevas características.
- **SimAS 5.0:** versiones posteriores con funcionalidades avanzadas.
- **SimAS 6.0:** y así sucesivamente.

2.9.2. Información sobre nuevas versiones

Para estar informado sobre nuevas versiones:

1. Consulte el sitio web oficial del proyecto.
2. Revise las publicaciones académicas relacionadas.
3. Contacte con el desarrollador para información sobre futuras versiones.

2.9.3. Migración entre versiones

Al instalar una nueva versión principal:

1. Sus gramáticas guardadas serán compatibles con versiones futuras.
2. Los informes PDF generados se mantendrán accesibles.
3. No será necesario migrar configuraciones, ya que SimAS no utiliza archivos de configuración permanentes.

Capítulo 3

Características de la interfaz

3.1. Introducción a la interfaz

SimAS 3.0 presenta una interfaz moderna e intuitiva desarrollada con JavaFX [6] que facilita el aprendizaje y uso de la aplicación. La interfaz está diseñada para ser clara, accesible y funcional, permitiendo a los usuarios concentrarse en los conceptos de análisis sintáctico sin distracciones innecesarias.

La aplicación utiliza un sistema de pestañas que permite trabajar con múltiples gramáticas y simulaciones simultáneamente, mejorando significativamente la productividad del usuario.

3.2. Pantalla de bienvenida

Al iniciar SimAS 3.0, la primera pantalla que aparece es la pantalla de bienvenida, que incluye:

- **Logo de la aplicación:** identificación visual clara de SimAS 3.0.
- **Información de versión:** muestra la versión actual de la aplicación.
- **Credenciales del desarrollador:** información sobre el autor y director del proyecto.
- **Indicador de progreso:** muestra el estado de carga de los componentes de la aplicación.

Esta pantalla permanece visible durante unos segundos mientras la aplicación inicializa todos sus componentes y recursos necesarios.

3.3. Menú principal

Una vez completada la carga, la aplicación muestra el menú principal, que constituye el centro de navegación de SimAS 3.0.

3.3.1. Elementos del menú principal

El menú principal, mostrado en la figura 3.1, incluye los siguientes elementos:

- **Título de la aplicación:** "SimAS 3.0" en la parte superior.
- **Subtítulo:** "Simulador de Análisis Sintáctico" que describe la funcionalidad.
- **Botones de navegación:** acceso a las principales funcionalidades.
- **Selector de idioma:** en la esquina superior derecha.
- **Información del desarrollador:** en la parte inferior.

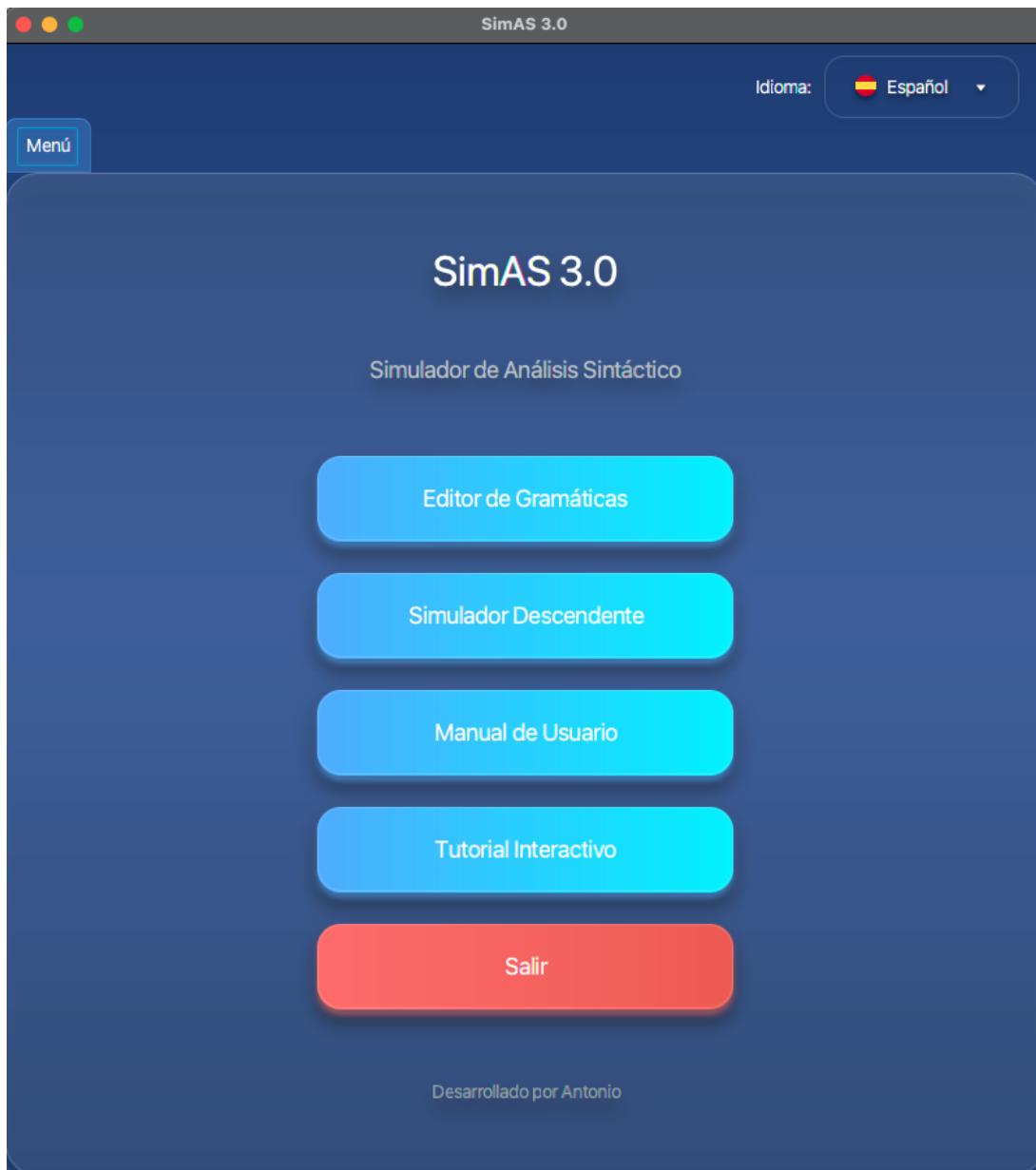


Figura 3.1: Menú principal de SimAS 3.0

3.3.2. Botones principales

Los botones del menú principal proporcionan acceso a las funcionalidades principales:

- **Editor de gramáticas:** abre el editor para crear y gestionar gramáticas.
- **Simulador descendente:** accede al simulador de análisis sintáctico.
- **Manual de usuario:** abre este manual de usuario.
- **Tutorial interactivo:** proporciona una guía paso a paso.
- **Sair:** cierra la aplicación.

3.4. Sistema de pestañas

SimAS 3.0 implementa un sistema de pestañas avanzado que permite trabajar con múltiples elementos simultáneamente.

3.4.1. Características del sistema de pestañas

- **Pestaña principal:** siempre visible, contiene el menú principal.
- **Pestañas dinámicas:** se crean automáticamente al abrir editores o simuladores.
- **Numeración automática:** las pestañas se numeran secuencialmente para facilitar la identificación.
- **Cierre individual:** cada pestaña puede cerrarse independientemente.
- **Reordenamiento:** las pestañas pueden arrastrarse para cambiar su orden.

3.4.2. Gestión de pestañas

- **Botón "Cerrar todas las pestañas":** cierra todas las pestañas excepto la principal.
- **Menú contextual:** clic derecho en una pestaña para opciones adicionales.
- **Separación de pestañas:** arrastrar una pestaña fuera del área principal crea una nueva ventana.
- **Reunión de ventanas:** arrastrar pestañas entre ventanas las consolida.
- **Grupos de pestañas:** las pestañas relacionadas (editor y simulador) se agrupan automáticamente.
- **Movimiento de grupos:** arrastrar una pestaña de un grupo mueve todo el grupo a una nueva ventana.
- **Numeración de grupos:** cada grupo de pestañas recibe un número único para identificación.

3.5. Selector de idioma

SimAS 3.0 soporta múltiples idiomas, permitiendo a los usuarios trabajar en su idioma preferido.

3.5.1. Idiomas disponibles

La aplicación incluye soporte para los siguientes idiomas:

- **Español:** idioma por defecto.
- **English:** inglés.
- **Français:** francés.
- **Português:** portugués.
- **Deutsch:** alemán.
- **Japanese:** japonés.

3.5.2. Cambio de idioma

Para cambiar el idioma de la interfaz:

1. Localice el selector de idioma en la esquina superior derecha de la ventana principal de la aplicación.
2. Haga clic en el menú desplegable.
3. Seleccione el idioma deseado de la lista.
4. La interfaz se actualizará inmediatamente al nuevo idioma.
5. El cambio se aplicará a todas las pestañas y ventanas abiertas.
6. El idioma seleccionado se mantiene durante toda la sesión de trabajo.

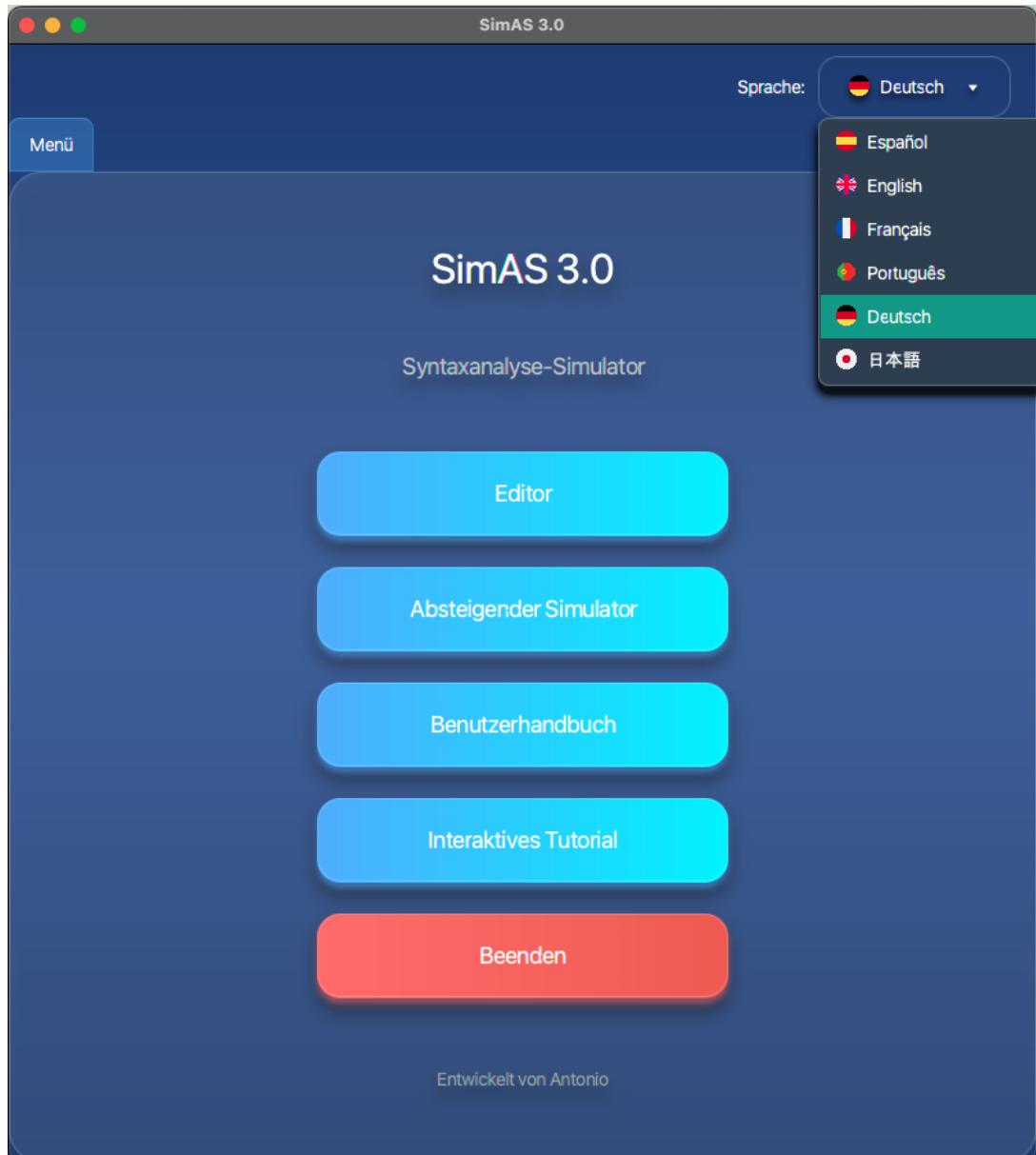


Figura 3.2: Selector de idiomas con menú desplegable

El selector de idiomas, ilustrado en la figura 3.2, muestra todos los idiomas disponibles en un menú desplegable fácil de usar.

3.6. Atajos de teclado

SimAS 3.0 incluye atajos de teclado para facilitar el acceso rápido a las funciones principales.

3.6.1. Atajos disponibles

SimAS 3.0 incluye los siguientes atajos de teclado:

- **Ctrl+N:** abre el editor de gramáticas.
- **Ctrl+S:** abre el simulador descendente.
- **Ctrl+H:** abre el manual de usuario.
- **Ctrl+W:** cierra la pestaña actual.
- **Ctrl+Shift+W:** cierra todas las pestañas.
- **Ctrl+Tab:** navega a la siguiente pestaña.
- **Ctrl+Shift+Tab:** navega a la pestaña anterior.
- **Ctrl+T:** crea una nueva pestaña del editor.
- **Ctrl+Shift+T:** crea una nueva ventana.
- **Ctrl+1-9:** navega directamente a la pestaña numerada.

3.6.2. Uso de atajos

Los atajos de teclado funcionan desde cualquier pestaña activa y proporcionan acceso rápido a las funcionalidades más utilizadas, mejorando la eficiencia del usuario.

3.7. Elementos visuales

La interfaz de SimAS 3.0 utiliza elementos visuales consistentes para mejorar la experiencia del usuario.

3.7.1. Iconos y gráficos

- **Iconos descriptivos:** cada botón principal incluye un ícono que representa su función.
- **Banderas de países:** el selector de idioma muestra banderas para identificación visual.
- **Indicadores de estado:** elementos visuales que muestran el estado de las operaciones.
- **Colores consistentes:** paleta de colores coherente en toda la aplicación.

3.7.2. Tipografía y espaciado

- **Fuentes legibles:** tipografía clara y fácil de leer.
- **Espaciado apropiado:** elementos bien distribuidos para evitar saturación visual.
- **Jerarquía visual:** diferentes tamaños de fuente para establecer importancia.
- **Contraste adecuado:** colores que facilitan la lectura.

3.8. Responsividad de la interfaz

La interfaz de SimAS 3.0 está diseñada para adaptarse a diferentes tamaños de pantalla y resoluciones.

3.8.1. Adaptación a la pantalla

- **Tamaño mínimo:** la ventana puede redimensionarse hasta 600x700 píxeles.
- **Tamaño por defecto:** 800x900 píxeles para una experiencia óptima.
- **Escalado automático:** los elementos se adaptan al tamaño de la ventana.
- **Barras de desplazamiento:** aparecen automáticamente cuando es necesario.

3.8.2. Optimización para diferentes resoluciones

La aplicación se optimiza automáticamente para:

- **Resoluciones estándar:** 1024x768, 1366x768, 1920x1080.
- **Pantallas de alta densidad:** soporte para pantallas Retina y similares.
- **Orientaciones:** funciona correctamente en orientación horizontal.
- **Proporciones:** se adapta a diferentes relaciones de aspecto.

3.9. Accesibilidad

SimAS 3.0 incluye características de accesibilidad para facilitar su uso por parte de todos los usuarios.

3.9.1. Características de accesibilidad

- **Atajos de teclado:** alternativa al uso del ratón.
- **Contraste visual:** colores que facilitan la distinción de elementos.
- **Tamaño de fuente:** texto legible en diferentes tamaños de pantalla.
- **Navegación por teclado:** todos los elementos son accesibles mediante teclado.

3.9.2. Compatibilidad con tecnologías asistivas

La aplicación es compatible con:

- **Amplificadores de pantalla:** elementos visuales claros y contrastados.
- **Dispositivos de entrada alternativos:** soporte para diferentes tipos de entrada (ratón, teclado, trackpad).
- **Software de control por voz:** compatible con sistemas de control por voz del sistema operativo.
- **Dispositivos de entrada adaptativos:** soporte para dispositivos especializados de accesibilidad.

3.10. Personalización de la interfaz

Aunque SimAS 3.0 mantiene una apariencia consistente, los usuarios pueden personalizar algunos aspectos de la experiencia.

3.10.1. Opciones de personalización

- **Idioma:** selección del idioma de la interfaz.
- **Tamaño de ventana:** redimensionamiento según preferencias.
- **Organización de pestañas:** reordenamiento y agrupación.
- **Atajos de teclado:** uso opcional de atajos para funciones rápidas.

3.10.2. Limitaciones de personalización

Por motivos de consistencia y funcionalidad, algunos elementos no son personalizables:

- **Colores de la interfaz:** paleta fija para mantener coherencia visual.

- **Tipografía:** fuente estándar para garantizar legibilidad.
- **Disposición de elementos:** estructura fija para optimizar el flujo de trabajo.
- **Iconos:** conjunto estándar para facilitar el reconocimiento.

3.11. Integración entre componentes

La interfaz de SimAS 3.0 está diseñada para facilitar la integración entre diferentes componentes de la aplicación.

3.11.1. Flujo de trabajo integrado

- **Editor a simulador:** transición directa desde el editor al simulador.
- **Sincronización de datos:** las gramáticas se comparten automáticamente entre componentes.
- **Estado consistente:** cambios en un componente se reflejan en otros.
- **Navegación fluida:** transiciones suaves entre diferentes funcionalidades.

3.11.2. Comunicación entre ventanas

- **Ventanas secundarias:** creación de ventanas adicionales para trabajo paralelo.
- **Sincronización de idioma:** cambios de idioma se aplican a todas las ventanas.
- **Gestión de estado:** estado compartido entre ventanas relacionadas.
- **Cierre coordinado:** cierre apropiado de ventanas dependientes.

3.12. Solución de problemas de interfaz

Si experimenta problemas con la interfaz de SimAS 3.0, considere las siguientes soluciones:

3.12.1. Problemas comunes

- **Interfaz no se muestra correctamente:** verifique la resolución de pantalla y actualice los controladores gráficos.
- **Elementos superpuestos:** redimensione la ventana o cambie la resolución de pantalla.
- **Texto borroso:** ajuste la configuración de escalado de Windows/macOS.

- **Atajos de teclado no funcionan:** asegúrese de que la ventana esté activa y enfocada.

3.12.2. Optimización del rendimiento

Para obtener el mejor rendimiento de la interfaz:

- **Cierre pestañas innecesarias:** mantenga solo las pestañas que esté utilizando.
- **Resolución apropiada:** use una resolución de pantalla compatible.
- **Memoria disponible:** asegúrese de tener suficiente memoria RAM libre.
- **Controladores actualizados:** mantenga actualizados los controladores gráficos.

3.13. Ventanas secundarias

SimAS 3.0 permite crear ventanas secundarias para trabajar con múltiples proyectos simultáneamente, mejorando significativamente la productividad del usuario.

3.13.1. Creación de ventanas secundarias

Las ventanas secundarias se pueden crear de las siguientes maneras:

- **Arrastrando pestañas:** arrastrar una pestaña fuera del área principal crea automáticamente una nueva ventana.
- **Menú contextual:** clic derecho en una pestaña y seleccionar "Nueva ventana".

3.13.2. Características de las ventanas secundarias

Cada ventana secundaria mantiene las siguientes características:

- **Independencia:** cada ventana funciona de manera independiente.
- **Sincronización de idioma:** cambios de idioma se aplican a todas las ventanas.
- **Gestión de pestañas:** cada ventana tiene su propio sistema de pestañas.
- **Estado persistente:** el estado de cada ventana se mantiene durante la sesión.
- **Comunicación entre ventanas:** las ventanas pueden compartir datos cuando es necesario.

3.13.3. Gestión de múltiples ventanas

Para gestionar eficientemente múltiples ventanas:

- **Organización:** cada ventana puede contener un proyecto diferente.
- **Intercambio de datos:** arrastrar pestañas entre ventanas para reorganizar el trabajo.
- **Cierre coordinado:** cerrar una ventana principal cierra todas las ventanas secundarias.

3.14. Barra de estado y notificaciones

SimAS 3.0 incluye una barra de estado que proporciona información importante sobre el estado de la aplicación y las operaciones en curso.

3.14.1. Elementos de la barra de estado

La barra de estado incluye:

- **Mensajes informativos:** notificaciones sobre el estado del sistema.
- **Contador de pestañas:** número total de pestañas abiertas.
- **Indicador de idioma:** idioma actualmente seleccionado.

3.14.2. Sistema de notificaciones

El sistema de notificaciones proporciona:

- **Notificaciones de éxito:** confirmación de operaciones completadas.
- **Advertencias:** alertas sobre posibles problemas.
- **Errores:** información detallada sobre errores encontrados.
- **Información contextual:** ayuda específica según la situación.

3.15. Menús contextuales

SimAS 3.0 implementa menús contextuales inteligentes que se adaptan al contexto actual.

3.15.1. Menús contextuales de pestañas

Al hacer clic derecho en una pestaña, se muestra un menú con opciones como:

- **Cerrar pestaña:** cierra la pestaña actual.
- **Cerrar todas las pestañas:** cierra todas las pestañas de la ventana actual.
- **Abrir en nueva ventana:** abre la pestaña en una nueva ventana.
- **Abrir en ventana existente:** abre la pestaña en una ventana existente.

3.16. Conclusión

La interfaz de SimAS 3.0 está diseñada para proporcionar una experiencia de usuario intuitiva, eficiente y profesional. Con su sistema de pestañas avanzado, soporte multiidioma completo, atajos de teclado extensivos, características de accesibilidad robustas e integración profunda con el sistema operativo, la aplicación facilita el aprendizaje y uso de los conceptos de análisis sintáctico descendente predictivo.

La capacidad de trabajar con múltiples elementos simultáneamente, la gestión inteligente de ventanas secundarias, los menús contextuales adaptativos y la configuración avanzada hacen de SimAS 3.0 una herramienta poderosa, flexible y profesional para estudiantes, profesores e investigadores en el campo de la informática y la teoría de lenguajes formales.

La interfaz no solo cumple con los estándares de usabilidad modernos, sino que también establece nuevas referencias en el campo de las aplicaciones educativas especializadas, proporcionando una experiencia de usuario a la altura de las mejores herramientas profesionales del mercado.

Capítulo 4

Editor de gramáticas

El editor de gramáticas es el componente central de SimAS 3.0, diseñado para facilitar la creación, edición y gestión de gramáticas libres de contexto en formato BNF [7]. Este editor proporciona un entorno intuitivo y guiado que permite a los usuarios, desde principiantes hasta expertos, crear gramáticas complejas de manera eficiente y sin errores.

4.1. Introducción al editor

El editor de gramáticas de SimAS 3.0 está construido sobre principios de usabilidad y pedagogía, ofreciendo una experiencia de aprendizaje progresiva que guía al usuario a través del proceso de creación de gramáticas mediante un asistente paso a paso.

4.1.1. Características principales

El editor incluye las siguientes características principales:

- **Asistente guiado:** proceso de creación en 4 pasos con validación al finalizar.
- **Validación automática:** verificación de gramáticas cargadas desde archivo y al finalizar el proceso de creación/edición.
- **Validación manual:** el usuario puede validar manualmente al finalizar el proceso de creación/edición.
- **Formato BNF:** soporte completo para el formato de notación Backus-Naur.
- **Interfaz intuitiva:** diseño claro y organizado que facilita la comprensión.
- **Gestión de archivos:** guardado y carga de gramáticas en formato XML.
- **Integración con simulador:** transición directa al simulador para probar la gramática.

4.1.2. Objetivos del editor

Los objetivos principales del editor son:

- **Educativo:** proporcionar una herramienta de aprendizaje para estudiantes de informática.
- **Profesional:** ofrecer funcionalidades avanzadas para usuarios experimentados.
- **Validación:** asegurar la corrección sintáctica y semántica de las gramáticas.
- **Eficiencia:** acelerar el proceso de creación y edición de gramáticas.

4.2. Acceso al editor

El editor de gramáticas se puede acceder de múltiples maneras desde la interfaz principal de SimAS 3.0.

4.2.1. Métodos de acceso

Para abrir el editor de gramáticas:

1. **Desde el menú principal:** haga clic en el botón "Editor de gramáticas".
2. **Atajo de teclado:** presione **Ctrl+N** desde cualquier ventana.

4.3. Interfaz del editor

Al abrir el editor, se presenta una interfaz organizada que muestra toda la información relacionada con la gramática actual.

4.3.1. Estado vacío del editor

Cuando se abre el editor por primera vez o no hay ninguna gramática cargada, la interfaz se muestra vacía como se puede ver en la figura 4.1.

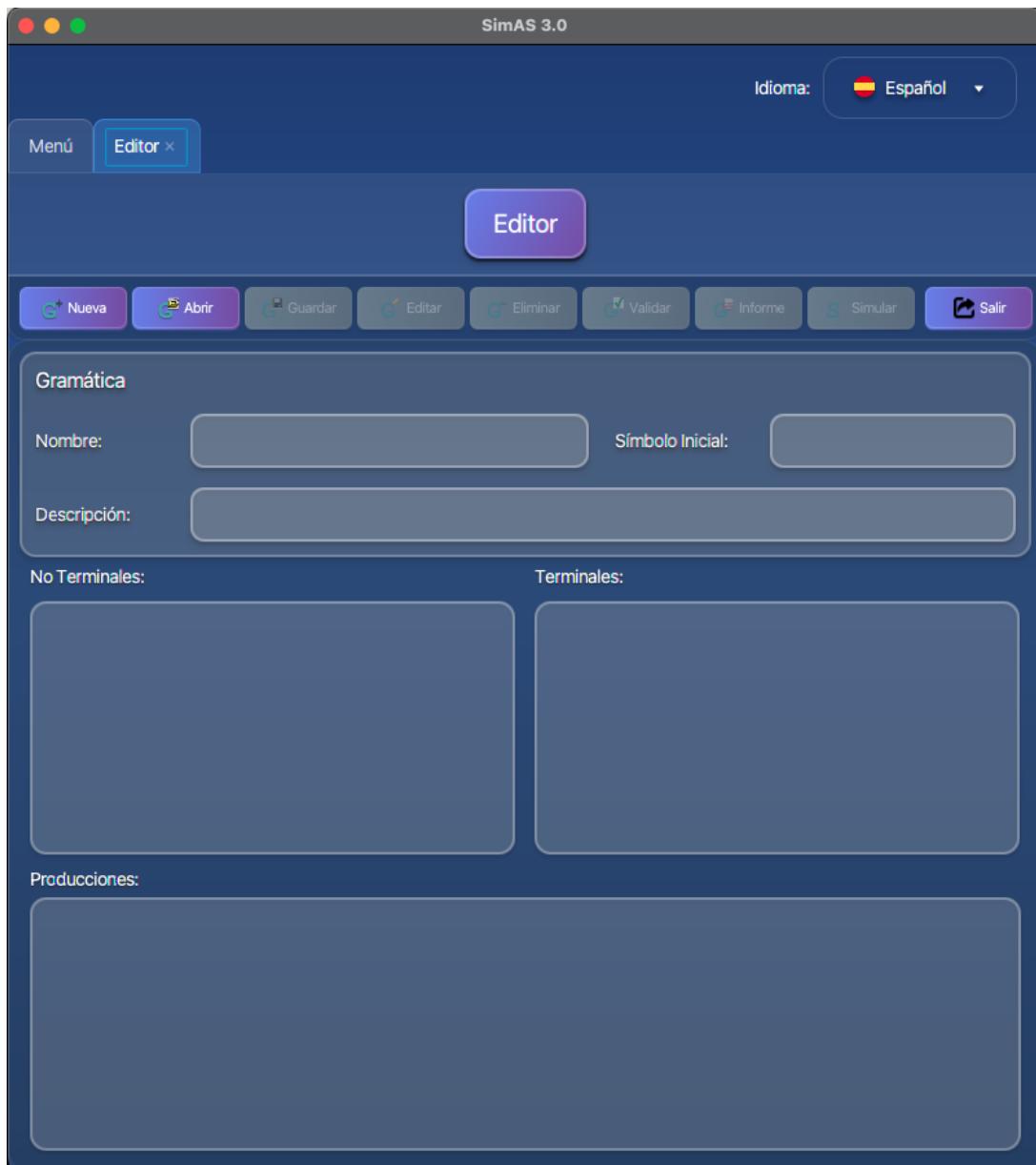


Figura 4.1: Editor de gramáticas en estado vacío

En este estado, la interfaz muestra:

- **Barra de herramientas:** con los botones "Nueva", "Abrir" y "Salir" habilitados, y el resto deshabilitados.
- **Área principal:** completamente vacía, lista para mostrar la información de una gramática.
- **Información de la gramática:** campos vacíos para nombre, descripción, símbolos y producciones.

4.3.2. Estado con gramática cargada

Cuando se carga una gramática existente, la interfaz se actualiza automáticamente para mostrar toda la información de la gramática, como se muestra en la figura 4.2.

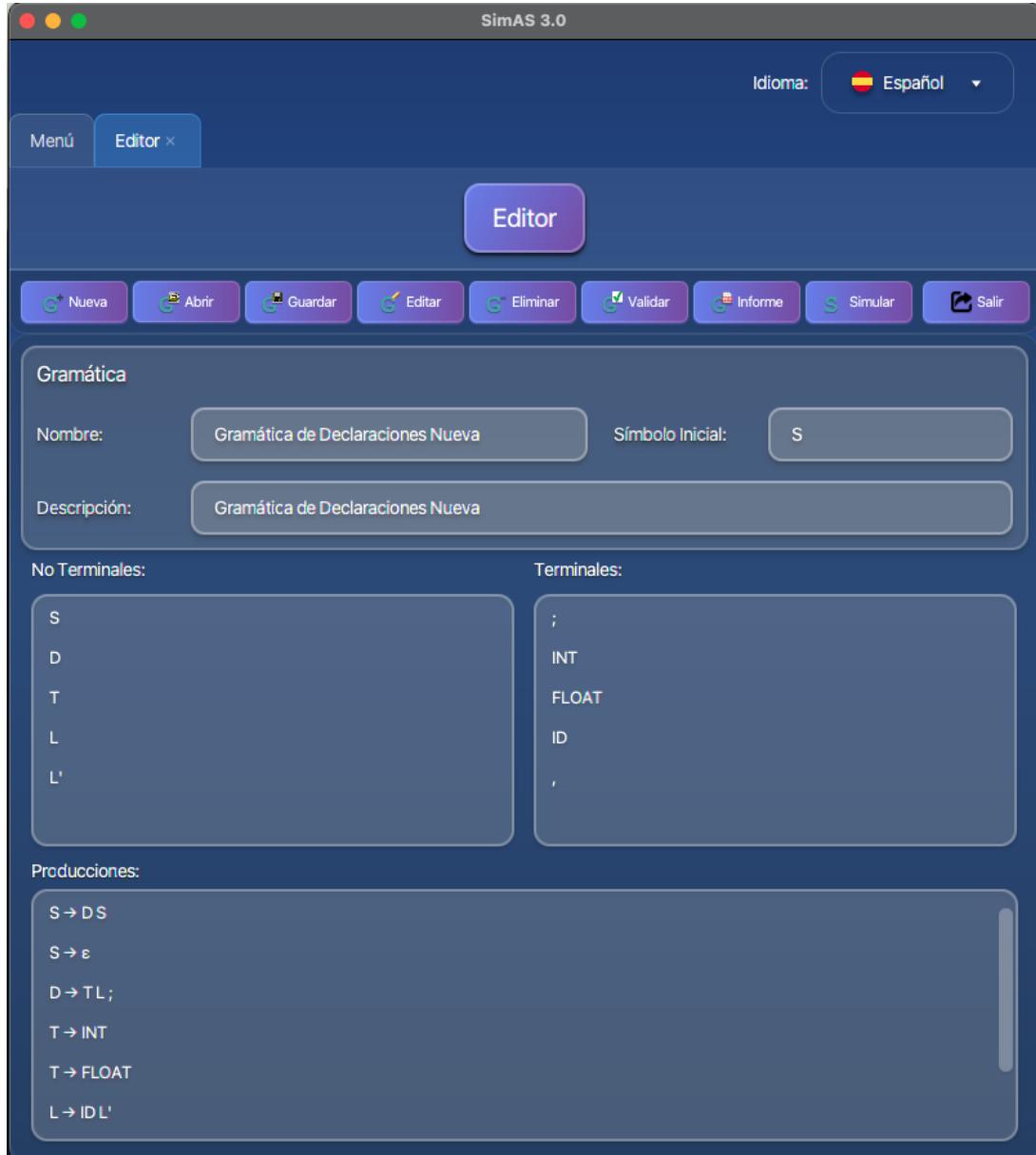


Figura 4.2: Editor de gramáticas con una gramática cargada

En este estado, la interfaz muestra:

- **Barra de herramientas:** con todos los botones habilitados.
- **Información de la gramática:** nombre, descripción, símbolos terminales, símbolos no terminales y producciones.

4.3.3. Barra de herramientas

La barra de herramientas del editor incluye los siguientes botones:

- **Botón "Nueva"**: abre el asistente de creación para crear una nueva gramática desde cero.
- **Botón "Abrir"**: carga una gramática existente desde un archivo XML.
- **Botón "Guardar"**: guarda la gramática actual en un archivo XML (solo disponible cuando hay una gramática cargada).
- **Botón "Editar"**: abre el asistente de edición con la gramática actual cargada (solo disponible cuando hay una gramática cargada).
- **Botón "Eliminar"**: elimina la gramática actual del editor (solo disponible cuando hay una gramática cargada).
- **Botón "Validar"**: ejecuta la validación de la gramática actual (solo disponible cuando hay una gramática cargada).
- **Botón "Informe"**: genera un informe detallado de la gramática actual (solo disponible cuando hay una gramática cargada).
- **Botón "Simular"**: transfiere la gramática al simulador para su análisis (solo disponible cuando hay una gramática cargada).
- **Botón "Salir"**: cierra la aplicación.

4.4. Asistente guiado de creación

El asistente guiado es una de las características más destacadas del editor, diseñado para simplificar el proceso de creación y edición de gramáticas mediante un enfoque paso a paso.

4.4.1. Acceso al asistente

El asistente se puede abrir de dos maneras:

- **Botón "Nueva"**: abre el asistente vacío para crear una nueva gramática desde cero.
- **Botón "Editar"**: abre el asistente con la gramática actual cargada para editarla (solo disponible cuando hay una gramática cargada).

4.4.2. Proceso de creación

El asistente divide la creación/edición de gramáticas en cuatro pasos lógicos:

1. **Paso 1 - Datos:** definición del nombre y descripción de la gramática.
2. **Paso 2 - Símbolos:** definición de símbolos terminales y no terminales.
3. **Paso 3 - Producciones:** creación de las reglas de producción.
4. **Paso 4 - Inicial:** selección del símbolo inicial y finalización.

4.4.3. Navegación del asistente

El asistente incluye un menú inferior que permite:

- **Navegación entre pasos:** avanzar y retroceder entre los diferentes pasos.
- **Validación de pasos:** no se permite continuar al siguiente paso si el actual está vacío.
- **Finalización:** al completar el paso 4, los cambios se reflejan en el editor principal.

4.4.4. Paso 1: Datos de la gramática

En el primer paso, el usuario define los datos básicos de la gramática: nombre y descripción.

4.4.4.1. Interfaz del paso 1

La interfaz del paso 1 se muestra en la figura 4.3.

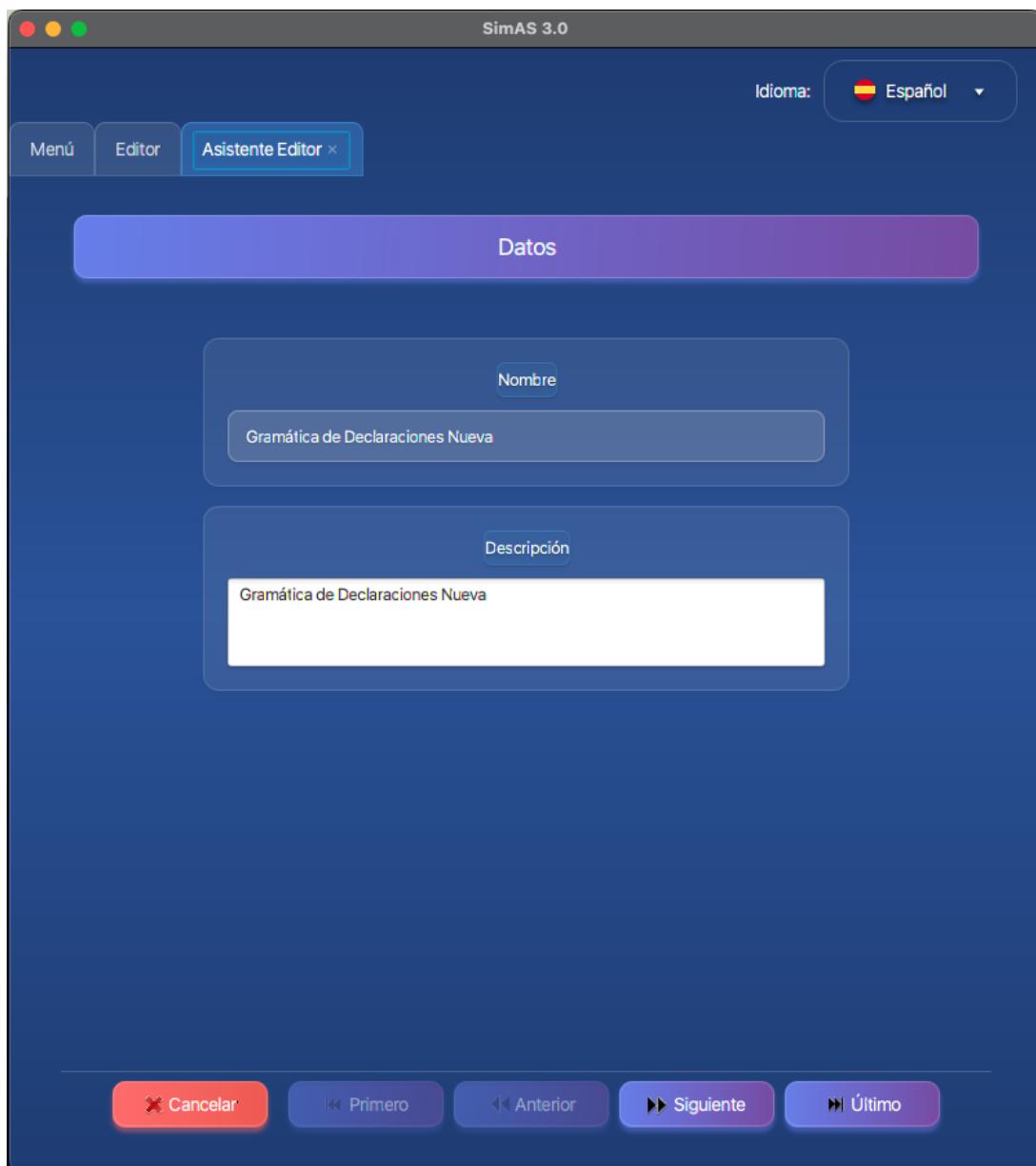


Figura 4.3: Paso 1: Datos de la gramática

4.4.4.2. Campos del paso 1

El paso 1 incluye los siguientes campos:

- **Nombre de la gramática:** campo de texto para introducir el nombre que identificará la gramática.
- **Descripción:** campo de texto para proporcionar una descripción detallada de la gramática.

4.4.4.3. Estado vacío del paso 1

Si el paso 1 está vacío, se muestra la interfaz de la figura 4.4 y no se permite continuar al siguiente paso.

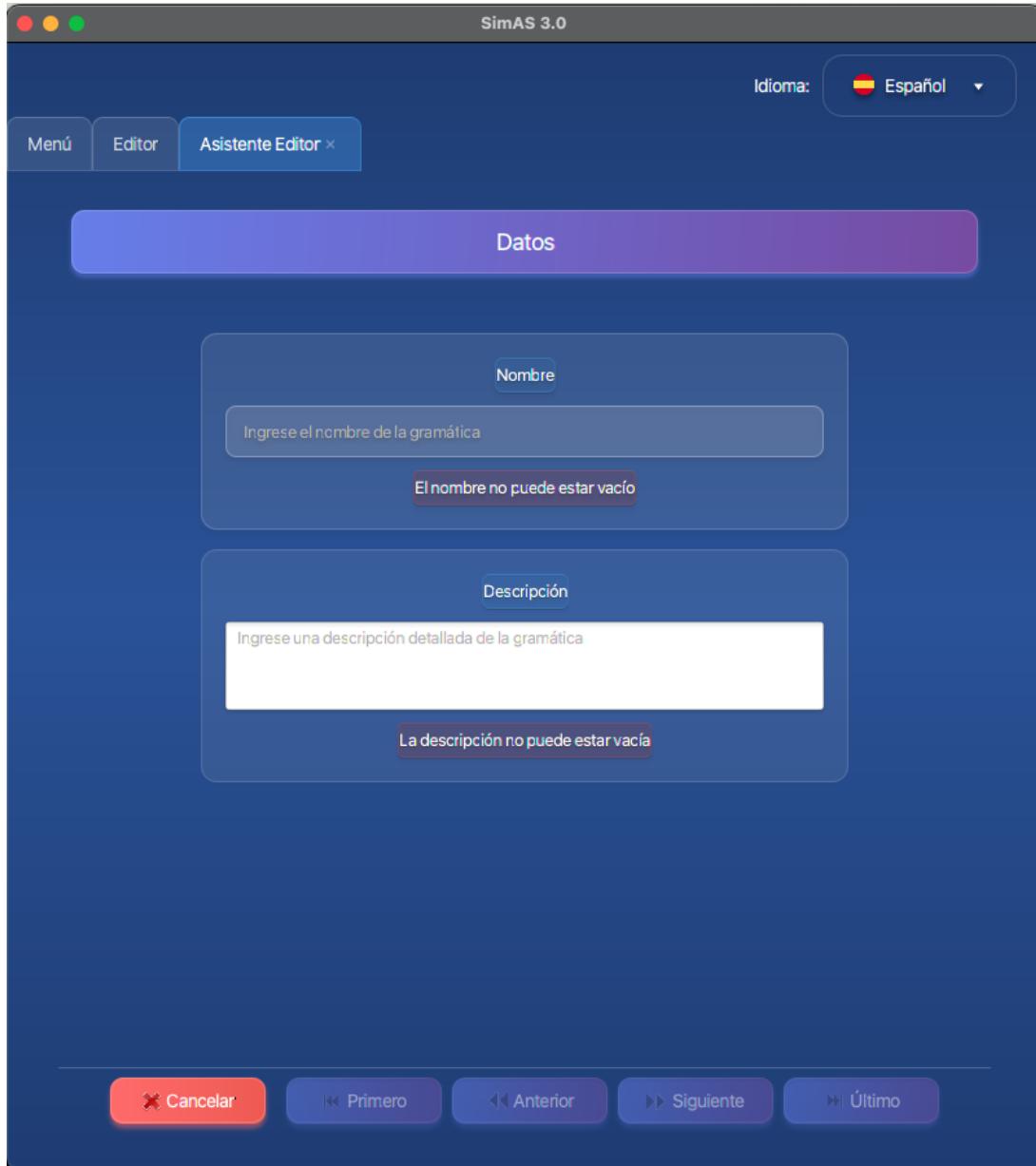


Figura 4.4: Paso 1: Estado vacío

4.4.4.4. Requisitos del paso 1

Para poder continuar al paso 2, es necesario:

- **Nombre obligatorio:** el campo nombre debe estar relleno.
- **Descripción obligatoria:** el campo descripción debe estar rellenado.

4.4.5. Paso 2: Símbolos

En el segundo paso, el usuario define tanto los símbolos terminales como los no terminales de la gramática.

4.4.5.1. Interfaz del paso 2

La interfaz del paso 2 se muestra en la figura 4.5.

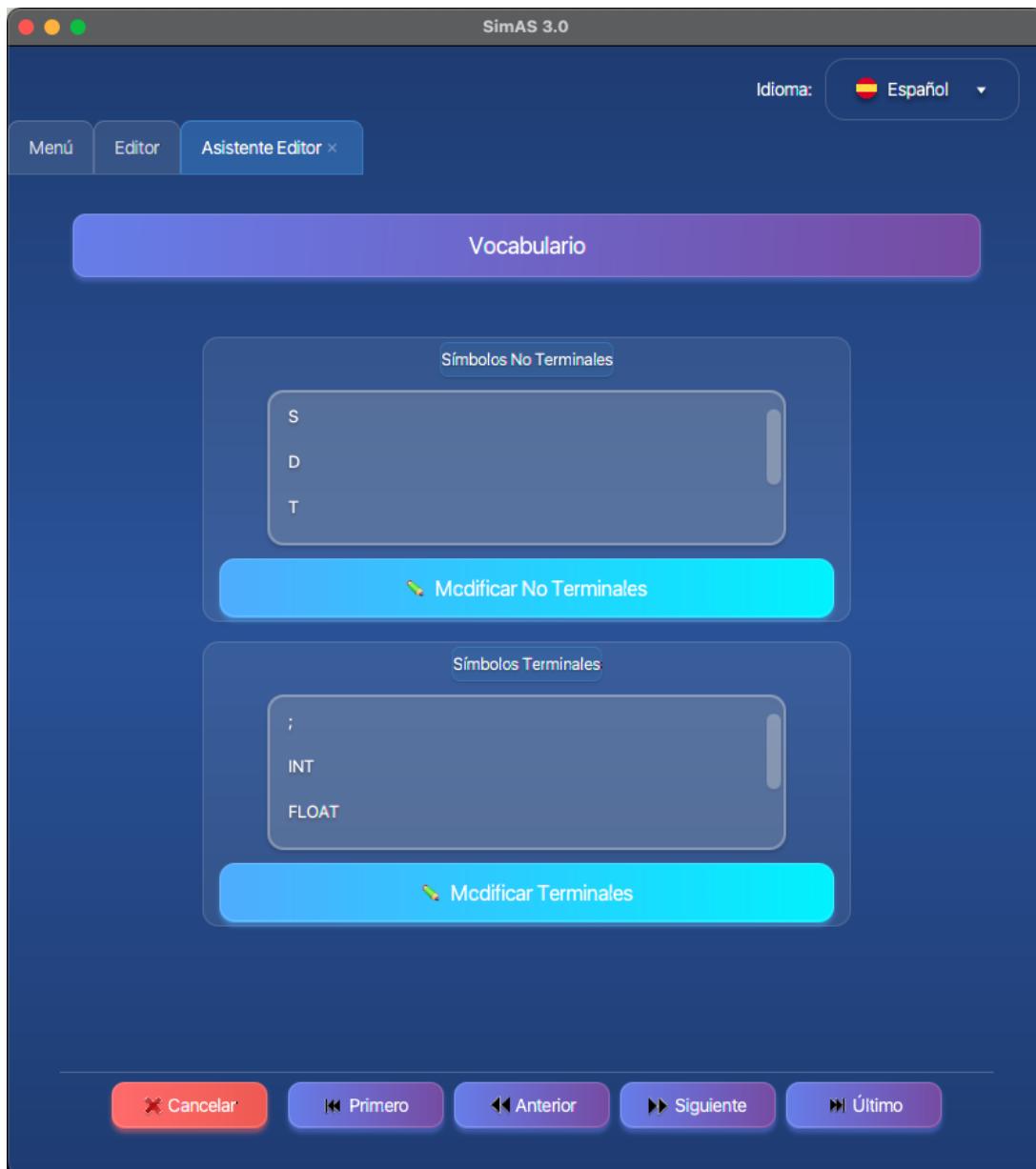


Figura 4.5: Paso 2: Símbolos de la gramática

4.4.5.2. Estado vacío del paso 2

Si el paso 2 está vacío, se muestra la interfaz de la figura 4.6 y no se permite continuar al siguiente paso.

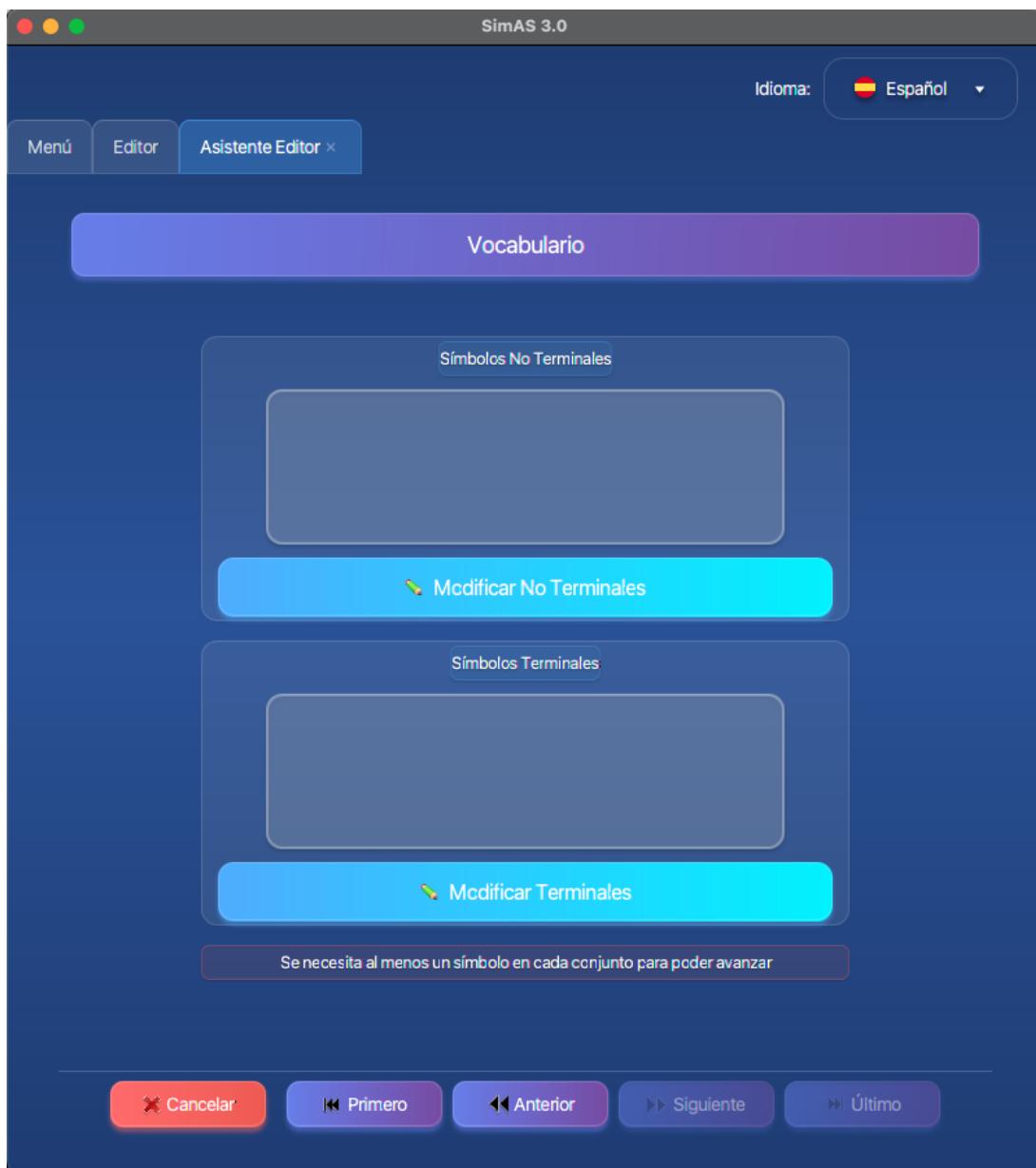


Figura 4.6: Paso 2: Estado vacío

4.4.5.3. Elementos del paso 2

El paso 2 incluye los siguientes elementos:

- **Lista de símbolos terminales:** muestra todos los símbolos terminales definidos.
- **Lista de símbolos no terminales:** muestra todos los símbolos no terminales definidos.
- **Botón "Modificar símbolos terminales":** abre el panel auxiliar para gestionar símbolos terminales.
- **Botón "Modificar símbolos no terminales":** abre el panel auxiliar para gestionar símbolos no terminales.

- **Indicador de estado:** muestra si el paso está incompleto.

4.4.5.4. Paneles auxiliares

El paso 2 permite abrir dos paneles auxiliares para la gestión de símbolos:

- **Panel de símbolos terminales:** para insertar, editar o eliminar símbolos terminales.
- **Panel de símbolos no terminales:** para insertar, editar o eliminar símbolos no terminales.

4.4.5.5. Panel de símbolos terminales

Al hacer clic en el botón correspondiente, se abre el panel de símbolos terminales mostrado en la figura 4.7.

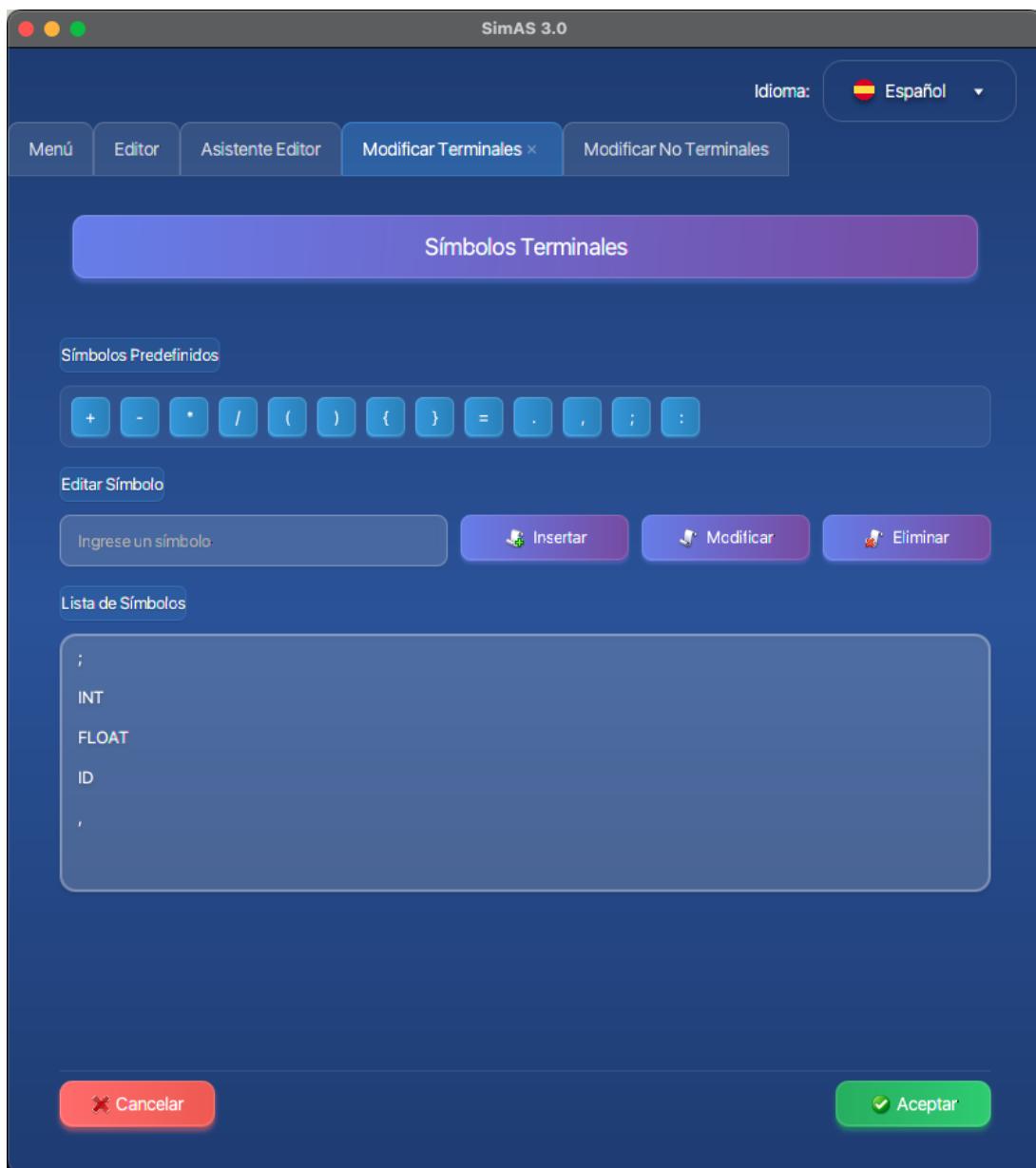


Figura 4.7: Panel de símbolos terminales

4.4.5.6. Elementos del panel de símbolos terminales

Este panel incluye los siguientes elementos:

- **Símbolos predefinidos:** lista de símbolos comunes que se pueden insertar rápidamente.
- **Campo de texto:** para introducir nuevos símbolos terminales.
- **Botón "Insertar":** añade el símbolo del campo de texto a la lista.
- **Botón "Modificar":** permite modificar un símbolo seleccionado.
- **Botón "Eliminar":** quita el símbolo seleccionado de la lista.

- **Botón "Aceptar"**: confirma los cambios y cierra el panel.
- **Botón "Cancelar"**: descarta los cambios y cierra el panel.

4.4.5.7. Funcionalidades del panel de símbolos terminales

Este panel permite:

- **Insertar terminales**: introducir nuevos símbolos terminales.
- **Modificar terminales**: editar símbolos terminales existentes.
- **Eliminar terminales**: quitar símbolos terminales de la gramática.
- **Usar símbolos predefinidos**: seleccionar símbolos comunes de una lista.

4.4.5.8. Panel de símbolos no terminales

Al hacer clic en el botón correspondiente, se abre el panel de símbolos no terminales mostrado en la figura 4.8.

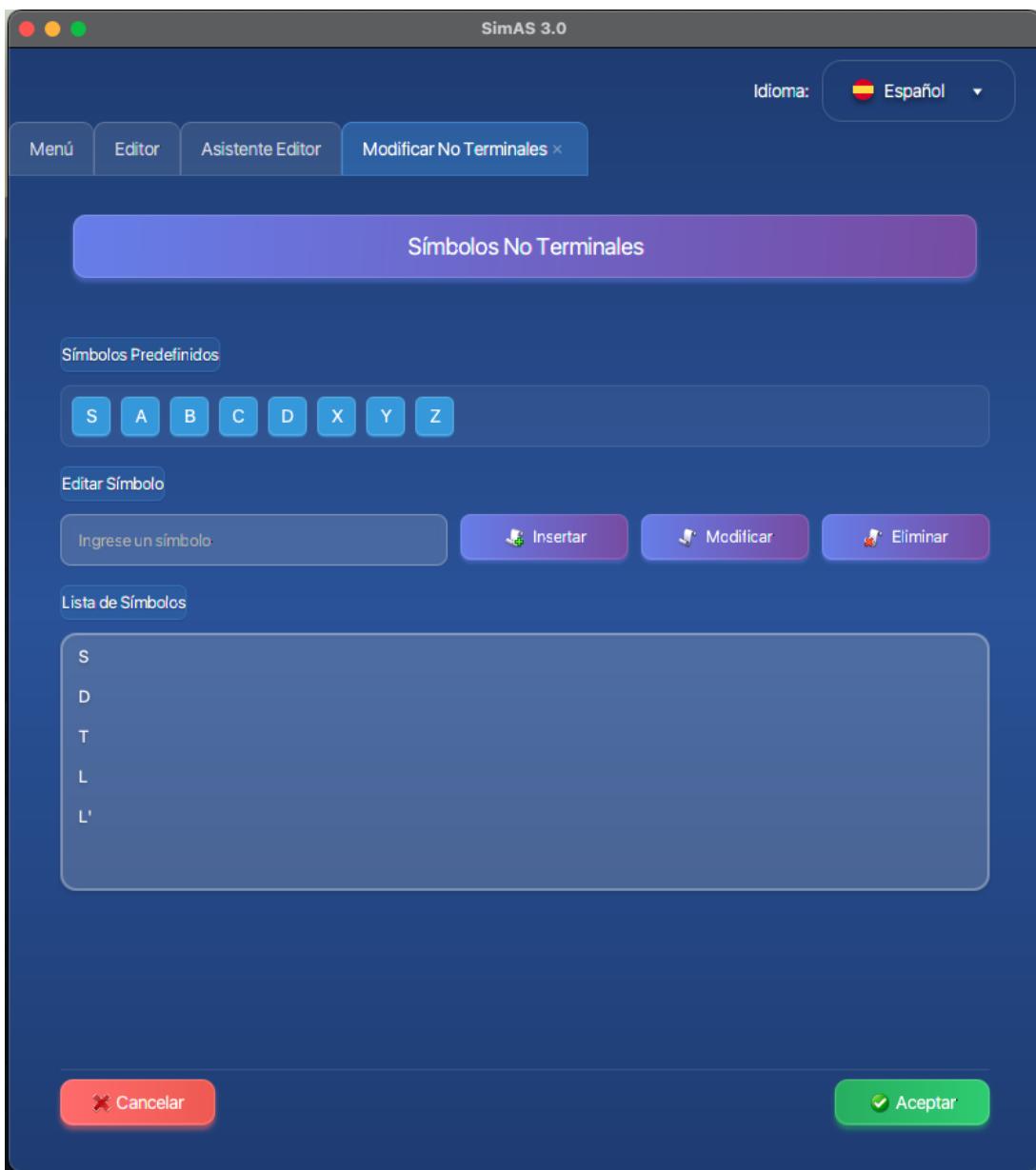


Figura 4.8: Panel de símbolos no terminales

4.4.5.9. Elementos del panel de símbolos no terminales

Este panel incluye los siguientes elementos:

- **Símbolos predefinidos:** lista de símbolos comunes que se pueden insertar rápidamente.
- **Campo de texto:** para introducir nuevos símbolos no terminales.
- **Botón "Insertar":** añade el símbolo del campo de texto a la lista.
- **Botón "Modificar":** permite editar un símbolo seleccionado.
- **Botón "Eliminar":** quita el símbolo seleccionado de la lista.

- **Botón "Aceptar"**: confirma los cambios y cierra el panel.
- **Botón "Cancelar"**: descarta los cambios y cierra el panel.

4.4.5.10. Funcionalidades del panel de símbolos no terminales

Este panel permite:

- **Insertar no terminales**: introducir nuevos símbolos no terminales.
- **Modificar no terminales**: editar símbolos no terminales existentes.
- **Eliminar no terminales**: quitar símbolos no terminales de la gramática.
- **Usar símbolos predefinidos**: seleccionar símbolos comunes de una lista.

4.4.5.11. Requisitos del paso 2

Para poder continuar al paso 3, es necesario:

- **Al menos un símbolo**: debe haber al menos un símbolo terminal y no terminal definido.
- **Símbolos únicos**: no puede haber símbolos duplicados.

4.4.6. Paso 3: Producciones

En el tercer paso, el usuario define las reglas de producción que relacionan los símbolos no terminales con secuencias de símbolos.

4.4.6.1. Interfaz del paso 3

La interfaz del paso 3 se muestra en la figura 4.9.

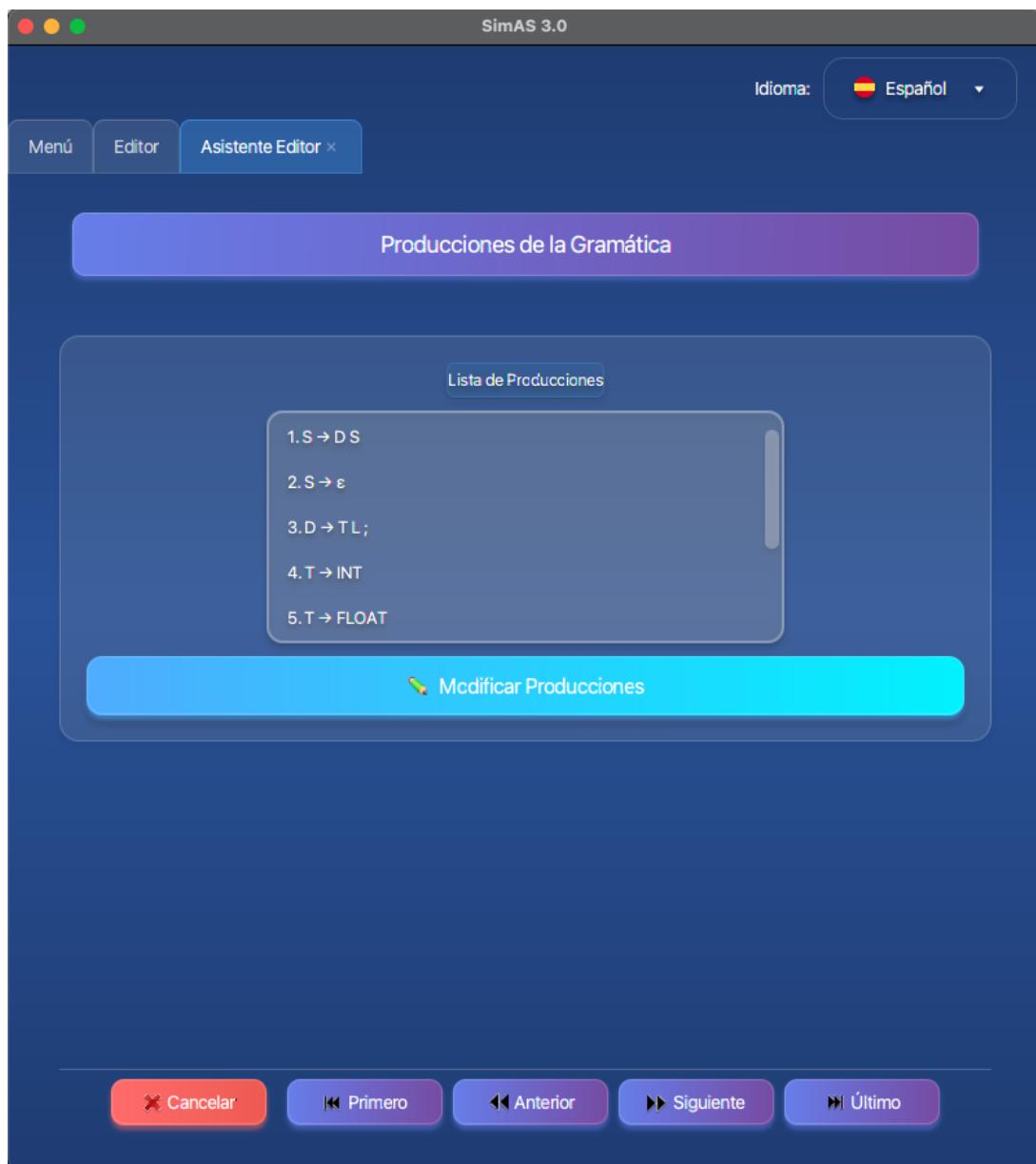


Figura 4.9: Paso 3: Producciones de la gramática

4.4.6.2. Estado vacío del paso 3

Si el paso 3 está vacío, se muestra la interfaz de la figura 4.10 y no se permite continuar al siguiente paso.

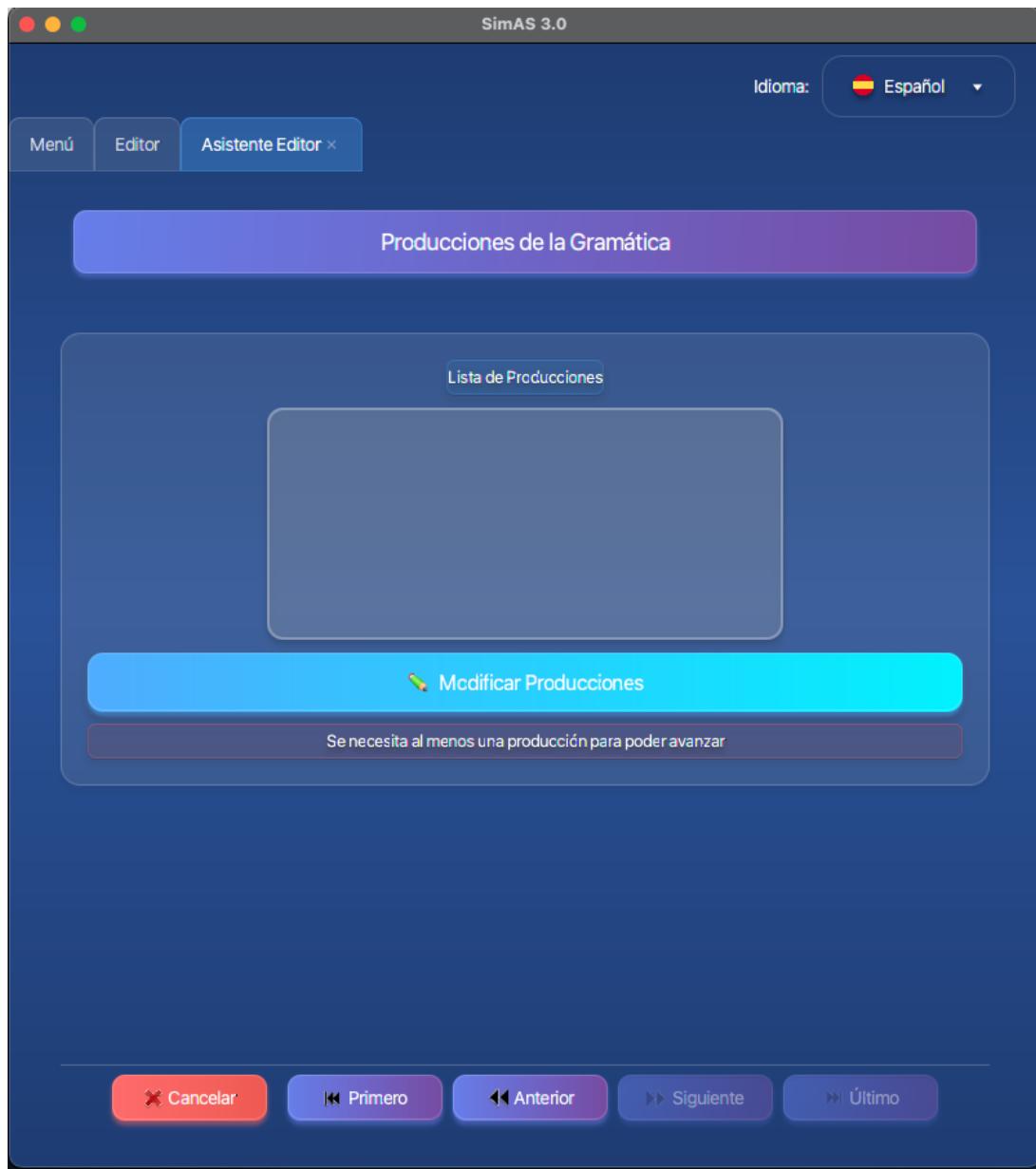


Figura 4.10: Paso 3: Estado vacío

4.4.6.3. Elementos del paso 3

El paso 3 incluye los siguientes elementos:

- **Lista de producciones:** muestra todas las reglas de producción definidas.
- **Botón "Modificar producciones":** abre el panel auxiliar para gestionar producciones.
- **Indicador de estado:** muestra si el paso está incompleto.

4.4.6.4. Panel de producciones

Al hacer clic en el botón correspondiente, se abre el panel de producciones mostrado en la figura 4.11.

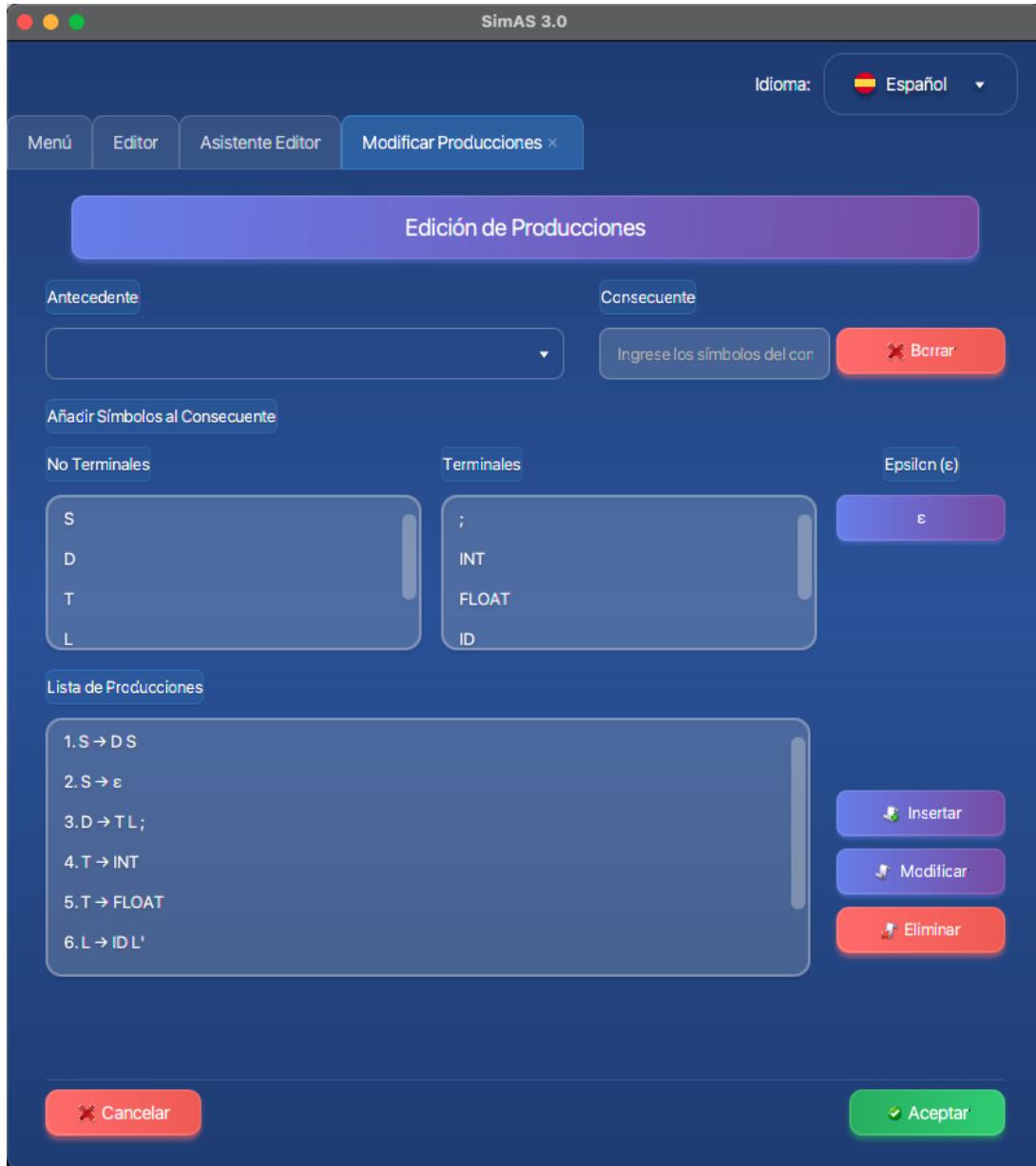


Figura 4.11: Panel de producciones

4.4.6.5. Elementos del panel de producciones

Este panel incluye los siguientes elementos:

- **Lista de producciones:** muestra todas las reglas de producción definidas.
- **Selector de símbolo no terminal:** permite elegir el símbolo no terminal (antecedente) para la producción.

- **Editor de consecuente:** área para construir la secuencia de símbolos del consecuente.
- **Lista de símbolos disponibles:** muestra los símbolos terminales y no terminales que se pueden usar.
- **Botón "Insertar":** añade la producción construida a la lista.
- **Botón "Modificar":** permite editar una producción seleccionada.
- **Botón "Eliminar":** quita la producción seleccionada de la lista.
- **Botón "Aceptar":** confirma los cambios y cierra el panel.
- **Botón "Cancelar":** descarta los cambios y cierra el panel.
- **Vista previa:** muestra cómo se verá la producción en formato BNF.

4.4.6.6. Funcionalidades del panel de producciones

Este panel permite:

- **Insertar producciones:** crear nuevas reglas de producción.
- **Modificar producciones:** editar producciones existentes.
- **Eliminar producciones:** quitar producciones de la gramática.
- **Construir consecuentes:** doble clic en los símbolos disponibles para crear secuencias.
- **Vista previa en tiempo real:** ver el formato BNF mientras se construye.

4.4.6.7. Formato de producciones

Las producciones siguen el formato estándar BNF:

- **Antecedente:** símbolo no terminal a la izquierda.
- **Flecha:** símbolo `::=` que indica "se define como".
- **Consecuente:** secuencia de símbolos a la derecha.

4.4.6.8. Requisitos del paso 3

Para poder continuar al paso 4, es necesario:

- **Al menos una producción:** debe haber al menos una regla de producción definida.
- **Producciones válidas:** todas las producciones deben estar bien formadas.

4.4.7. Paso 4: Símbolo inicial

En el cuarto paso, el usuario selecciona el símbolo inicial de la gramática y finaliza el proceso de creación/edición.

4.4.7.1. Interfaz del paso 4

La interfaz del paso 4 se muestra en la figura 4.12.

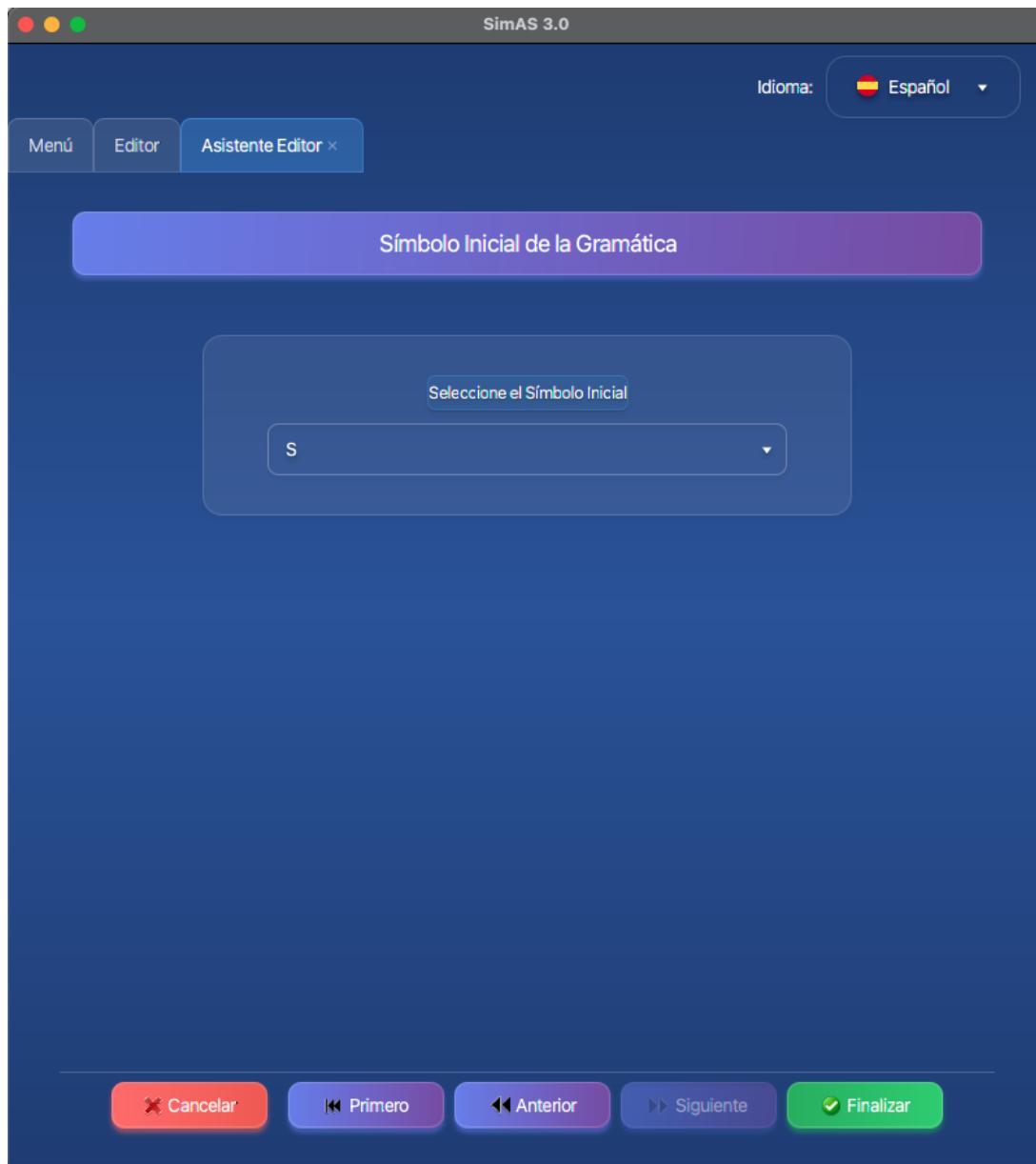


Figura 4.12: Paso 4: Selección del símbolo inicial

4.4.7.2. Estado vacío del paso 4

Si el paso 4 está vacío, se muestra la interfaz de la figura 4.13 y no se permite finalizar el proceso.

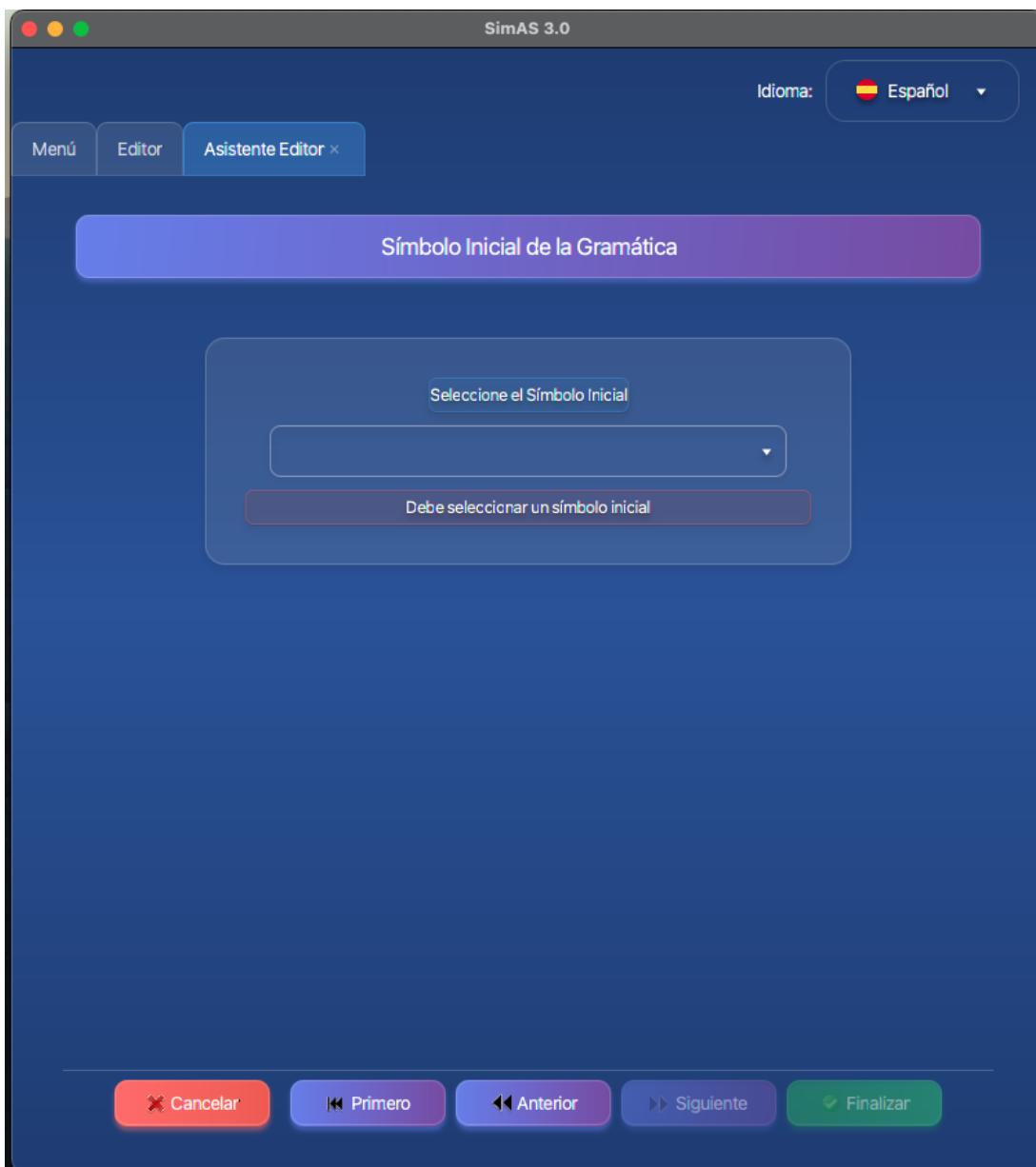


Figura 4.13: Paso 4: Estado vacío (no se puede finalizar)

4.4.7.3. Selección del símbolo inicial

En este paso, el usuario debe:

- **Seleccionar símbolo inicial:** elegir uno de los símbolos no terminales definidos como símbolo inicial de la gramática.
- **Finalizar proceso:** al completar la selección, se puede finalizar el proceso.

4.4.7.4. Finalización del proceso

Al finalizar el paso 4:

- **Reflejo en el editor:** todos los cambios se reflejan automáticamente en la pestaña del editor principal.
- **Validación automática:** la gramática es validada automáticamente por el sistema.
- **Cierre del asistente:** el asistente se cierra y regresa al editor principal.

4.5. Gestión de archivos

El editor incluye un sistema completo de gestión de archivos que permite guardar, cargar y exportar gramáticas.

4.5.1. Formato de archivo

Las gramáticas se guardan en formato XML, que proporciona:

- **Estructura clara:** organización jerárquica de los elementos.
- **Extensibilidad:** facilidad para agregar nuevas características.
- **Compatibilidad:** estándar ampliamente soportado.
- **Legibilidad:** formato que puede ser leído por humanos.

4.5.2. Operaciones de archivo

Las operaciones disponibles incluyen:

- **Nuevo:** crear una nueva gramática en blanco.
- **Abrir:** cargar una gramática existente desde el disco.
- **Guardar:** guardar la gramática actual.
- **Eliminar:** eliminar la gramática actual.
- **Validar:** validar la gramática actual.
- **Informe:** generar un informe detallado de la gramática actual.

4.5.3. Ubicación de archivos

Los archivos se pueden guardar en:

- **Ubicación personalizada:** el usuario elige dónde guardar.

- **Carpeta de proyectos:** directorio dedicado para gramáticas.
- **Ubicación temporal:** se guarda en el directorio temporal para trabajos en progreso.
- **Ubicación de respaldo:** copias de seguridad generadas por el usuario.

4.6. Validación de gramáticas

La validación de gramáticas es un proceso fundamental que asegura la corrección sintáctica y semántica de las gramáticas creadas.

4.6.1. Tipos de validación

El editor realiza dos tipos de validación:

- **Validación automática:** se ejecuta automáticamente cuando se carga una gramática desde archivo.
- **Validación manual:** debe ser ejecutada por el usuario al finalizar el proceso de creación o edición.

4.6.2. Proceso de validación manual

Para validar una gramática manualmente:

1. Complete el proceso de creación/edición en el asistente.
2. Haga clic en el botón "Validar" en la barra de herramientas del editor.
3. El sistema ejecutará la validación y mostrará los resultados.

4.6.3. Validación exitosa

Cuando la gramática es válida, se muestra la interfaz de la figura 4.14.

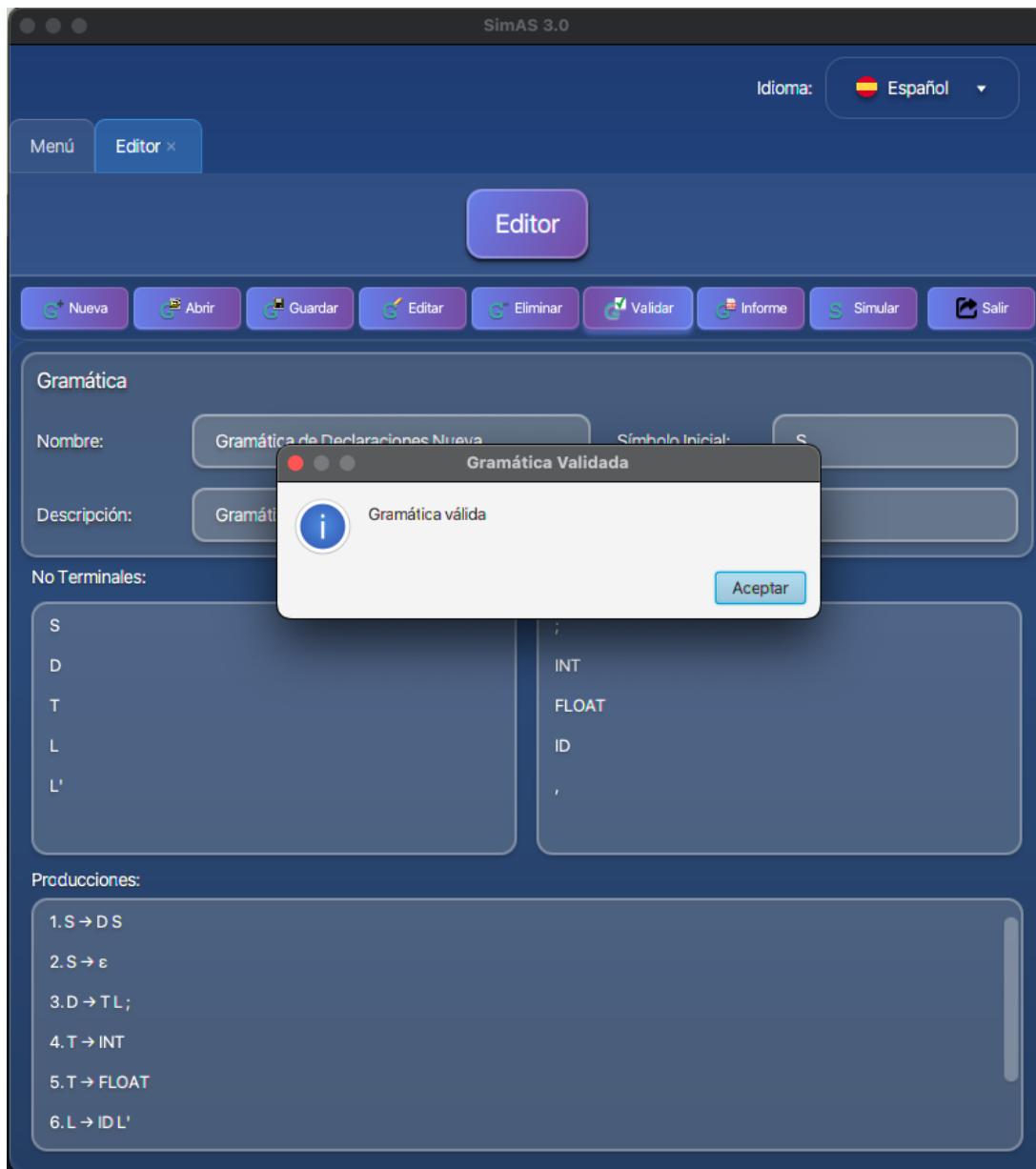


Figura 4.14: Validación exitosa de la gramática

4.6.4. Validación con errores

Cuando la gramática contiene errores, se muestra la interfaz de la figura 4.15.

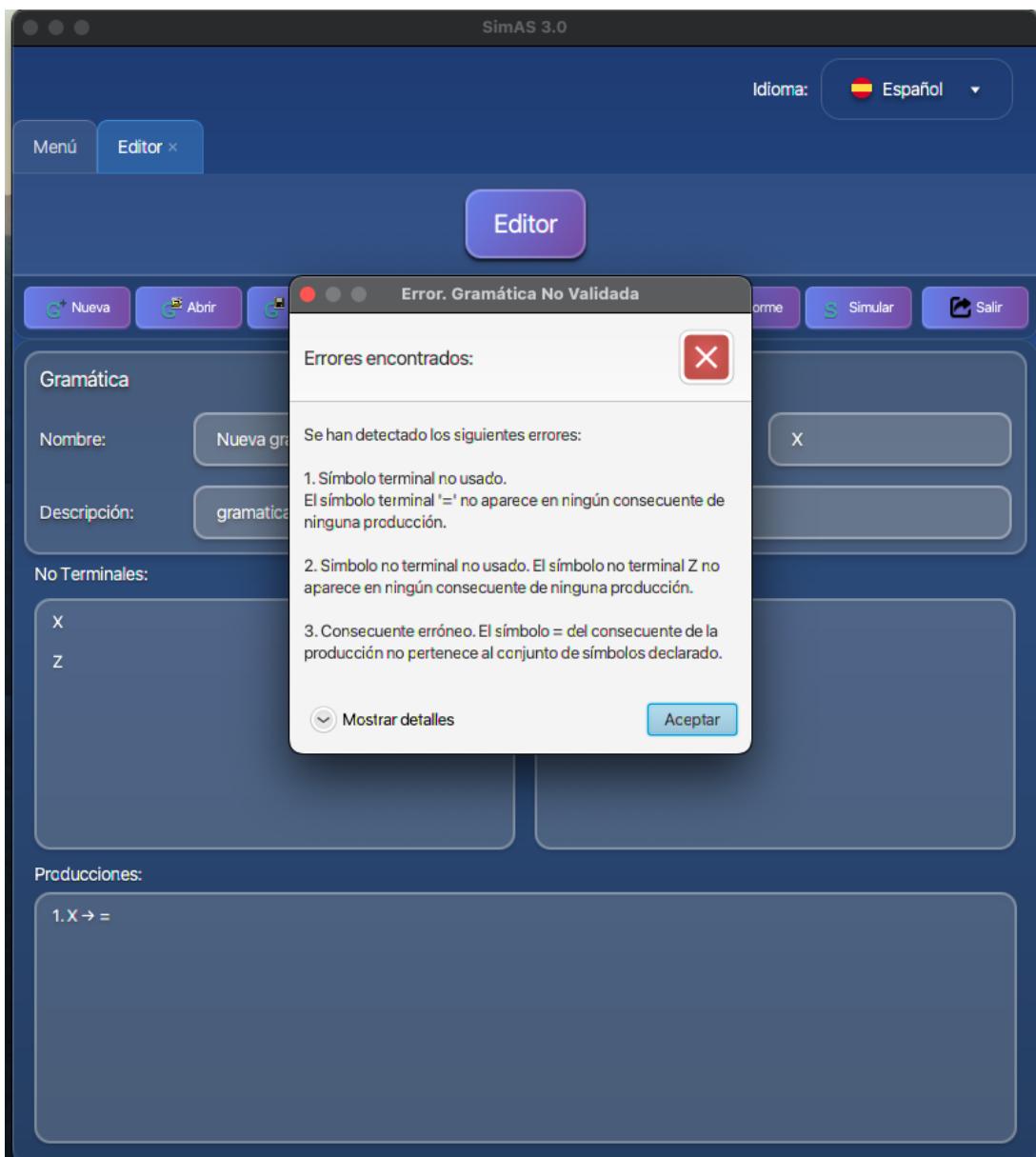


Figura 4.15: Validación con errores en la gramática

4.6.5. Tipos de errores detectados

El sistema de validación detecta los siguientes tipos de errores:

- **Símbolos duplicados:** cuando se define el mismo símbolo múltiples veces.
- **Referencias no definidas:** cuando se usa un símbolo que no ha sido definido.
- **Producciones mal formadas:** cuando las producciones no siguen el formato correcto.
- **Símbolo inicial no definido:** cuando el símbolo inicial no está en la lista de no terminales.

4.6.6. Ejemplo de error: símbolo duplicado

La figura 4.16 muestra un ejemplo de error por símbolo duplicado.

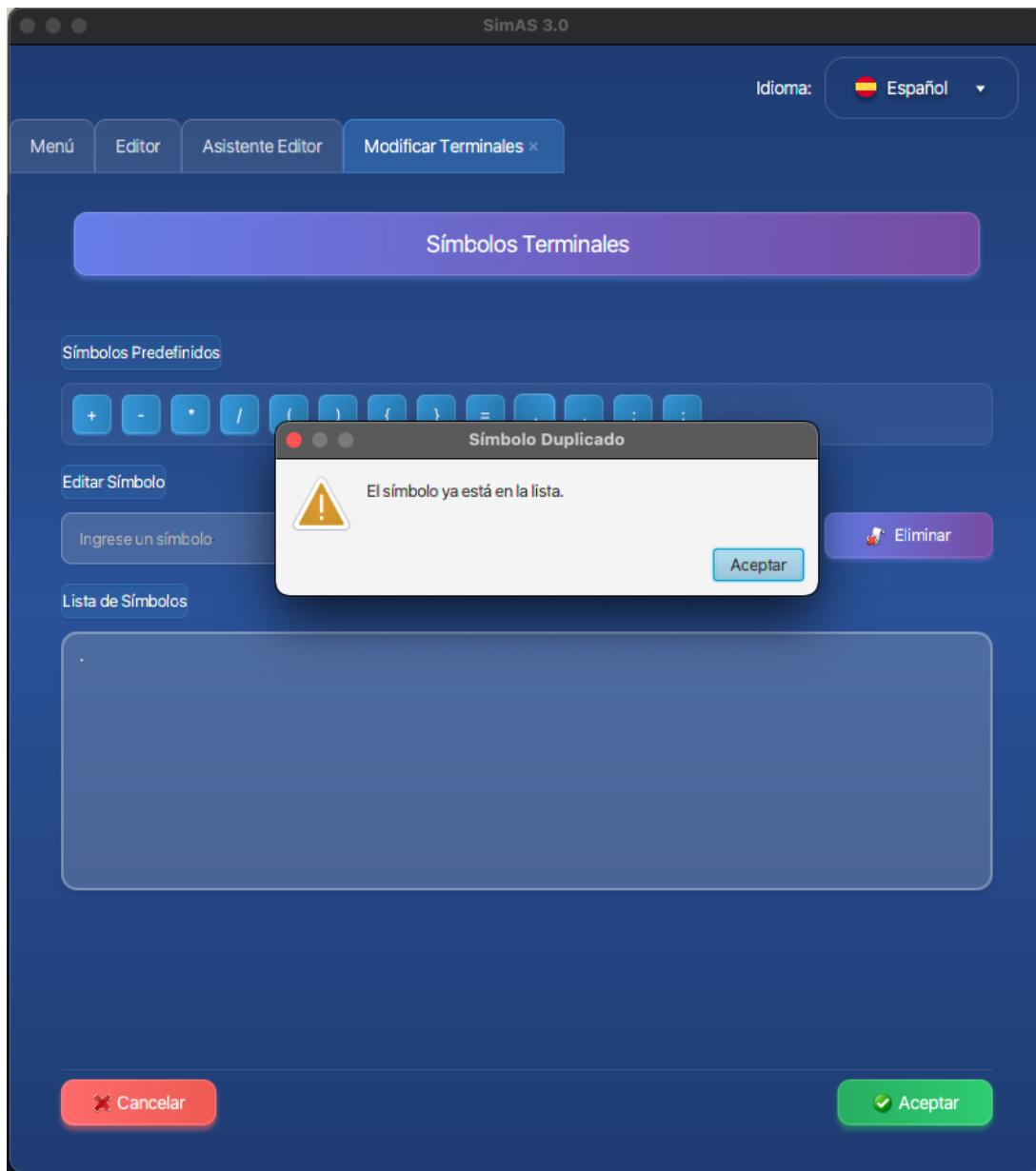


Figura 4.16: Ejemplo de error: símbolo duplicado

4.6.7. Corrección de errores

Para corregir errores de validación:

1. Revise los mensajes de error mostrados en la interfaz.
2. Identifique el tipo de error y su ubicación.
3. Use el botón "Editar" para abrir el asistente y corregir los errores.

4. Vuelva a validar la gramática hasta que sea correcta.

4.7. Integración con el simulador

El editor está completamente integrado con el simulador de análisis sintáctico, permitiendo una transición fluida entre la creación y la prueba de gramáticas.

4.7.1. Transición al simulador

Para probar una gramática:

1. Complete la creación y validación de la gramática en el editor.
2. Haga clic en el botón "Simular" en la barra de herramientas.
3. La gramática se transfiere automáticamente al simulador.
4. Se abre una nueva pestaña con el simulador.

4.7.2. Sincronización de datos

La integración incluye:

- **Transferencia automática:** la gramática se pasa al simulador sin intervención manual.
- **Estado compartido:** ambos componentes mantienen el mismo estado de la gramática.
- **Navegación fluida:** transición suave entre editor y simulador.

4.8. Resolución de problemas

Esta sección aborda los problemas más comunes que pueden surgir al usar el editor.

4.8.1. Problemas de validación

Si la gramática no pasa la validación:

1. Revise los mensajes de error mostrados en la interfaz.
2. Verifique que todos los símbolos estén correctamente definidos.
3. Asegúrese de que las producciones sigan el formato correcto.
4. Use el botón "Editar" para corregir los errores identificados.

4.8.2. Problemas de archivos

Para resolver problemas con archivos:

- **Formato de archivo:** asegúrese de que los archivos estén en formato XML válido.
- **Versión:** verifique que esté usando la versión correcta de SimAS 3.0.
- **Caracteres especiales:** evite caracteres no estándar en nombres de símbolos.
- **Encoding:** use codificación UTF-8 para archivos de texto.

4.9. Conclusión

El editor de gramáticas de SimAS 3.0 representa una herramienta completa y profesional para la creación y gestión de gramáticas libres de contexto. Su diseño intuitivo, basado en un asistente guiado de 4 pasos, lo convierte en una solución ideal tanto para estudiantes que están aprendiendo los conceptos básicos como para profesionales que requieren herramientas sofisticadas.

El asistente guiado simplifica significativamente el proceso de creación de gramáticas, dividiendo la tarea compleja en pasos manejables: definición de datos, gestión de símbolos, creación de producciones y selección del símbolo inicial. La validación automática para gramáticas cargadas y las creadas o modificadas aseguran la corrección de las gramáticas antes de su uso.

La integración perfecta con el simulador, la gestión robusta de archivos en formato XML, y la capacidad de detectar y corregir errores comunes hacen del editor una herramienta indispensable para cualquier persona que trabaje con gramáticas formales. El enfoque pedagógico del asistente guiado, junto con las funciones de validación y corrección de errores, asegura que la herramienta sea útil en todos los niveles de experiencia.

La capacidad de trabajar con el formato BNF estándar, la gestión eficiente de archivos, y la integración profunda con el resto de la aplicación, posicionan al editor de gramáticas como el componente central de SimAS 3.0, facilitando el aprendizaje y la aplicación práctica de los conceptos de análisis sintáctico descendente predictivo.

Capítulo 5

Simulación del analizador descendente predictivo

El simulador de análisis sintáctico descendente predictivo representa el núcleo tecnológico más sofisticado de SimAS 3.0, diseñado específicamente para simular y visualizar el proceso completo de análisis sintáctico de gramáticas libres de contexto utilizando el algoritmo LL(1). Este componente avanzado no solo ejecuta el análisis, sino que proporciona una experiencia educativa inmersiva que permite a los usuarios comprender profundamente los mecanismos internos del análisis sintáctico predictivo, desde la construcción de la tabla de análisis hasta la ejecución paso a paso del algoritmo.

5.1. Introducción al simulador

El simulador de SimAS 3.0 implementa un analizador descendente predictivo de última generación que utiliza una tabla de análisis precomputada para determinar de manera determinista qué producción aplicar en cada paso del análisis. Este enfoque algorítmico avanzado garantiza un análisis eficiente y completamente determinista de las cadenas de entrada, siempre que la gramática cumpla con las condiciones LL(1). La implementación incluye optimizaciones específicas para el manejo de gramáticas complejas y sistemas de recuperación de errores robustos que mejoran significativamente la experiencia del usuario.

5.1.1. Características principales

El simulador incorpora un conjunto integral de funcionalidades avanzadas que lo convierten en una herramienta educativa y profesional de primer nivel:

- **Asistente guiado inteligente:** proceso de configuración estructurado en 5 pasos secuenciales con validación automática en tiempo real y retroalimentación inmediata sobre la corrección de cada transformación.
- **Refactorización automática avanzada:** eliminación sistemática de recursividad izquierda directa e indirecta, factorización de producciones con prefijos comunes, y optimización de la estructura gramatical para análisis LL(1).

- **Cálculo de conjuntos PRIMERO y SIGUIENTE:** generación automática y eficiente de los conjuntos fundamentales utilizando algoritmos optimizados que manejan gramáticas de cualquier complejidad.
- **Construcción de tabla predictiva:** generación automática de la tabla de análisis LL(1) con detección de conflictos y sugerencias de resolución.
- **Sistema de funciones de error:** implementación de un sistema avanzado de recuperación de errores con funciones predefinidas y capacidad de personalización completa.
- **Simulación interactiva completa:** análisis paso a paso con control granular del usuario, incluyendo avance, retroceso, pausa y reinicio en cualquier momento del proceso.
- **Visualización dinámica:** generación automática de derivaciones en formato BNF y árboles sintácticos que se actualizan en tiempo real durante la simulación.
- **Múltiples simulaciones concurrentes:** capacidad de ejecutar y gestionar múltiples simulaciones simultáneamente para análisis comparativo y validación exhaustiva.

5.2. Acceso al simulador

El simulador está diseñado con múltiples puntos de acceso estratégicos que facilitan su uso en diferentes contextos de trabajo, optimizando el flujo de trabajo del usuario:

5.2.1. Desde el menú principal

El acceso principal al simulador se realiza a través de la interfaz principal de la aplicación:

- **Menú "Simulador":** acceso directo e inmediato al simulador principal, ideal para usuarios que desean comenzar una nueva simulación desde cero.
- **Atajo de teclado "Ctrl + S":** acceso rápido mediante combinación de teclas para usuarios experimentados que prefieren la eficiencia del teclado.

5.2.2. Desde el editor de gramáticas

La integración con el editor proporciona una transición fluida y contextual:

- **Botón "Simular":** disponible únicamente cuando hay una gramática válida cargada en el editor, garantizando que el usuario siempre trabaje con gramáticas correctas y completas.

Esta integración asegura que la gramática editada se transfiera automáticamente al simulador, manteniendo la consistencia y evitando errores de configuración.

5.3. Asistente guiado del simulador

El asistente del simulador representa una innovación en la experiencia de usuario, diseñado para guiar de manera intuitiva y educativa a través de un proceso estructurado de 5 pasos que transforma una gramática libre de contexto en un sistema de análisis sintáctico completamente funcional. Cada paso está cuidadosamente diseñado para realizar transformaciones específicas y cálculos fundamentales, proporcionando al usuario una comprensión profunda de cada etapa del proceso.

5.3.1. Navegación del asistente

El asistente incorpora un sistema de navegación sofisticado que maximiza la usabilidad y el control del usuario:

- **Navegación bidireccional:** botones "Anterior" y "Siguiente" permiten moverse libremente entre los pasos, facilitando la revisión y corrección de configuraciones previas.
- **Acceso a la gramática original:** el botón "Gramática" está disponible en el menú inferior de todos los pasos, proporcionando acceso inmediato a la gramática original sin refactorizar ni eliminar recursividad, como se puede observar en la figura 5.1. Esta funcionalidad es crucial para comparar el estado original con las transformaciones aplicadas.
- **Validación automática en tiempo real:** cada paso implementa un sistema de validación automática que verifica la corrección de los resultados antes de permitir continuar, garantizando la integridad del proceso y proporcionando retroalimentación inmediata.

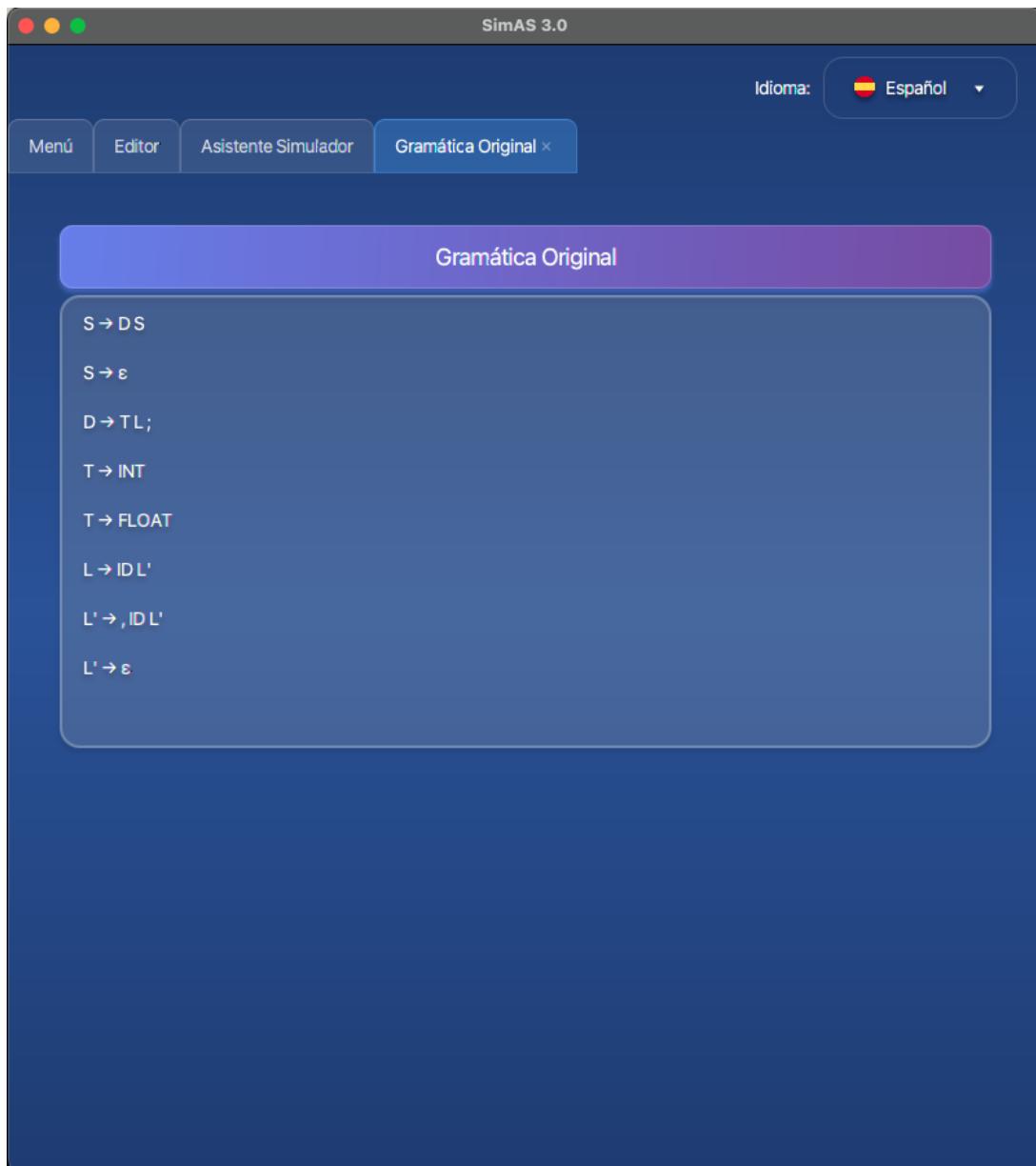


Figura 5.1: Vista de la gramática original desde el botón "Gramática".

5.3.2. Paso 1: Refactorización y eliminación de recursividad

El primer paso del asistente constituye la fase más crítica del proceso, donde el simulador procesa la gramática original aplicando transformaciones algorítmicas avanzadas para prepararla adecuadamente para el análisis LL(1). Este paso es fundamental porque determina la viabilidad del análisis predictivo y establece las bases para todos los cálculos posteriores.

5.3.2.1. Procesos realizados

El simulador ejecuta una secuencia de algoritmos especializados:

- **Eliminación de recursividad izquierda:** implementa algoritmos avanzados que detectan y eliminan tanto recursividad directa como indirecta, transformando la gramática en una forma equivalente que no presenta este problema estructural.
- **Factorización de producciones:** aplica técnicas de factorización para identificar y resolver producciones con prefijos comunes, eliminando ambigüedades que podrían causar conflictos en la tabla predictiva.
- **Validación LL(1):** ejecuta un análisis exhaustivo para verificar que la gramática resultante cumple con todas las condiciones necesarias para ser LL(1), incluyendo la ausencia de conflictos en la tabla predictiva.

5.3.2.2. Interfaz del paso 1

La interfaz del primer paso, ilustrada en la figura 5.2, presenta una vista comprensiva del proceso de transformación:

- **Gramática refactorizada:** muestra la gramática completamente transformada con todas las modificaciones aplicadas, permitiendo al usuario verificar la corrección de las transformaciones.
- **Estado de la gramática:** proporciona indicadores visuales claros sobre la validez y completitud de la gramática, incluyendo métricas de calidad y advertencias sobre posibles problemas.
- **Detalles de transformaciones:** presenta información detallada sobre cada cambio realizado, incluyendo el tipo de transformación aplicada y su justificación algorítmica.

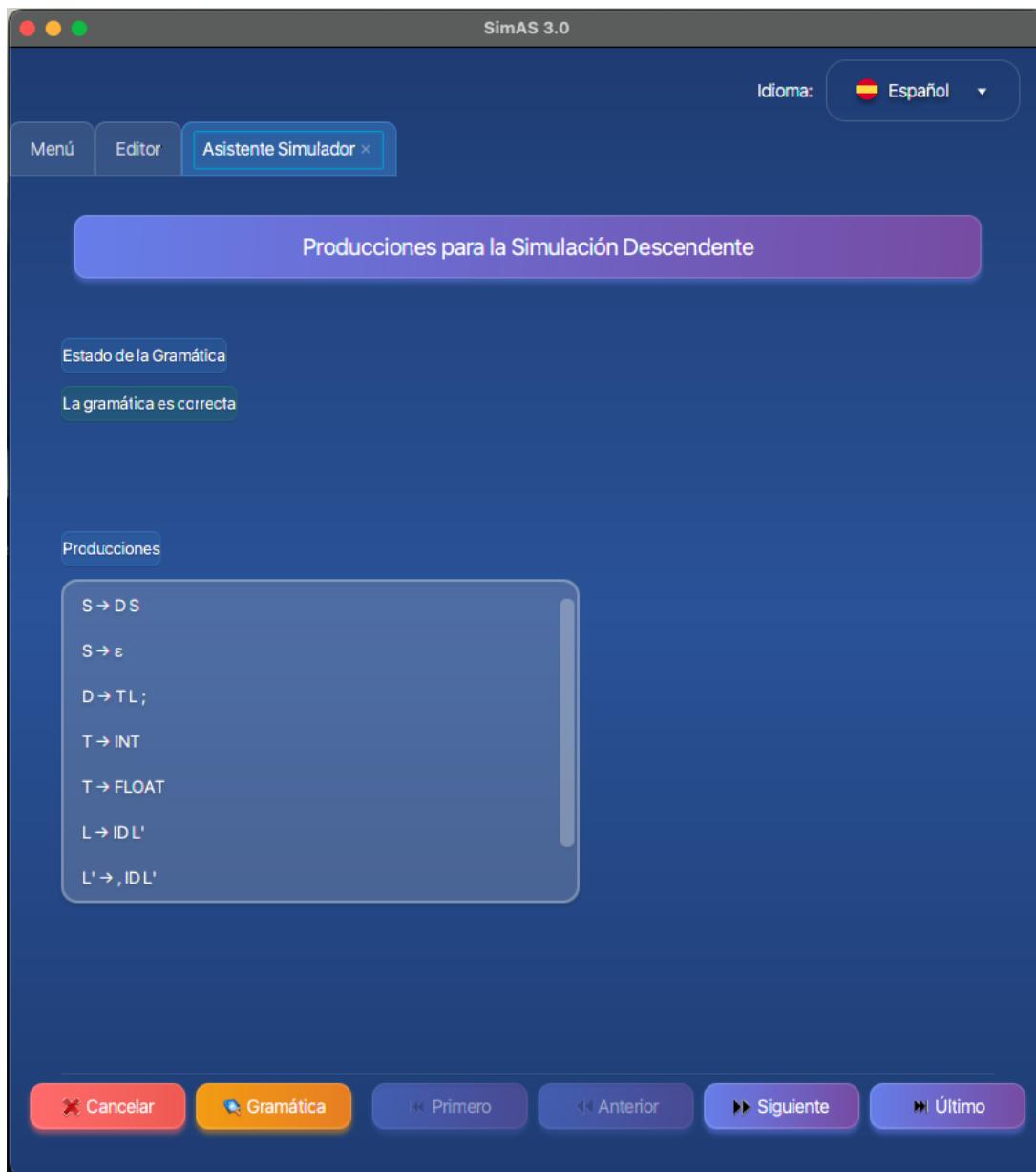


Figura 5.2: Paso 1: Refactorización y eliminación de recursividad

5.3.3. Paso 2: Construcción de conjuntos PRIMERO y SIGUIENTE

El segundo paso representa el núcleo computacional del análisis LL(1), donde el simulador calcula los conjuntos PRIMERO y SIGUIENTE utilizando algoritmos optimizados. Estos conjuntos son fundamentales para la construcción de la tabla predictiva, ya que determinan qué producción aplicar en cada situación durante el análisis.

5.3.3.1. Conjunto PRIMERO

El conjunto PRIMERO de un símbolo α , denotado como $PRIMERO(\alpha)$, contiene todos los símbolos terminales que pueden aparecer como primer símbolo en las derivaciones

de α . El cálculo de PRIMERO es crucial porque determina las condiciones bajo las cuales una producción puede ser aplicada durante el análisis descendente.

El algoritmo implementado maneja casos especiales como:

- Símbolos terminales: $PRIMERO(a) = \{a\}$ para cualquier terminal a .
- Símbolos no terminales: cálculo recursivo considerando todas las producciones.
- Cadenas de símbolos: propagación de PRIMERO a través de concatenaciones.
- Manejo de ε : inclusión de ε cuando es derivable.

5.3.3.2. Conjunto SIGUIENTE

El conjunto SIGUIENTE de un símbolo no terminal A , denotado como $SIGUIENTE(A)$, contiene todos los símbolos terminales que pueden aparecer inmediatamente después de A en alguna derivación desde el símbolo inicial. SIGUIENTE es esencial para manejar producciones que derivan ε .

El algoritmo considera:

- El símbolo inicial: $\$ \in SIGUIENTE(S)$ donde S es el símbolo inicial.
- Propagación a través de producciones: si $A \rightarrow \alpha B \beta$, entonces $PRIMERO(\beta) \subseteq SIGUIENTE(B)$.
- Propagación de ε : si $A \rightarrow \alpha B$ y $\varepsilon \in PRIMERO(\beta)$, entonces $SIGUIENTE(A) \subseteq SIGUIENTE(B)$.

5.3.3.3. Interfaz del paso 2

La interfaz del segundo paso, mostrada en la figura 5.3, presenta los resultados del cálculo de manera clara y organizada:

- **Conjuntos PRIMERO:** presentación tabular de todos los conjuntos PRIMERO calculados para cada símbolo no terminal, con indicadores visuales para símbolos especiales como ε .
- **Conjuntos SIGUIENTE:** visualización completa de todos los conjuntos SIGUIENTE, organizados de manera que facilite la verificación manual y la comprensión de las relaciones entre símbolos.

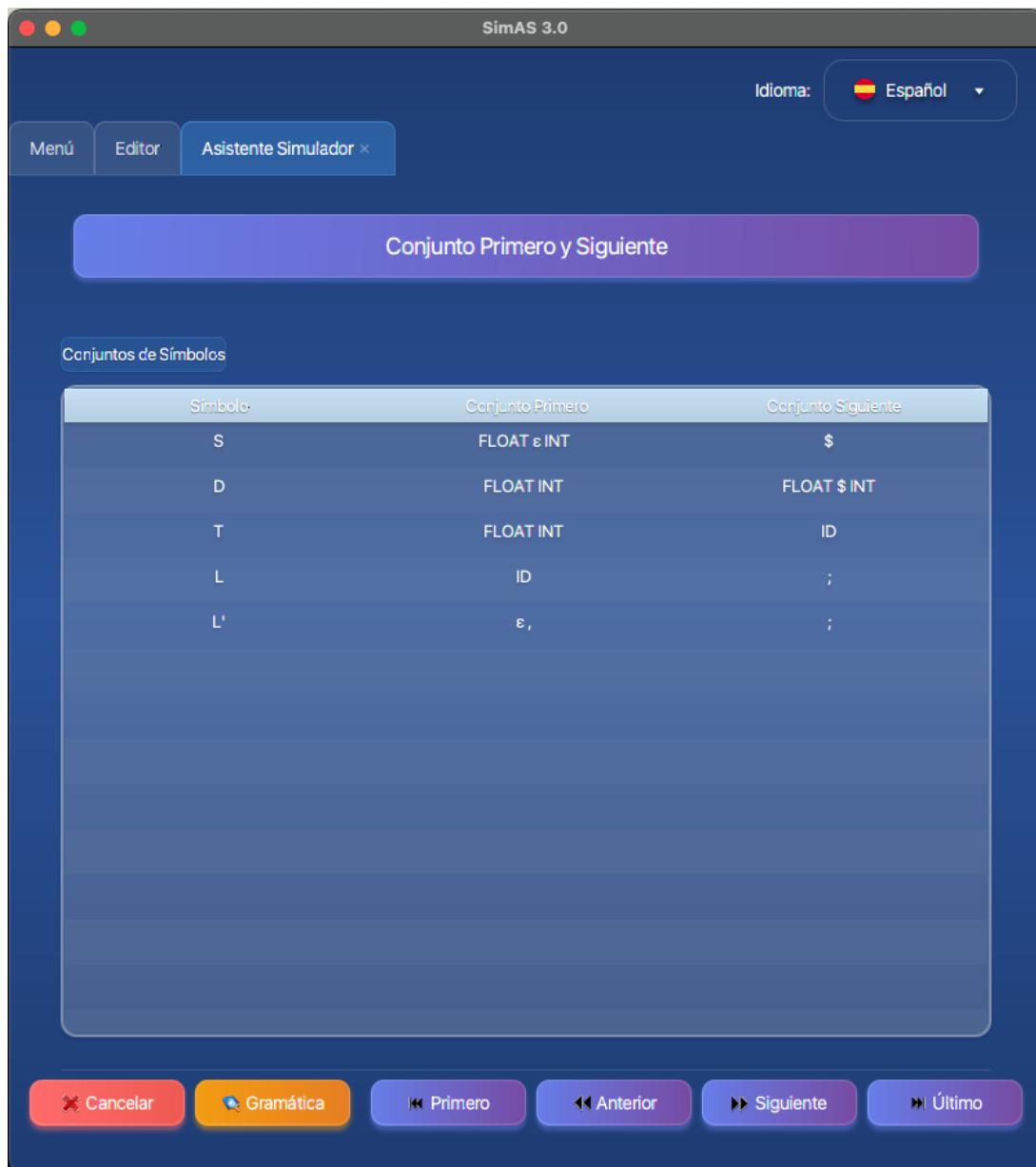


Figura 5.3: Paso 2: Construcción de conjuntos PRIMERO y SIGUIENTE

5.3.4. Paso 3: Construcción de la tabla predictiva

El tercer paso representa la culminación del proceso de preparación, donde el simulador construye la tabla de análisis predictivo utilizando los conjuntos PRIMERO y SIGUIENTE calculados en el paso anterior. Esta tabla es el corazón del analizador LL(1), ya que determina de manera determinista qué acción tomar en cada situación durante el análisis.

5.3.4.1. Algoritmo de construcción

La tabla predictiva M se construye aplicando un algoritmo sistemático que garantiza la corrección y completitud:

1. **Inicialización:** todas las entradas de la tabla se inicializan como vacías.
2. **Construcción para cada producción:** para cada producción $A \rightarrow \alpha$:
 - Si $a \in PRIMERO(\alpha)$ y $a \neq \varepsilon$, entonces $M[A, a] = A \rightarrow \alpha$.
 - Si $\varepsilon \in PRIMERO(\alpha)$ y $a \in SIGUIENTE(A)$, entonces $M[A, a] = A \rightarrow \alpha$.
3. **Detección de conflictos:** el algoritmo detecta automáticamente conflictos (múltiples producciones para la misma entrada) y los reporta al usuario.
4. **Validación LL(1):** verifica que la tabla resultante define una función total, confirmado que la gramática es LL(1).

5.3.4.2. Interfaz del paso 3

La interfaz del tercer paso, ilustrada en la figura 5.4, presenta la tabla predictiva de manera intuitiva y educativa:

- **Tabla predictiva completa:** matriz bidimensional que muestra la intersección de símbolos no terminales (filas) y terminales (columnas), con cada celda conteniendo la producción correspondiente o indicando que está vacía.
- **Producciones asignadas:** cada celda no vacía muestra claramente la producción que debe aplicarse, facilitando la comprensión del proceso de análisis y la verificación manual de la corrección.
- **Indicadores visuales:** uso de colores y símbolos para distinguir entre diferentes tipos de entradas y facilitar la identificación de patrones en la tabla.

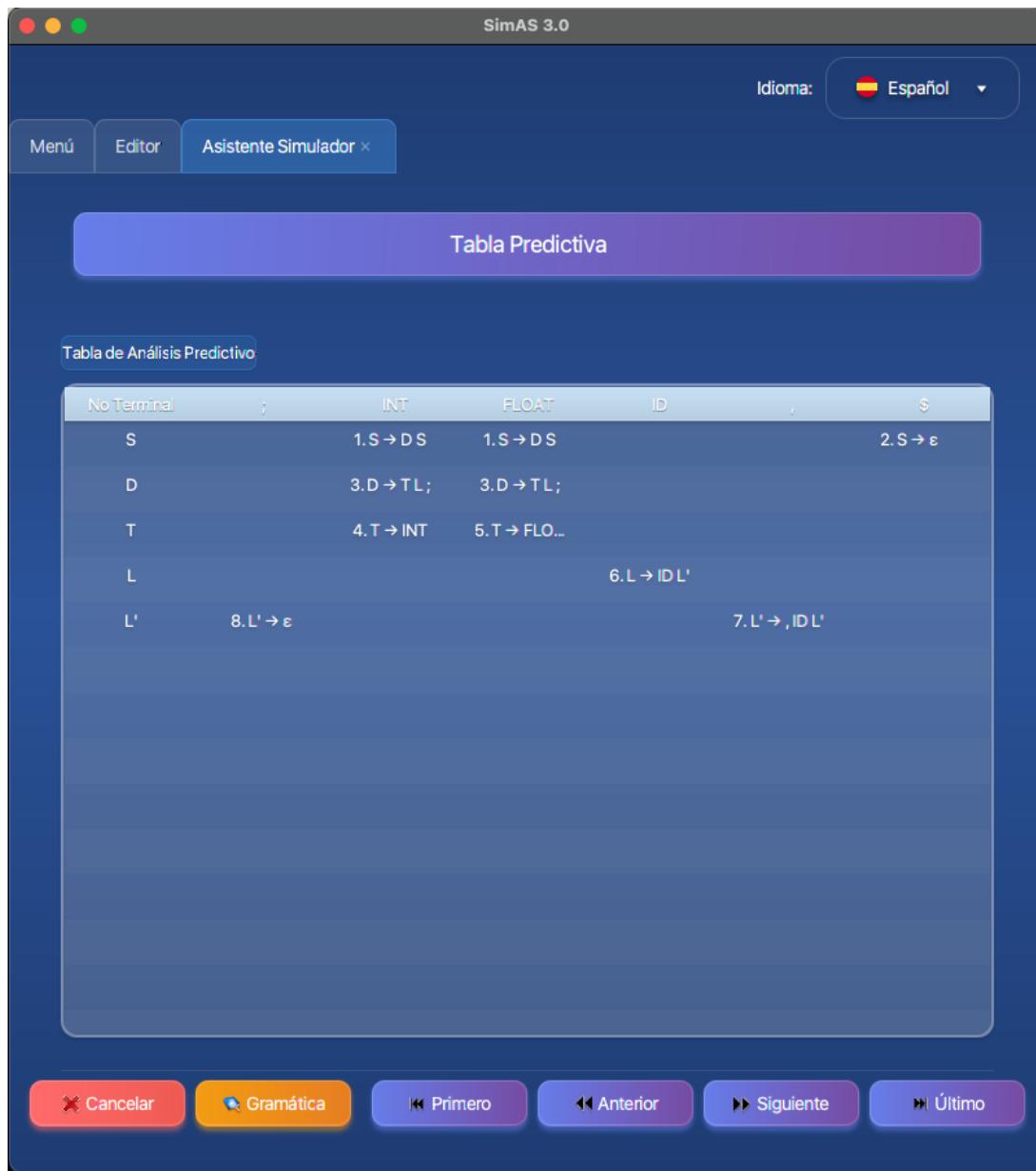


Figura 5.4: Paso 3: Construcción de la tabla predictiva

5.3.5. Paso 4: Funciones de error

El cuarto paso introduce una funcionalidad avanzada que permite configurar un sistema sofisticado de recuperación de errores. Aunque este paso es opcional, su implementación mejora significativamente la experiencia de simulación al proporcionar mecanismos robustos para manejar cadenas de entrada que no pertenecen al lenguaje definido por la gramática.

5.3.5.1. Funciones de error del sistema

El sistema incluye un conjunto predefinido de funciones de error que implementan estrategias estándar de recuperación:

- **Error de inserción:** inserta automáticamente un símbolo terminal esperado en la posición actual, permitiendo que el análisis continúe. Esta función es útil cuando falta un símbolo obligatorio.
- **Error de eliminación:** elimina un símbolo terminal inesperado de la entrada, saltándolo para continuar con el siguiente símbolo. Ideal para manejar símbolos extra o mal posicionados.
- **Error de reemplazo:** reemplaza un símbolo terminal inesperado por otro esperado, manteniendo la posición en la entrada. Útil para correcciones de símbolos similares o errores tipográficos.

5.3.5.2. Panel de funciones de error

Al hacer clic en "Nueva", se abre un panel auxiliar especializado, como se puede observar en las figuras 5.5 y 5.6, que permite:

- **Asignación automática de ID:** el sistema asigna automáticamente el siguiente identificador disponible, garantizando la unicidad y evitando conflictos de nomenclatura.
- **Definición de la función:** interfaz intuitiva para especificar el comportamiento detallado de la función de error, incluyendo condiciones de activación y acciones a realizar.
- **Validación en tiempo real:** verificación automática de que la función definida sea sintácticamente correcta y semánticamente válida antes de permitir su guardado.
- **Guardado seguro:** confirmación y almacenamiento de la función en el sistema, con verificación de integridad y compatibilidad con la gramática actual.

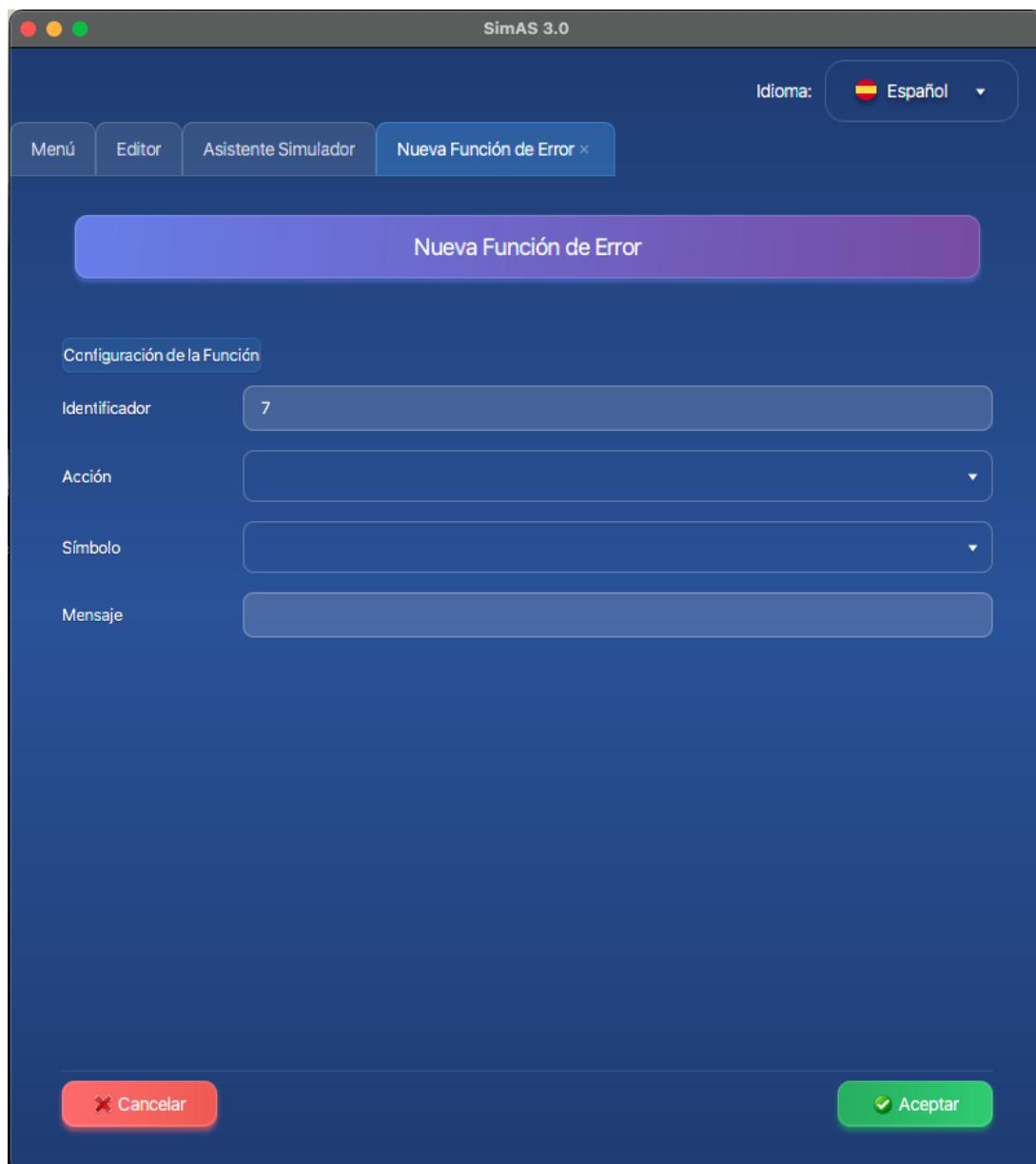


Figura 5.5: Panel auxiliar para crear nuevas funciones de error

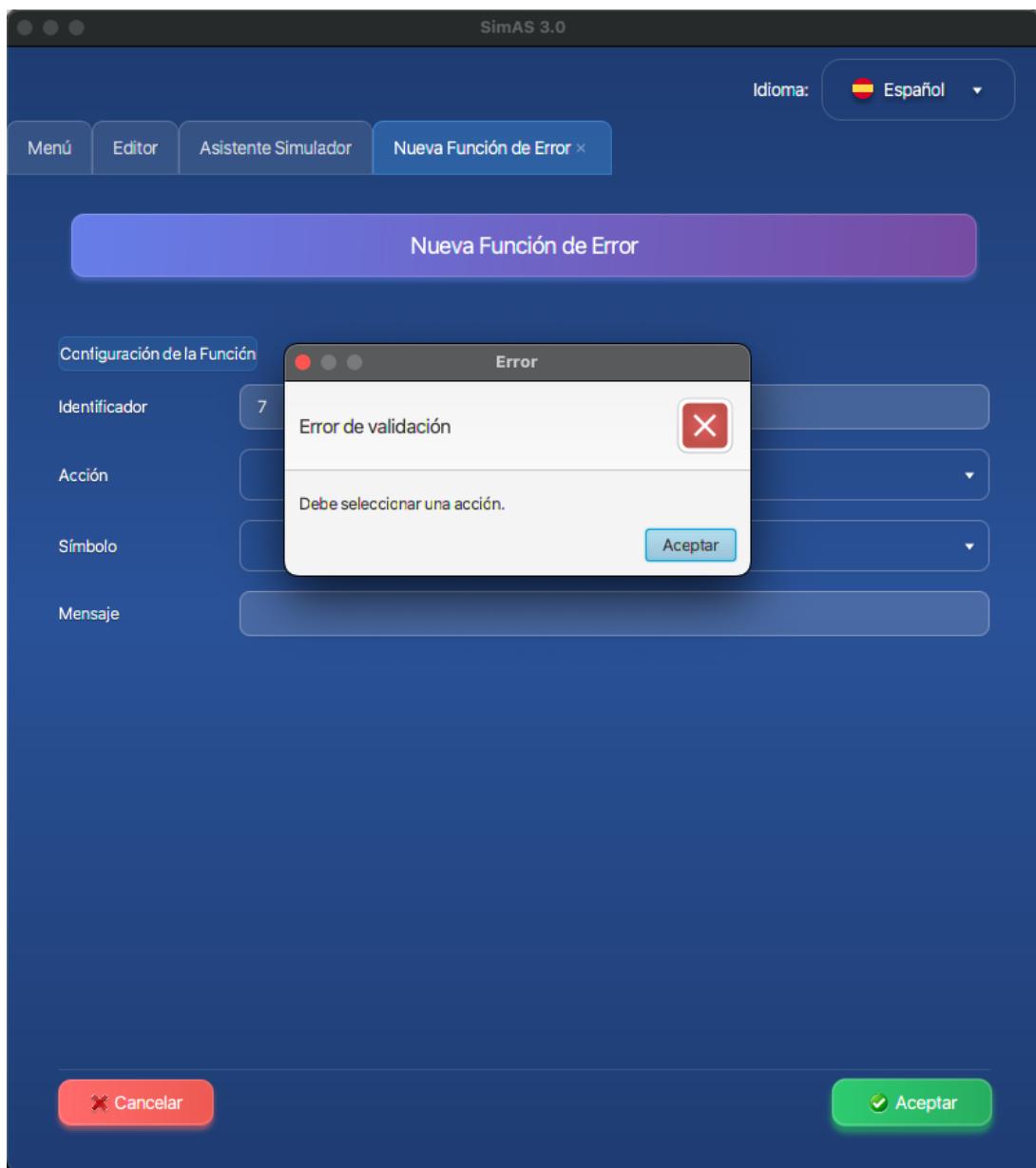


Figura 5.6: Panel de funciones de error en estado vacío

5.3.5.3. Gestión de funciones de error

El sistema de gestión, ilustrado en las figuras 5.7 y 5.8, permite al usuario:

- **Añadir nuevas funciones:** capacidad de crear funciones personalizadas de recuperación que se adapten a necesidades específicas o casos de uso particulares.
- **Eliminar funciones:** funcionalidad para quitar funciones no deseadas o que han demostrado ser problemáticas durante las pruebas.
- **Omitir funciones de error:** opción de continuar sin sistema de recuperación para análisis más estrictos o cuando se desea que el analizador falle inmediatamente ante errores.

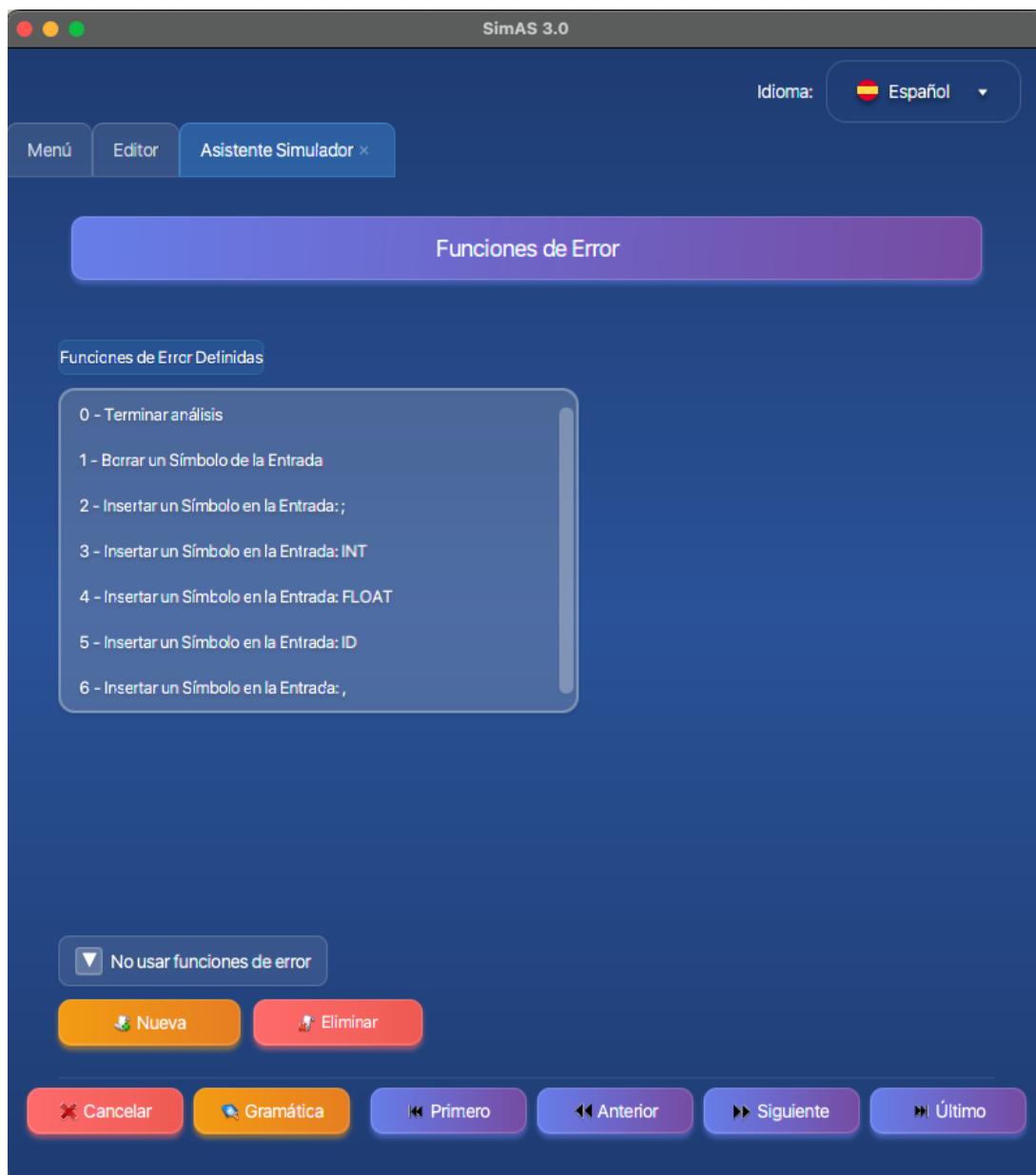


Figura 5.7: Paso 4: Configuración de funciones de error

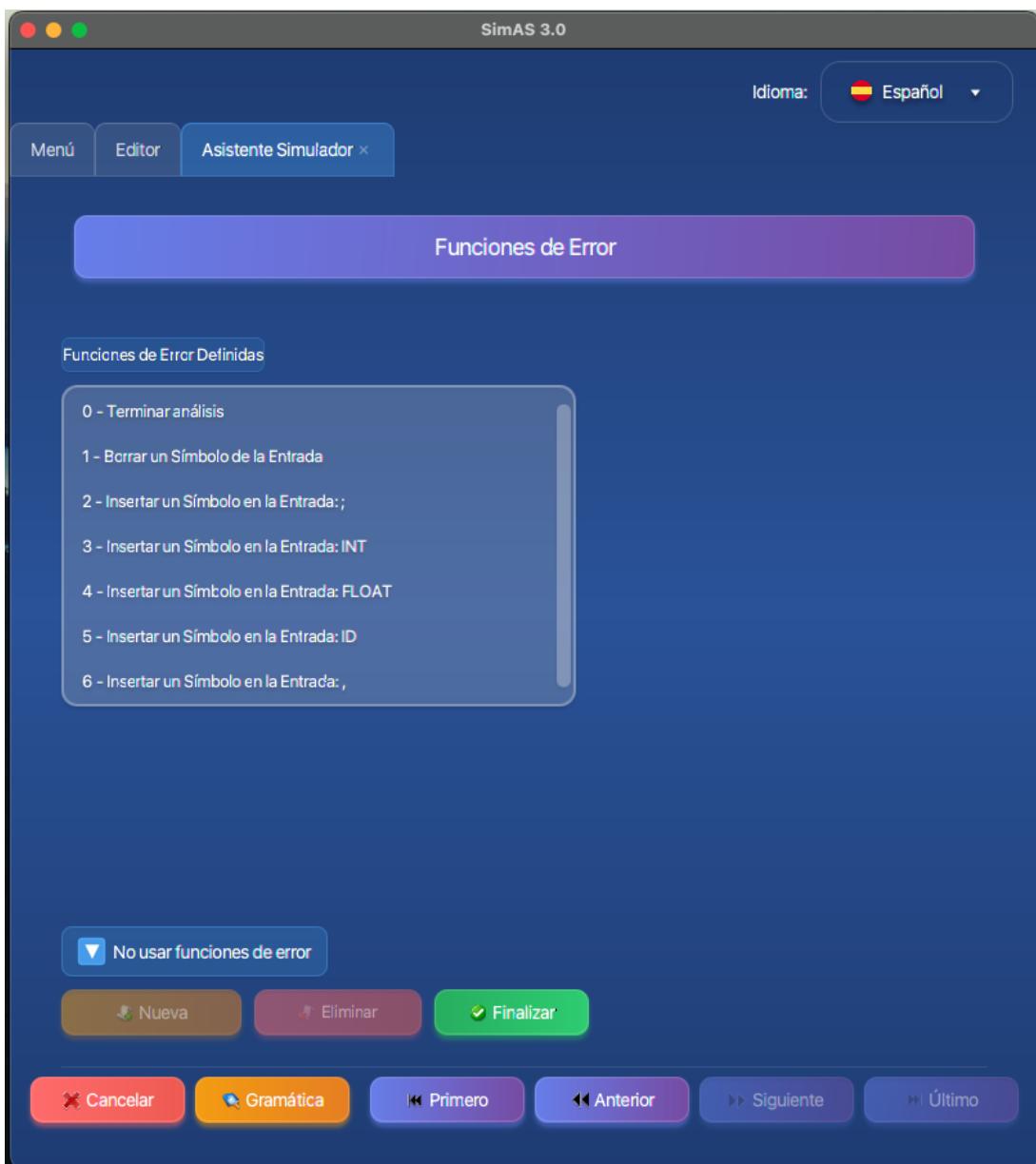


Figura 5.8: Paso 4: Opción de no usar funciones de error

5.3.6. Paso 5: Tabla predictiva completa

El quinto y último paso del asistente presenta la tabla predictiva final en su forma completa, integrando todas las producciones calculadas con las funciones de error configuradas. Este paso proporciona al usuario la oportunidad de realizar ajustes finales y personalizaciones antes de proceder con la simulación, asegurando que la configuración sea exactamente como se desea.

5.3.6.1. Interfaz del paso 5

La interfaz del paso final, mostrada en la figura 5.9, presenta una vista comprensiva y editable de la tabla predictiva:

- **Tabla predictiva completa:** visualización integral que combina todas las producciones calculadas con las funciones de error configuradas, proporcionando una vista unificada del comportamiento del analizador.
- **Celdas editables interactivas:** cada celda de la tabla es clickeable, permitiendo al usuario añadir, modificar o eliminar funciones de error en posiciones específicas según sus necesidades.
- **Herramientas de edición avanzadas:** conjunto completo de herramientas para modificar la tabla, incluyendo operaciones de edición masiva y validación en tiempo real.
- **Vista previa en tiempo real:** representación visual que se actualiza instantáneamente con cada modificación, permitiendo al usuario verificar inmediatamente el impacto de sus cambios.

The screenshot shows the 'SimAS 3.0' application window. At the top, there are three buttons: 'Menú', 'Editor', and 'Asistente Simulador'. The 'Asistente Simulador' button is currently selected. In the top right corner, there is a language selection dropdown set to 'Español'. Below the menu bar, a purple header bar says 'Tabla Simulación'. Underneath, a sub-header says 'Seleccione una función de error' followed by a dropdown menu showing '3 - Insertar un Símbolo en la Entrada: INT'. The main area contains a large table representing the parser's state transition matrix. The columns are labeled with non-terminals: S, D, T, L, L', ;, INT, FLOAT, ID, ', and \$. The rows are labeled with terminals: INT, FLOAT, ID, ', and \$. The table cells contain various entries: some are empty (ε), some are labeled with production rules like '1. S → DS', and others are highlighted in different colors (blue, green, red). At the bottom of the table, there are several buttons: 'Borrar' (Delete), 'Rellenar' (Fill), 'Resetear' (Reset), 'Cancelar' (Cancel), 'Gramática' (Grammar), 'Primero' (First), 'Anterior' (Previous), 'Siguiente' (Next), and 'Simulación' (Simulation).

Figura 5.9: Paso 5: Tabla predictiva completa con herramientas de edición

5.3.6.2. Herramientas de edición

El paso 5 incorpora un conjunto sofisticado de herramientas de edición diseñadas para facilitar la personalización de la tabla:

- **Borrar selectivo:** elimina el contenido de una celda seleccionada, restringido únicamente a funciones de error para preservar la integridad de las producciones fundamentales.
- **Resetear a estado original:** restaura la tabla a su configuración inicial, manteniendo solo las producciones básicas y eliminando todas las funciones de error añadidas.
- **Rellenar con epsilon:** llena las celdas vacías con símbolos epsilon para mejorar la legibilidad visual, aunque esta operación es puramente cosmética.

5.4. Simulador principal

Una vez completados exitosamente los 5 pasos del asistente, el usuario accede al simulador principal, que representa el núcleo operativo del sistema. Esta transición se realiza mediante el botón "Simulación", que activa el entorno de simulación con toda la configuración previamente establecida.

5.4.1. Interfaz del simulador

El simulador principal, ilustrado en la figura 5.10, presenta una interfaz diseñada para proporcionar acceso inmediato a todas las funcionalidades esenciales:

- **Resumen de configuración:** panel comprensivo que muestra las producciones modificadas, las funciones de error configuradas y la tabla predictiva final, proporcionando una vista de conjunto de toda la configuración.
- **Botón "Informe":** genera automáticamente un informe PDF detallado que documenta toda la configuración y los resultados del proceso de preparación.
- **Botón "Simular":** inicia el proceso de simulación interactiva, abriendo una nueva pestaña dedicada al análisis sintáctico.

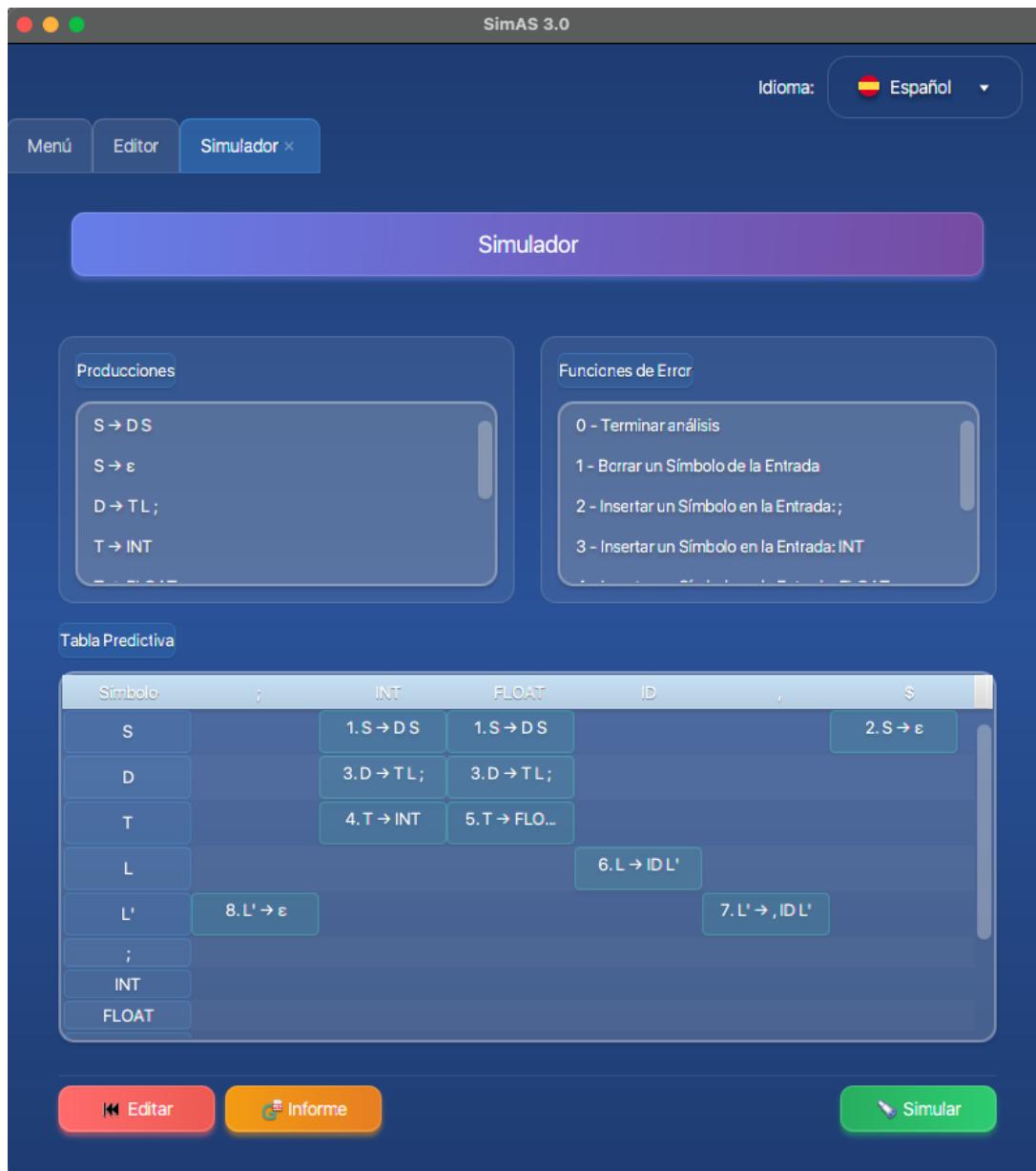


Figura 5.10: Interfaz del simulador principal

5.4.2. Generación de informes

El sistema de generación de informes representa una funcionalidad avanzada que produce documentación profesional y completa. El botón "Informe" genera un documento PDF estructurado que incluye:

- **Información del editor:** documentación completa de la gramática original y todas las modificaciones aplicadas durante el proceso de refactorización.
- **Conjuntos PRIMERO y SIGUIENTE:** presentación detallada de todos los conjuntos calculados en el paso 2, con explicaciones de su significado y uso.
- **Tabla predictiva:** representación completa de la tabla final, incluyendo todas las producciones y funciones de error integradas.

- **Funciones de error:** documentación exhaustiva de todas las funciones de error configuradas, incluyendo sus definiciones, comportamientos y casos de uso.
- **Estadísticas y métricas:** análisis cuantitativo de la gramática y la configuración, incluyendo métricas de complejidad, cobertura y eficiencia.

5.5. Simulación interactiva

La simulación interactiva representa el corazón de la experiencia educativa de SimAS 3.0. Al hacer clic en "Simular", se abre una nueva pestaña especializada que proporciona un entorno completo para realizar análisis sintáctico con control total del usuario sobre cada aspecto del proceso.

5.5.1. Interfaz de simulación

La pestaña de simulación, mostrada en las figuras 5.11 y 5.12, incorpora una interfaz intuitiva y funcional que incluye:

- **Campo de entrada inteligente:** área de texto especializada para introducir la cadena a analizar, con validación en tiempo real y sugerencias de símbolos válidos.
- **Símbolos disponibles:** panel que muestra todos los símbolos terminales válidos para la gramática actual, facilitando la construcción de cadenas de entrada correctas.
- **Controles de simulación avanzados:** conjunto completo de botones para iniciar, pausar, reanudar y controlar granularmente la simulación.
- **Historial de pasos detallado:** registro comprensivo de cada paso del análisis, incluyendo estados intermedios y decisiones tomadas.
- **Navegación bidireccional:** botones para avanzar y retroceder en los pasos, permitiendo revisar y analizar cualquier momento del proceso.

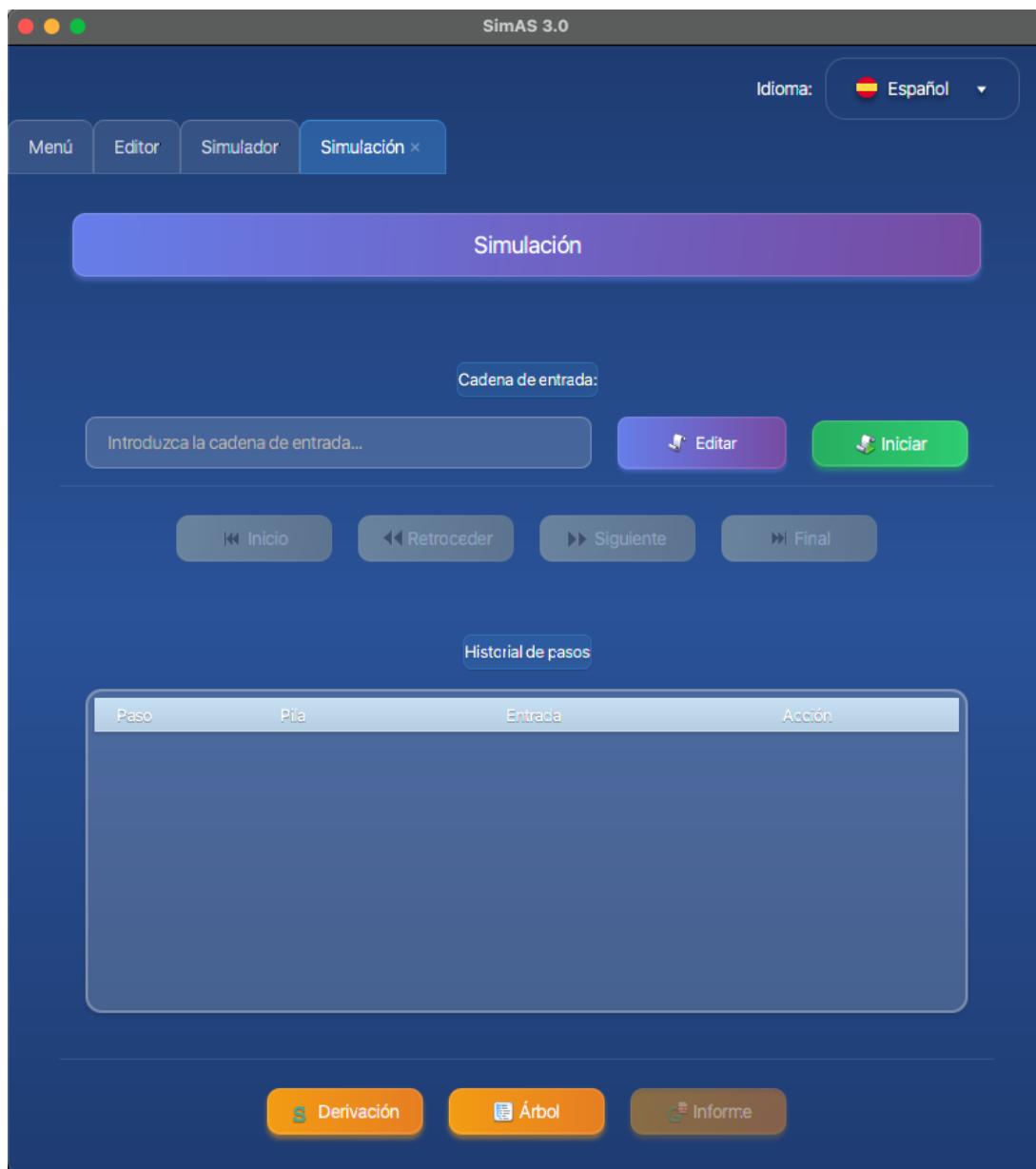


Figura 5.11: Interfaz de simulación en estado vacío

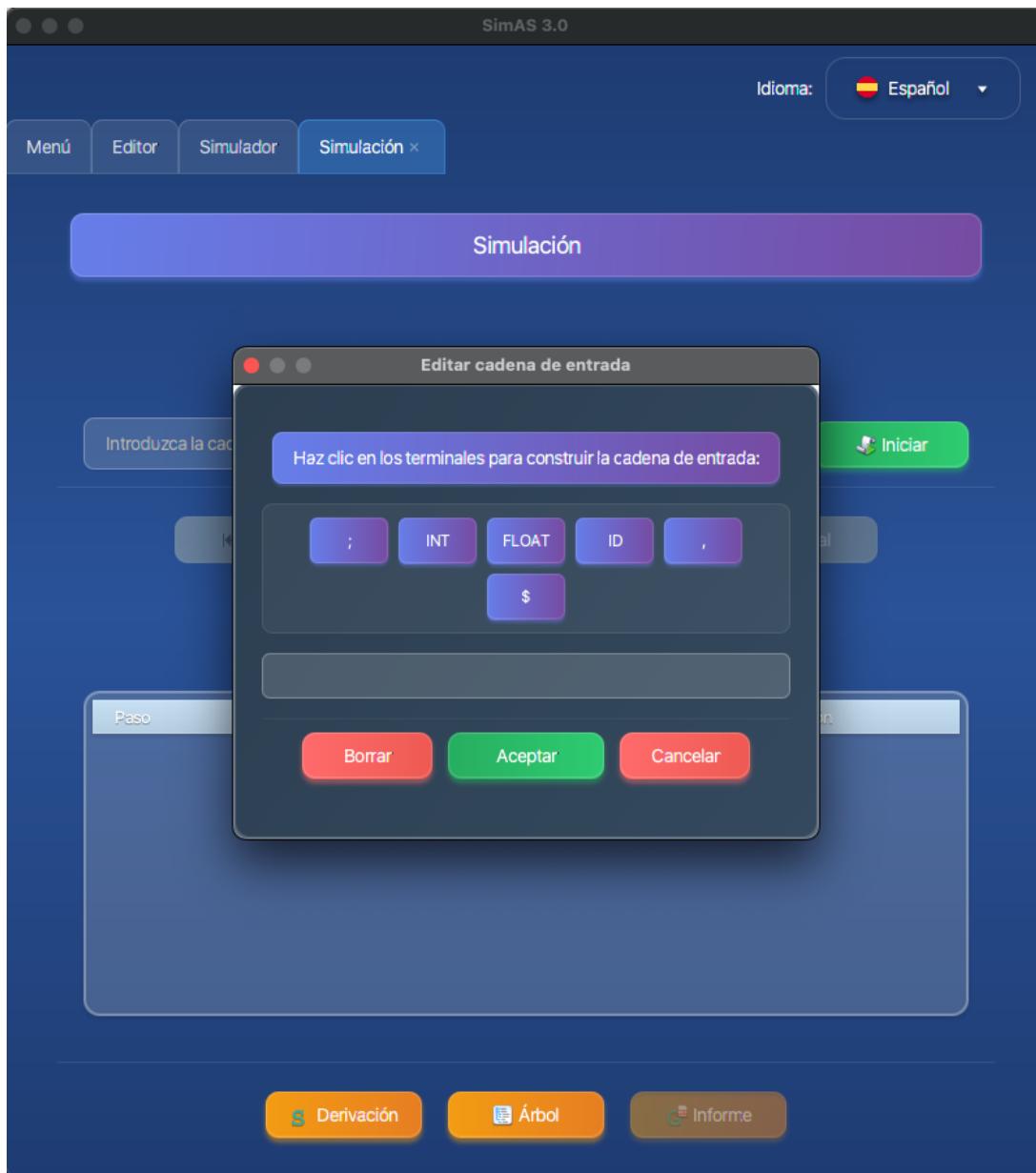


Figura 5.12: Interfaz de simulación con cadena de entrada

5.5.2. Proceso de simulación

El proceso de simulación implementa un algoritmo LL(1) completo que sigue una secuencia estructurada:

1. **Entrada y validación de cadena:** el usuario introduce la cadena a analizar, y el sistema valida que contenga únicamente símbolos terminales válidos para la gramática.
2. **Inicialización del analizador:** se configura la pila de análisis con el símbolo inicial y se prepara el entorno de simulación.

3. **Ejecución del algoritmo LL(1)**: el analizador procesa la cadena utilizando la tabla predictiva, aplicando producciones y funciones de error según sea necesario.
4. **Control interactivo**: el usuario mantiene control total sobre el avance de la simulación, pudiendo pausar, avanzar paso a paso o retroceder en cualquier momento.
5. **Resultado final**: el sistema determina si la cadena es aceptada o rechazada, proporcionando detalles completos del proceso y estadísticas del análisis.

5.5.3. Historial de pasos

El sistema de historial, ilustrado en la figura 5.13, proporciona una vista detallada de cada paso del análisis:

- **Pila de análisis**: representación visual del estado actual de la pila, mostrando los símbolos pendientes de procesar.
- **Acción aplicada**: documentación clara de la producción o función de error utilizada en cada paso, con justificación de la decisión.
- **Estado del análisis**: información comprensiva sobre el progreso actual, incluyendo posición en la entrada y estado de la pila.
- **Métricas de rendimiento**: estadísticas en tiempo real sobre el número de pasos, eficiencia del análisis y uso de funciones de error.

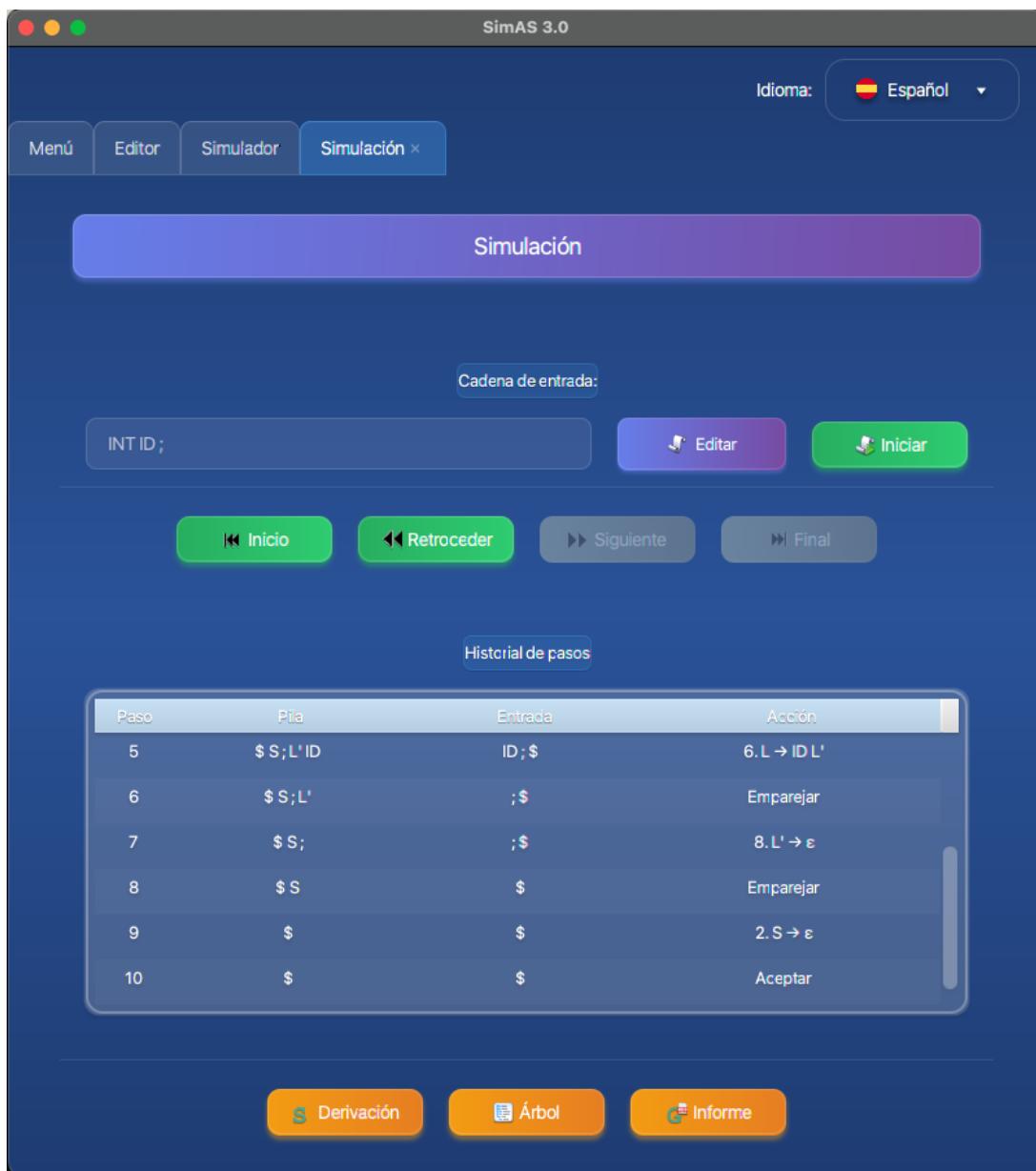


Figura 5.13: Simulación completa con historial de pasos

5.5.4. Navegación en la simulación

El sistema de navegación proporciona control granular sobre la simulación:

- **Avance paso a paso:** ejecuta un único paso del análisis, permitiendo al usuario examinar cada decisión en detalle.
- **Retroceso inteligente:** permite volver al paso anterior manteniendo la integridad del estado de la simulación.
- **Finalización automática:** salta al final de la simulación para obtener el resultado inmediatamente.

- **Reinicio completo:** restaura la simulación a su estado inicial, permitiendo comenzar de nuevo con la misma cadena o una nueva.

5.6. Visualización de resultados

El simulador incorpora un sistema avanzado de visualización que proporciona dos tipos complementarios de representación que se actualizan dinámicamente durante la simulación, facilitando la comprensión del proceso de análisis sintáctico.

5.6.1. Derivación

La derivación muestra la secuencia de producciones aplicadas durante el análisis, como se puede observar en la figura 5.14:

- **Acceso directo:** botón "Derivación" en la pestaña de simulación que abre una nueva pestaña dedicada.
- **Actualización automática:** se actualiza en tiempo real conforme avanza la simulación, mostrando cada paso de derivación.
- **Formato BNF estándar:** representación clara y estándar de cada paso de derivación, facilitando la comprensión del proceso.
- **Navegación sincronizada:** los cambios en la derivación reflejan exactamente la posición actual en la simulación, manteniendo la coherencia.

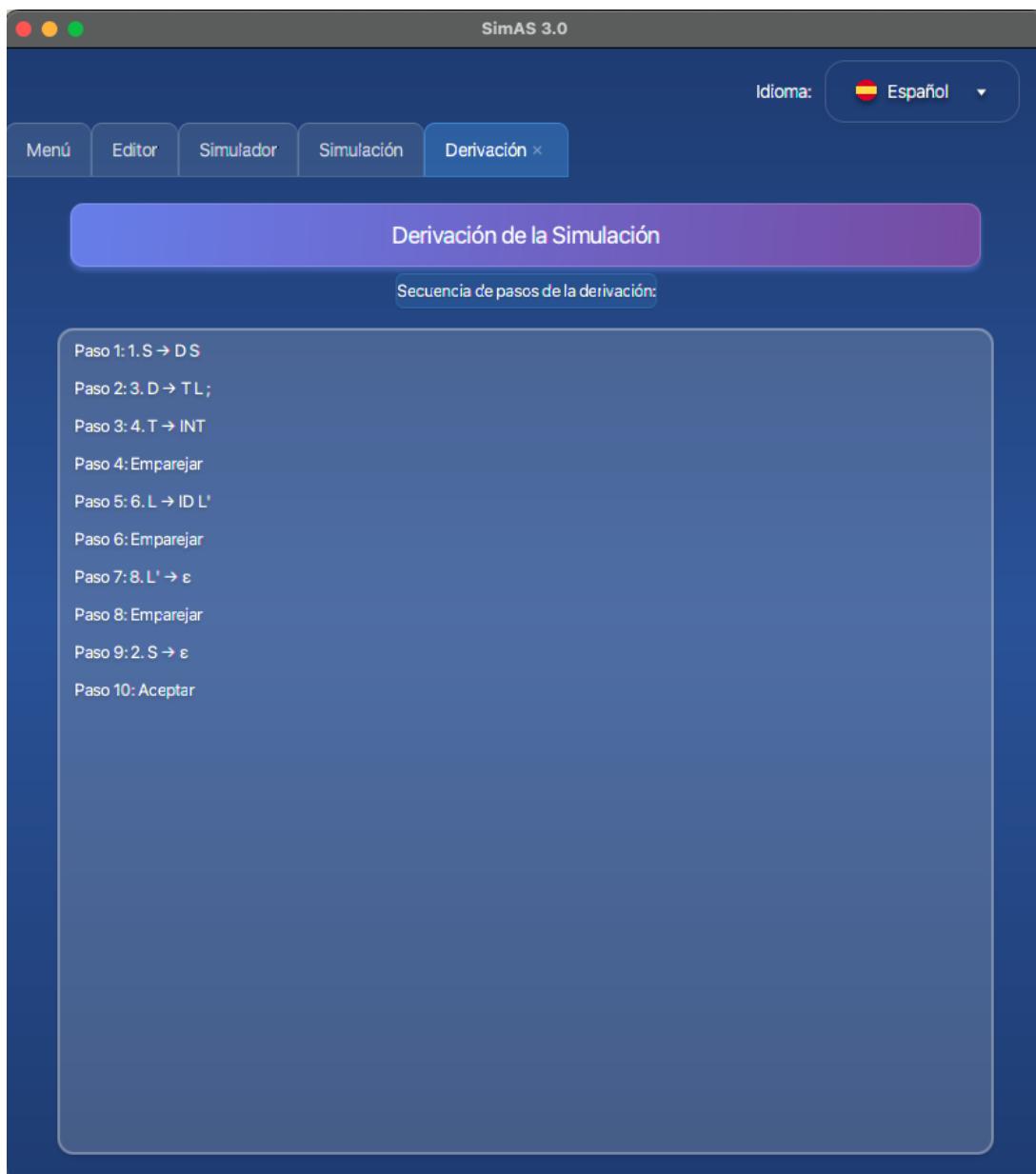


Figura 5.14: Vista de la derivación durante la simulación

5.6.2. Árbol sintáctico

El árbol sintáctico proporciona una representación visual de la estructura de la cadena, ilustrado en la figura 5.15:

- **Acceso directo:** botón "Árbol" en la pestaña de simulación que abre una nueva pestaña dedicada.
- **Construcción progresiva:** se construye paso a paso durante la simulación, mostrando el crecimiento del árbol en tiempo real.
- **Representación gráfica avanzada:** estructura jerárquica clara y legible con elementos visuales que facilitan la comprensión.

- **Sincronización perfecta:** refleja exactamente el estado actual de la simulación, manteniendo la coherencia con la derivación.

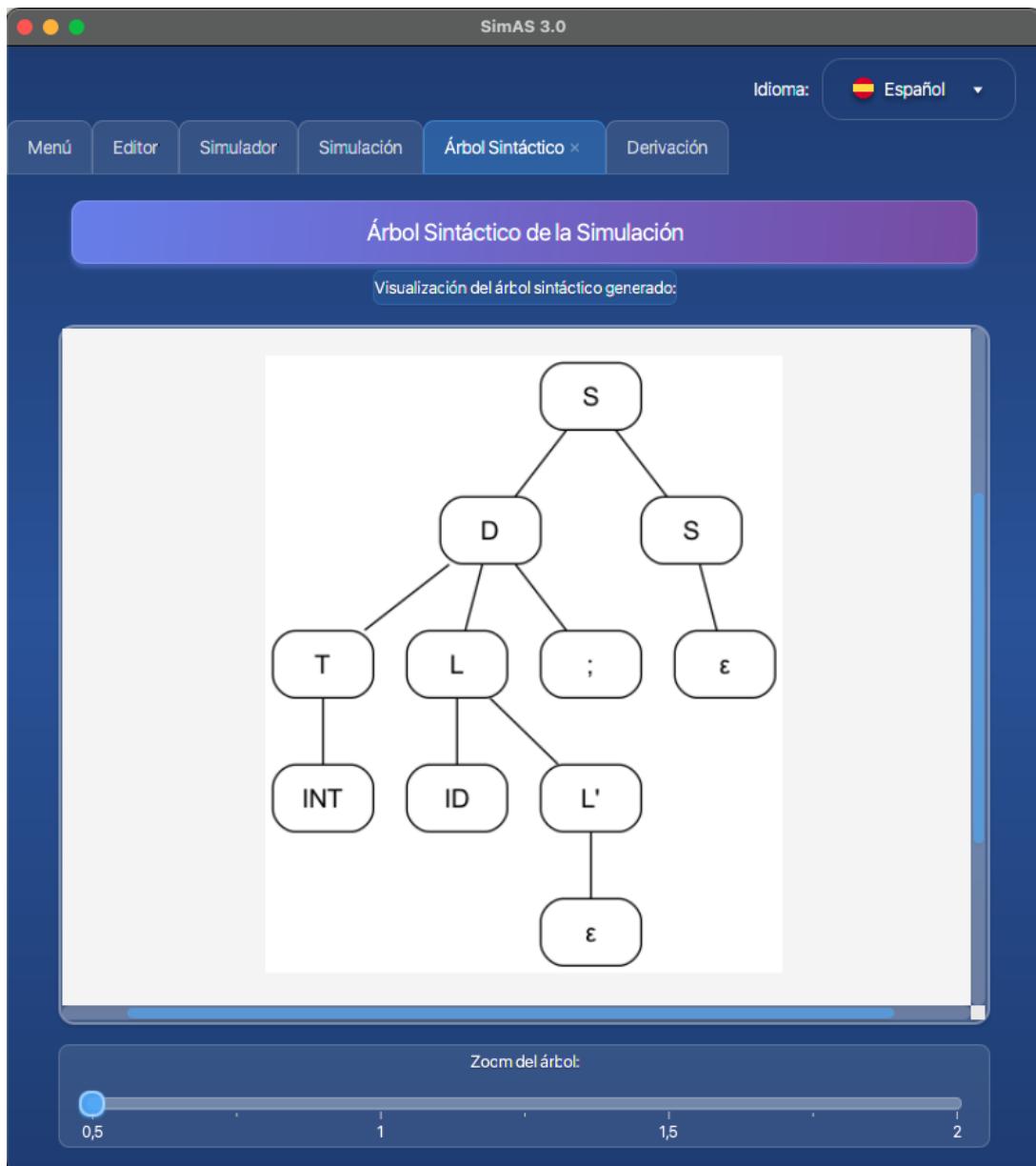


Figura 5.15: Vista del árbol sintáctico durante la simulación

5.7. Múltiples simulaciones

Una característica avanzada y distintiva del simulador es la capacidad de manejar múltiples simulaciones simultáneamente, proporcionando una experiencia de análisis comparativo y exhaustivo que no se encuentra en herramientas similares.

5.7.1. Gestión de simulaciones

Para cada configuración de simulador, el sistema permite una gestión sofisticada:

- **Crear nuevas simulaciones:** capacidad de simular diferentes cadenas de entrada

manteniendo la misma configuración de gramática y funciones de error.

- **Mantener simulaciones activas:** múltiples simulaciones pueden estar abiertas simultáneamente, cada una en su propia pestaña independiente.
- **Comparar resultados:** análisis comparativo de diferentes cadenas con la misma gramática, facilitando la identificación de patrones y comportamientos.
- **Gestión independiente:** cada simulación mantiene su propio estado, historial y visualizaciones, permitiendo análisis paralelos sin interferencias.

5.7.2. Ventajas de múltiples simulaciones

Esta funcionalidad avanzada, mostrada en la figura 5.16, permite:

- **Análisis comparativo avanzado:** probar diferentes cadenas con la misma gramática, facilitando la identificación de patrones y comportamientos específicos.
- **Validación exhaustiva:** verificar el comportamiento del analizador con múltiples casos de prueba, incluyendo casos límite y escenarios complejos.
- **Experimentos educativos:** explorar diferentes escenarios de análisis, permitiendo a los estudiantes comprender mejor los conceptos teóricos.
- **Eficiencia operativa:** no es necesario reconfigurar el simulador para cada nueva cadena, optimizando el flujo de trabajo y la productividad.

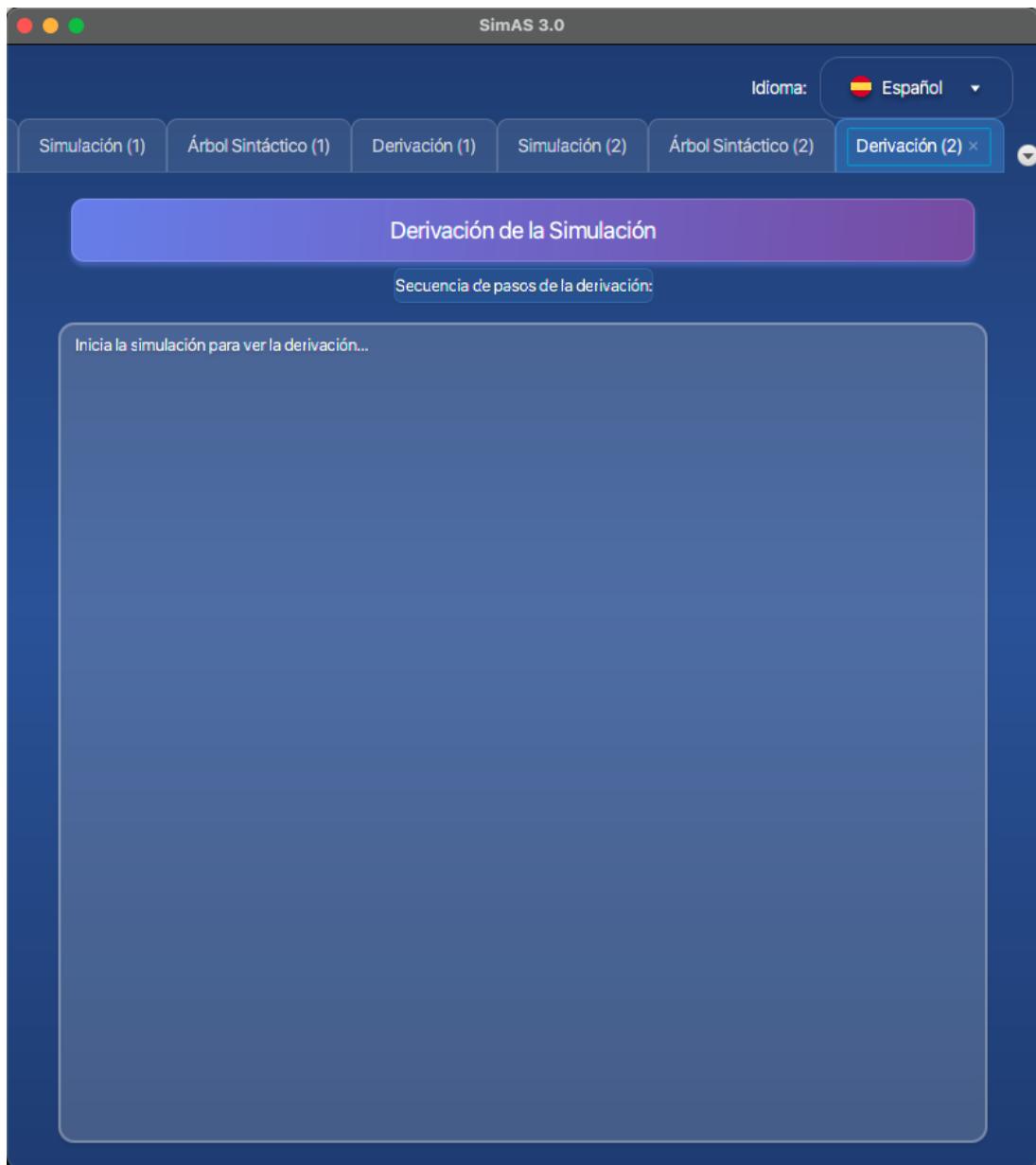


Figura 5.16: Ejemplo de múltiples simulaciones simultáneas

5.8. Integración con el editor

El simulador mantiene una integración profunda y transparente con el editor de gramáticas, creando un flujo de trabajo unificado y eficiente:

- **Transición automática:** paso directo y sin interrupciones del editor al simulador, manteniendo toda la configuración y estado.
- **Sincronización de datos en tiempo real:** cambios realizados en el editor se reflejan inmediatamente en el simulador, garantizando la coherencia.
- **Consistencia garantizada:** el sistema asegura que la gramática simulada es exactamente la misma que la editada, eliminando errores de configuración.

- **Flujo de trabajo optimizado:** proceso continuo y fluido desde la creación y edición de gramáticas hasta su simulación y análisis.

5.9. Resolución de problemas

El simulador incluye un sistema robusto de detección y resolución de problemas que ayuda a los usuarios a identificar y solucionar dificultades comunes durante el proceso de simulación.

5.9.1. Problemas comunes

El sistema está diseñado para manejar y resolver automáticamente los siguientes problemas típicos:

- **Gramática no LL(1):** el sistema detecta automáticamente conflictos en la tabla predictiva y proporciona sugerencias específicas para resolverlos.
- **Recursividad no eliminable:** se informa al usuario sobre limitaciones estructurales y se sugieren alternativas de diseño.
- **Funciones de error incorrectas:** validación automática previene errores de configuración y proporciona retroalimentación detallada.
- **Cadenas inválidas:** el sistema valida la entrada antes de la simulación y proporciona información específica sobre símbolos no válidos.

5.9.2. Soluciones recomendadas

Para resolver problemas comunes, se recomiendan las siguientes estrategias:

- **Revisar la gramática:** verificar que la gramática sea LL(1) utilizando las herramientas de validación del editor.
- **Ajustar funciones de error:** modificar o eliminar funciones problemáticas utilizando el panel de gestión del paso 4.
- **Validar entrada:** asegurar que la cadena contenga únicamente símbolos terminales válidos para la gramática actual.
- **Consultar la documentación:** revisar ejemplos y casos de uso para comprender mejor el comportamiento esperado.

5.10. Conclusión

El simulador de análisis sintáctico descendente predictivo de SimAS 3.0 representa una herramienta educativa de vanguardia que combina la potencia algorítmica del análisis LL(1) con una interfaz intuitiva y educativa de última generación. El asistente guiado de 5 pasos simplifica significativamente la configuración compleja requerida para el análisis sintáctico, mientras que las funciones avanzadas de visualización y múltiples simulaciones proporcionan una experiencia de aprendizaje rica, completa y profundamente educativa.

La integración perfecta y transparente con el editor de gramáticas, el sistema avanzado de recuperación de errores, y la capacidad de generar informes detallados y profesionales hacen del simulador una herramienta indispensable para el estudio, la práctica y la investigación del análisis sintáctico. La funcionalidad innovadora de múltiples simulaciones permite una exploración exhaustiva y comparativa de las capacidades de la gramática, mientras que la navegación paso a paso y las visualizaciones dinámicas facilitan la comprensión profunda de los conceptos teóricos subyacentes.

El simulador no solo demuestra la corrección y eficiencia de las gramáticas, sino que también educa de manera integral sobre el proceso completo de análisis sintáctico, proporcionando una base sólida y comprensiva para el aprendizaje de conceptos avanzados en teoría de lenguajes formales, compiladores y análisis sintáctico. Su diseño educativo y su implementación técnica robusta lo convierten en una herramienta de referencia para la enseñanza y el aprendizaje de estos conceptos fundamentales en ciencias de la computación.

Capítulo 6

Ejemplos prácticos

6.1. Introducción

Este capítulo presenta un ejemplo práctico completo que demuestra el uso de SimAS 3.0 para crear una gramática de contexto libre y realizar un análisis sintáctico descendente. El ejemplo seguirá todo el flujo de trabajo desde la creación de la gramática hasta la simulación del análisis, mostrando las capacidades completas de la aplicación.

El ejemplo utiliza una gramática para declaraciones de variables que permite comprender los conceptos fundamentales del análisis sintáctico descendente, incluyendo la construcción de conjuntos PRIMERO y SIGUIENTE, la generación de tablas predictivas, y la simulación paso a paso del análisis.

6.2. Creación de una gramática de contexto libre

En esta sección se detalla el proceso completo de creación de una gramática utilizando el editor de SimAS 3.0. El ejemplo utiliza una gramática para declaraciones de variables que permite definir variables de tipo entero y flotante.

6.2.1. Acceso al editor

Para comenzar, accedemos al editor de gramáticas desde el menú principal, como se muestra en la figura 6.1.

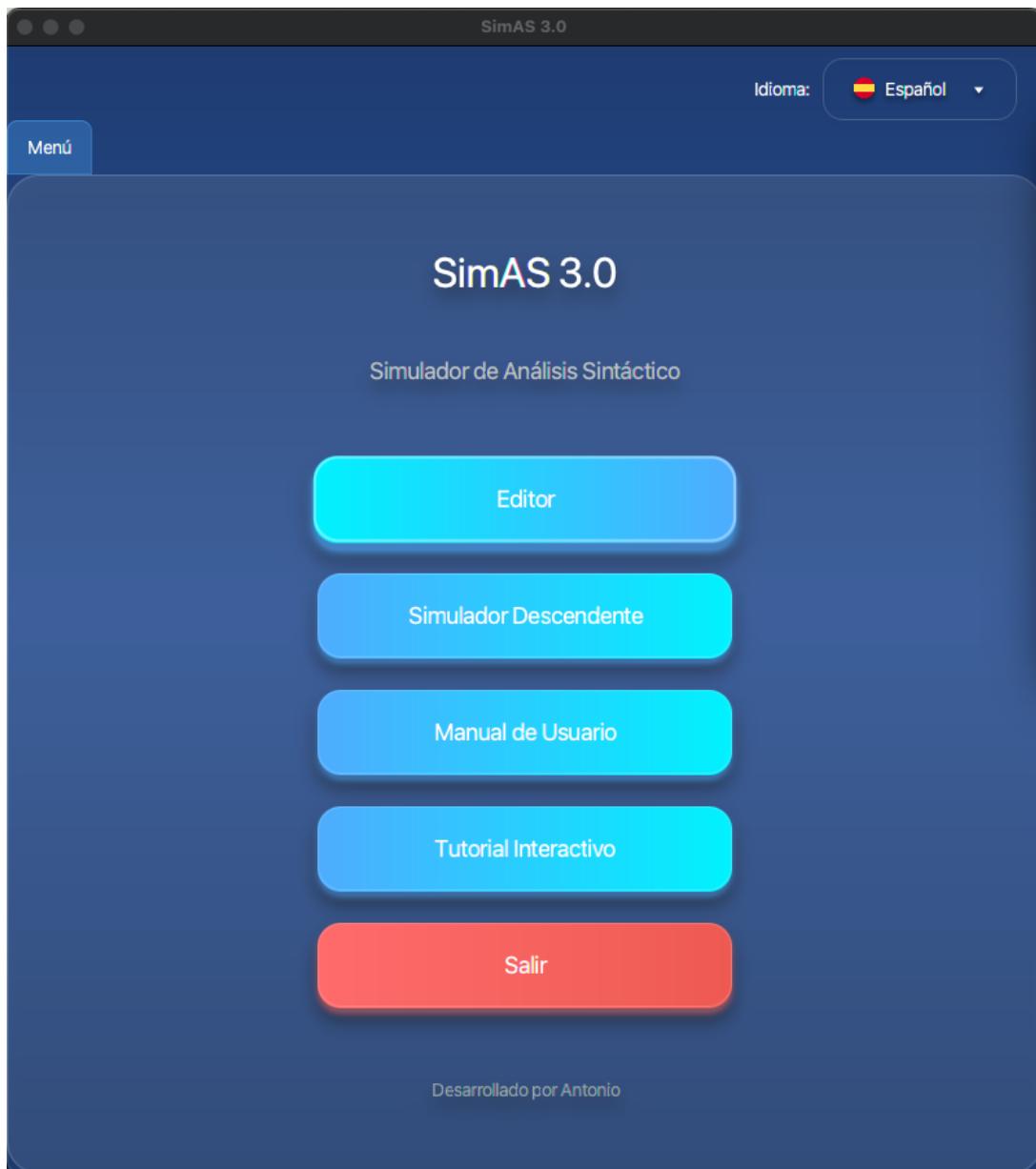


Figura 6.1: Acceso al editor desde el menú principal

6.2.2. Paso 1: Datos de la gramática

El primer paso consiste en definir los datos básicos de la gramática. En la figura 6.2 se muestra la configuración inicial:

6.2. CREACIÓN DE UNA GRAMÁTICA DE CONTEXTO LIBRE

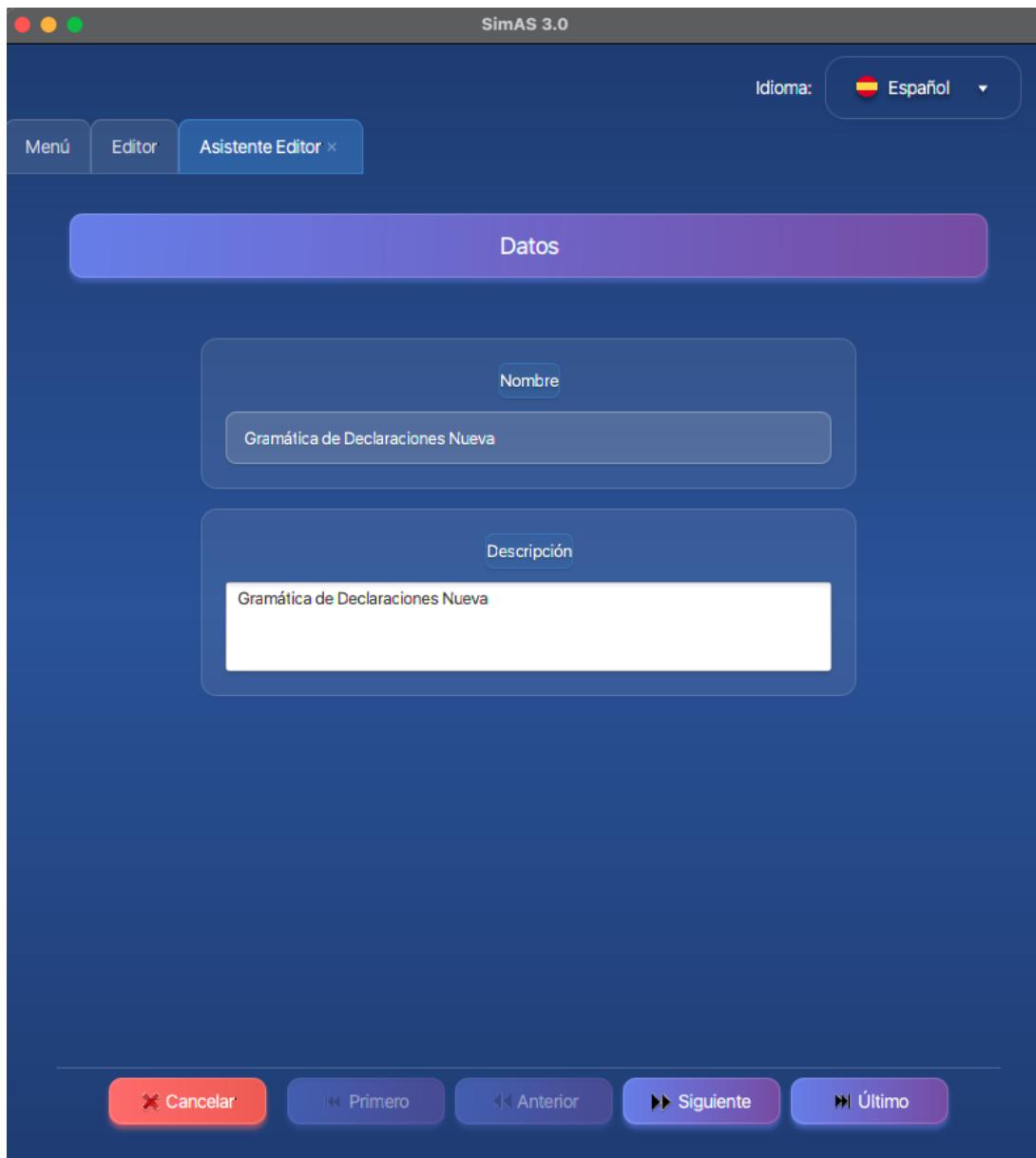


Figura 6.2: Paso 1: Configuración de datos básicos de la gramática

En este paso se define:

- **Nombre:** "Gramática de Declaraciones Nueva"
- **Descripción:** "Gramática de Declaraciones Nueva"

6.2.3. Paso 2: Símbolos terminales y no terminales

El segundo paso implica la definición de todos los símbolos que formarán parte de la gramática. La figura 6.3 muestra la configuración de símbolos:

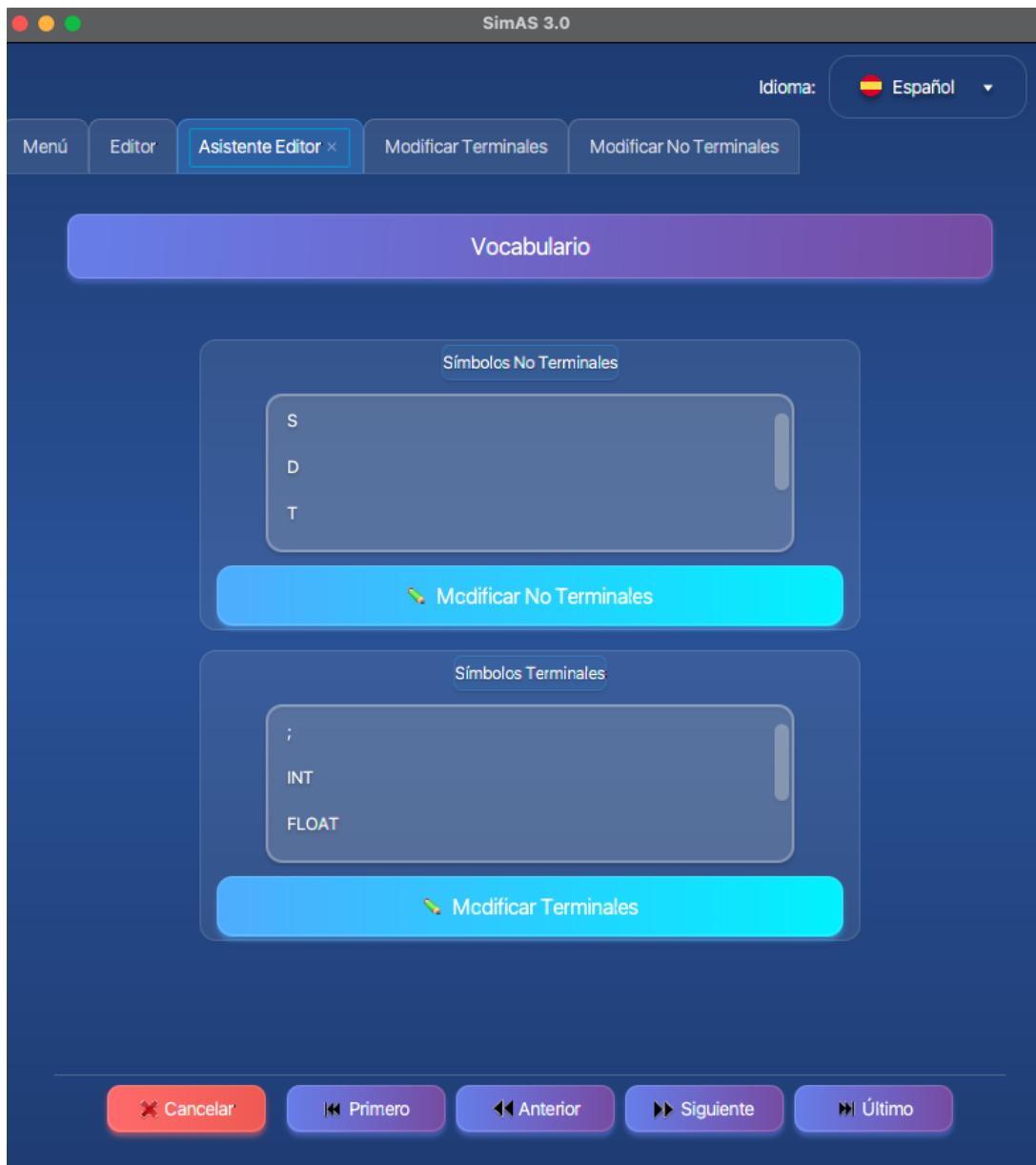


Figura 6.3: Paso 2: Definición de símbolos terminales y no terminales

6.2.3.1. Símbolos terminales

Los símbolos terminales definidos incluyen:

- **Tipos de datos:** INT, FLOAT
- **Identificadores:** ID (representando nombres de variables)
- **Separadores:** , (coma), ; (punto y coma)

La figura 6.4 muestra el panel auxiliar para la gestión de símbolos terminales:

6.2. CREACIÓN DE UNA GRAMÁTICA DE CONTEXTO LIBRE

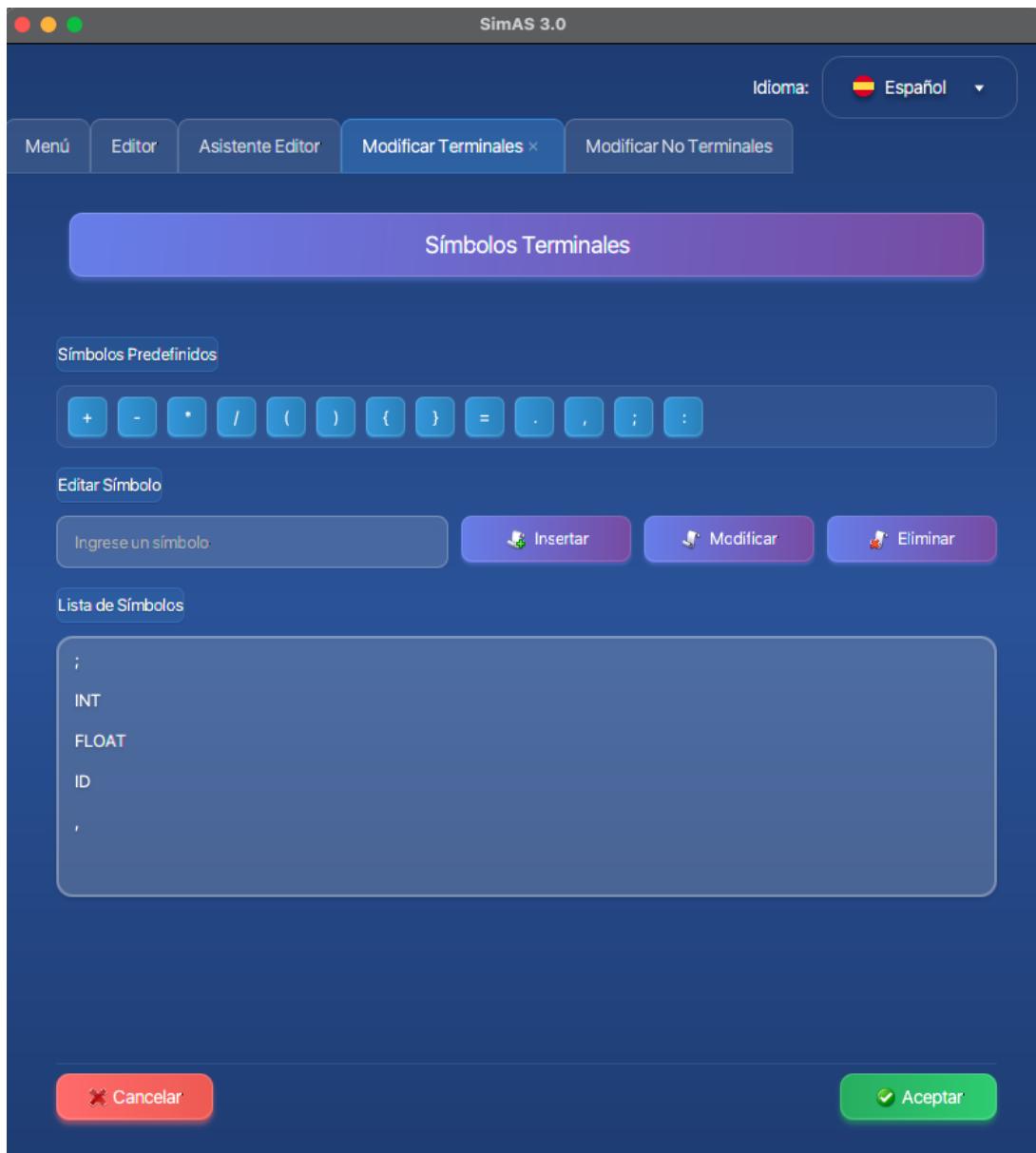


Figura 6.4: Panel auxiliar para gestión de símbolos terminales

6.2.3.2. Símbolos no terminales

Los símbolos no terminales definidos son:

- **S**: Secuencia de declaraciones
- **D**: Declaración individual
- **T**: Tipo de dato
- **L**: Lista de identificadores
- **L'**: Lista de identificadores (recursiva)

La figura 6.5 muestra el panel auxiliar para la gestión de símbolos no terminales:

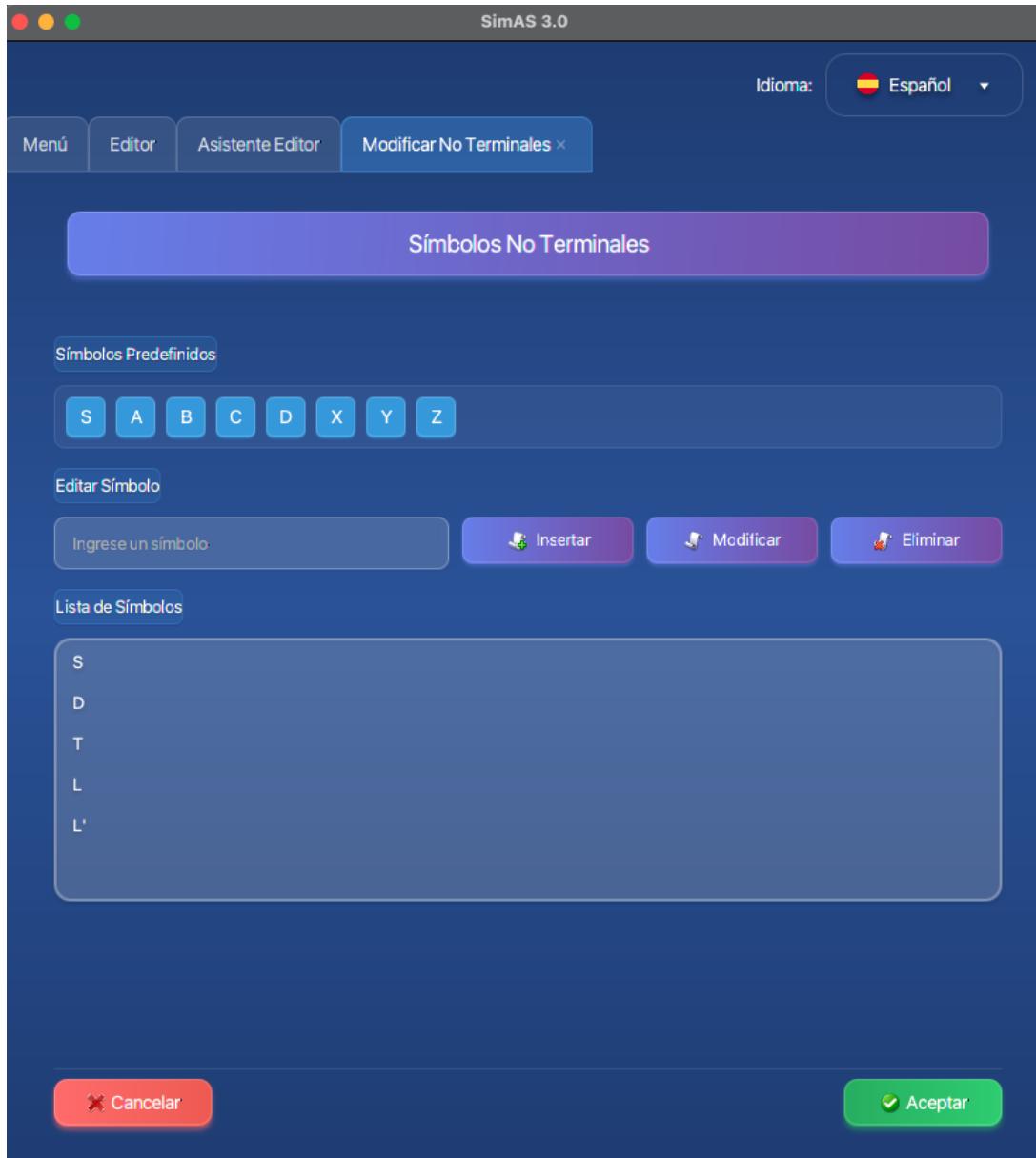


Figura 6.5: Panel auxiliar para gestión de símbolos no terminales

6.2.4. Paso 3: Producciones

El tercer paso consiste en definir las reglas de producción de la gramática. La figura 6.6 muestra la configuración de producciones:

6.2. CREACIÓN DE UNA GRAMÁTICA DE CONTEXTO LIBRE

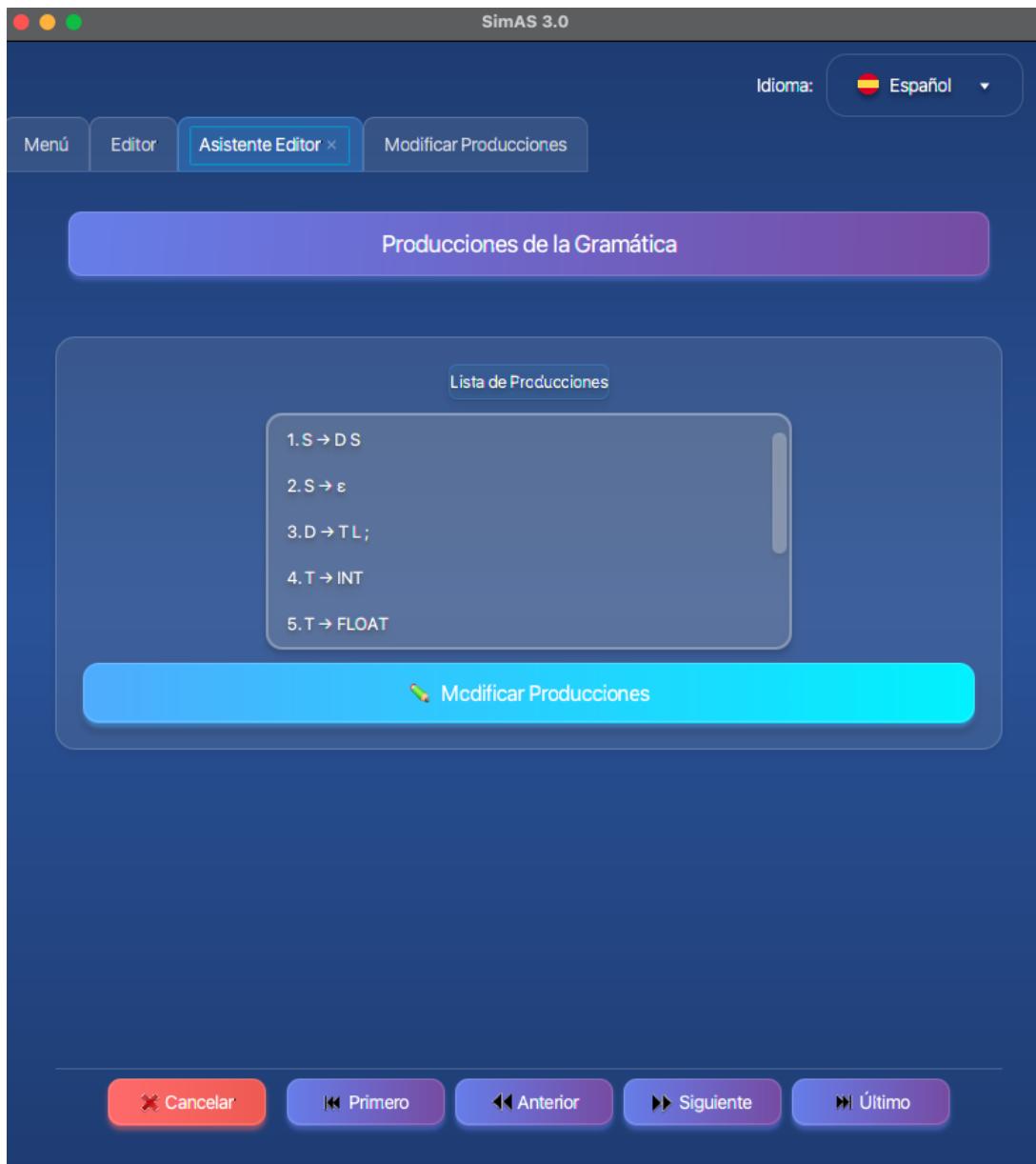


Figura 6.6: Paso 3: Definición de reglas de producción

Las producciones definidas para esta gramática son:

- $S \rightarrow D S \mid \epsilon$
- $D \rightarrow T L ;$
- $T \rightarrow INT \mid FLOAT$
- $L \rightarrow ID L'$
- $L' \rightarrow , ID L' \mid \epsilon$

La figura 6.7 muestra el panel auxiliar para la gestión de producciones:

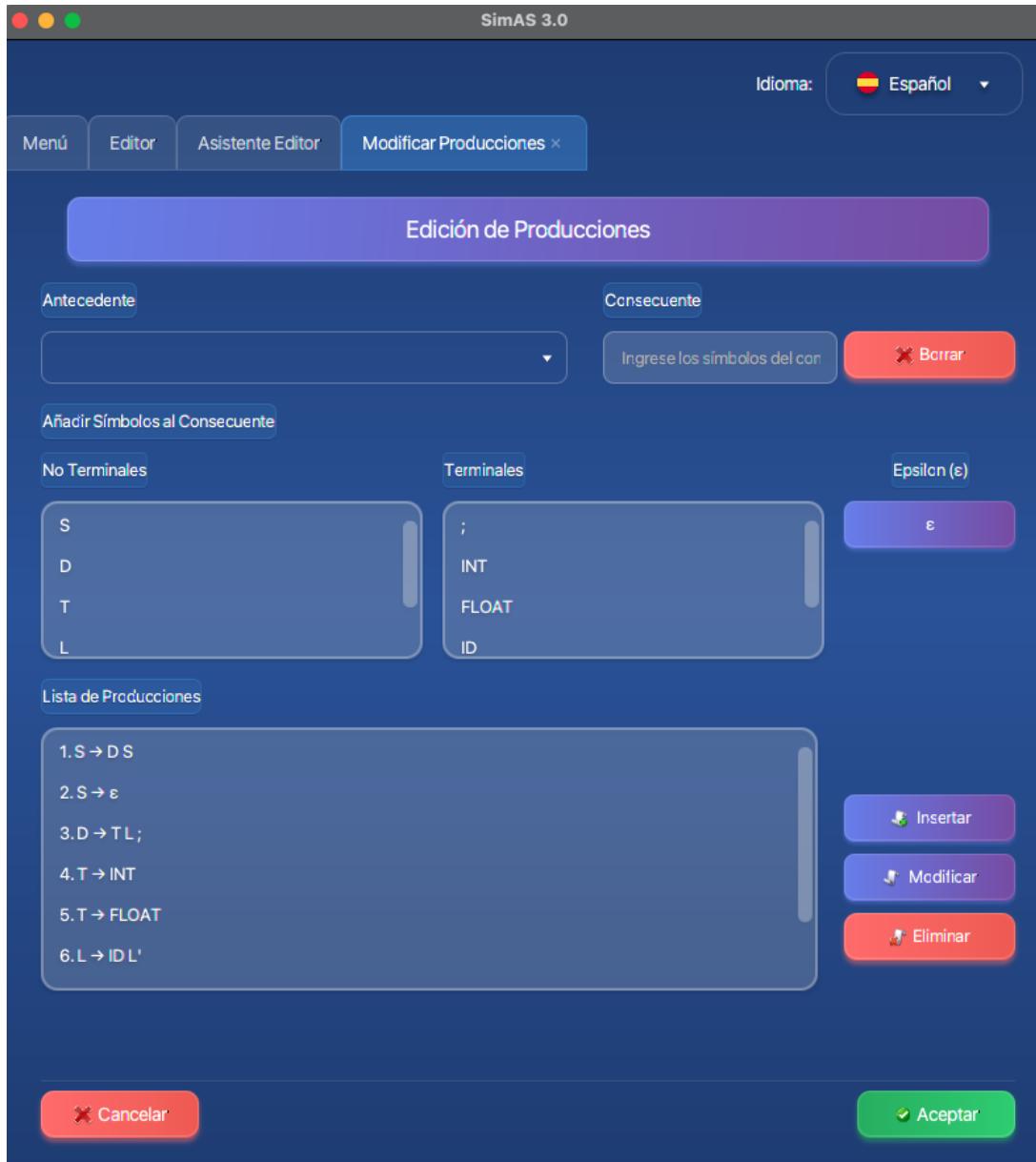


Figura 6.7: Panel auxiliar para gestión de producciones

6.2.5. Paso 4: Símbolo inicial

El cuarto y último paso del asistente de creación consiste en seleccionar el símbolo inicial de la gramática. La figura 6.8 muestra esta configuración:

6.2. CREACIÓN DE UNA GRAMÁTICA DE CONTEXTO LIBRE

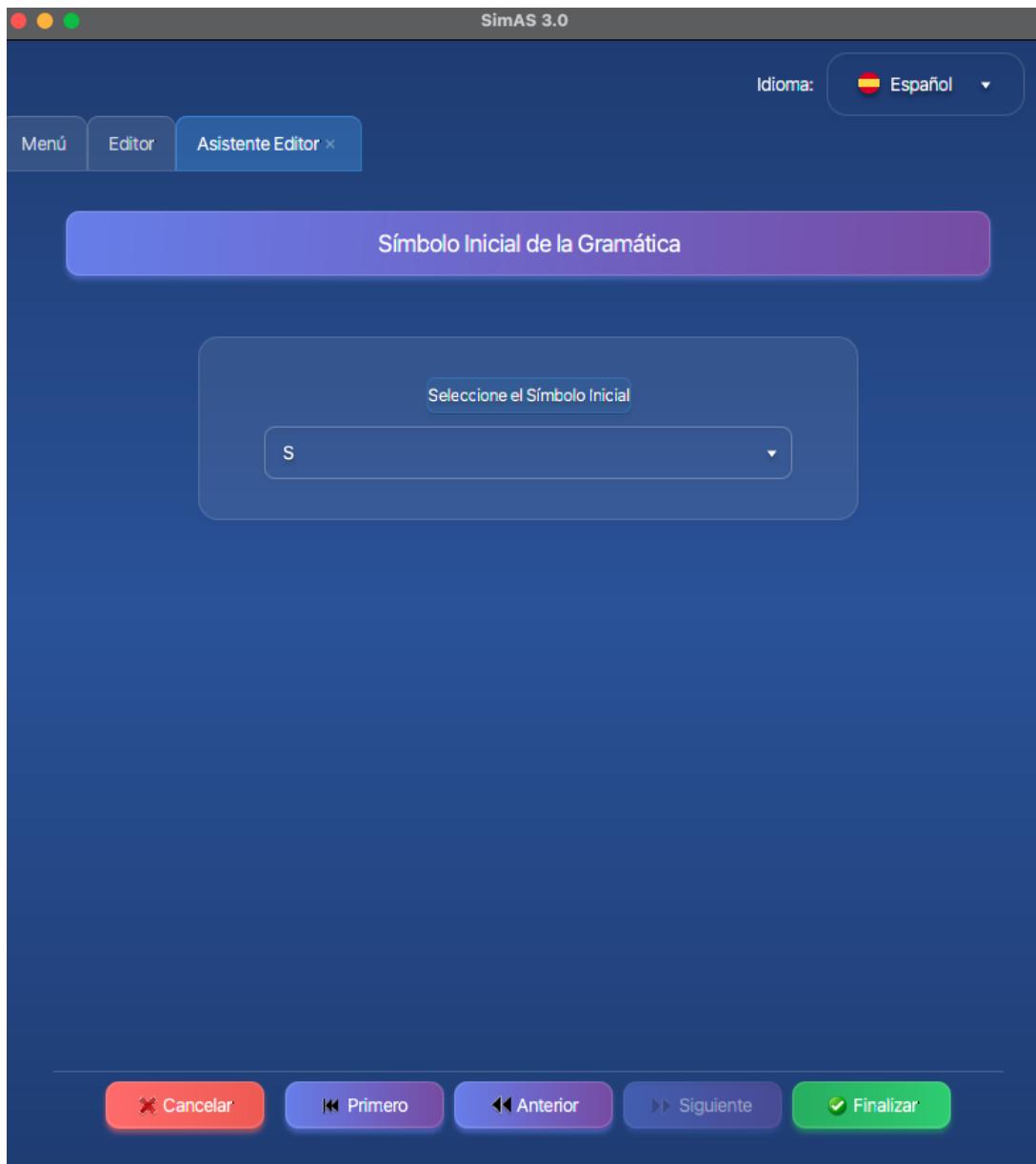


Figura 6.8: Paso 4: Selección del símbolo inicial

En este caso, se selecciona "S" como símbolo inicial, ya que representa la secuencia de declaraciones de la gramática.

6.2.6. Validación de la gramática

Una vez completados todos los pasos, es necesario validar la gramática para asegurar que está correctamente definida. La figura 6.9 muestra el resultado de la validación:



Figura 6.9: Validación exitosa de la gramática

La validación confirma que la gramática está correctamente definida y lista para ser utilizada en el simulador.

6.2.7. Vista final del editor

La figura 6.10 muestra la vista final del editor con la gramática completamente definida:

6.3. SIMULACIÓN DEL ANÁLISIS SINTÁCTICO DESCENDENTE



Figura 6.10: Vista final del editor con la gramática completa

6.3. Simulación del análisis sintáctico descendente

Una vez creada y validada la gramática, procedemos a utilizar el simulador de análisis sintáctico descendente para analizar cadenas de entrada.

6.3.1. Acceso al simulador

El simulador puede accederse desde el editor utilizando el botón "Simular", que lleva directamente al paso 1 del asistente de simulación.

6.3.2. Paso 1: Refactorización de la gramática

El primer paso del asistente de simulación consiste en refactorizar la gramática para eliminar la recursividad por la izquierda y factorizar las producciones. La figura 6.11 muestra este proceso:

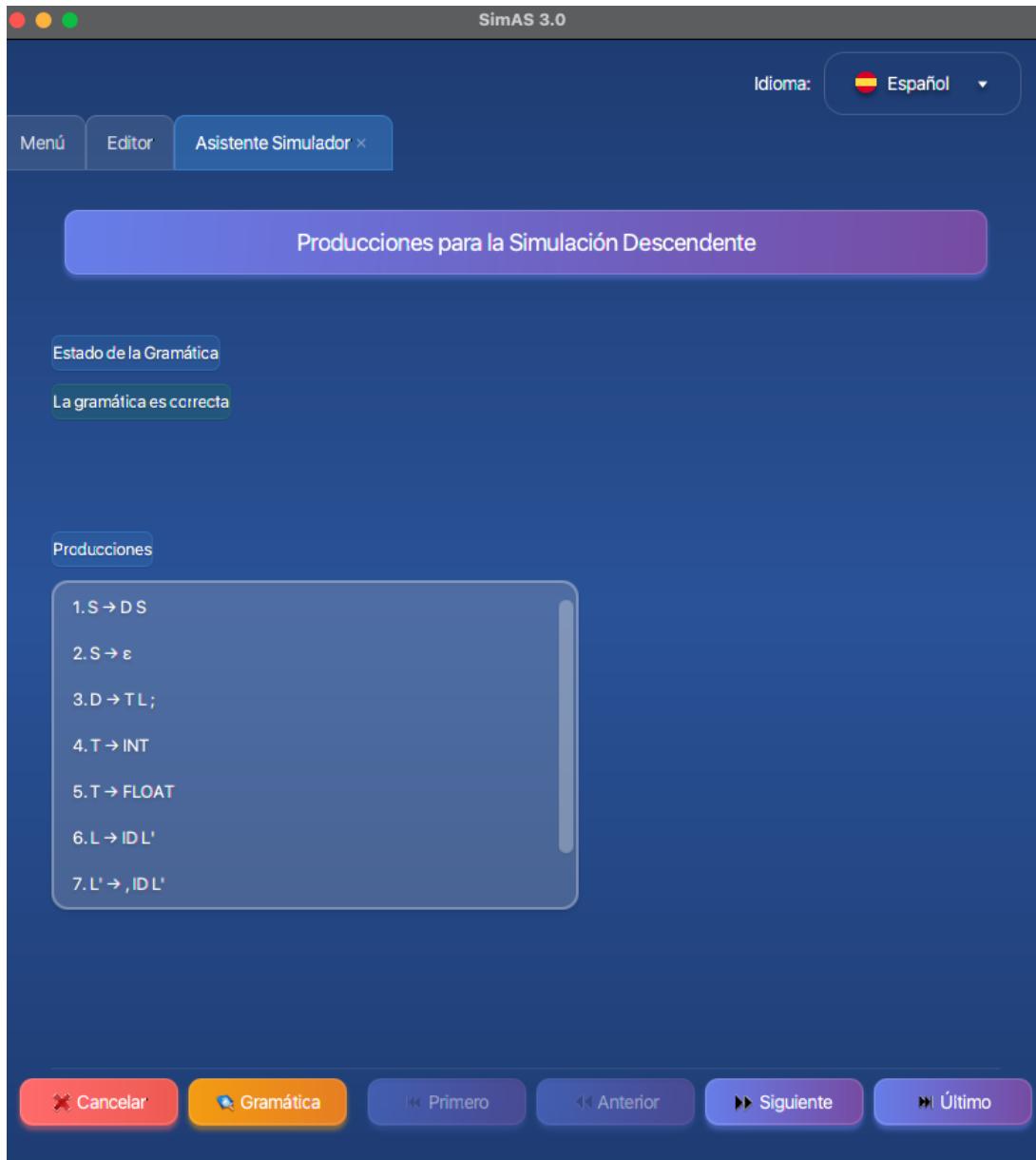


Figura 6.11: Paso 1: Refactorización de la gramática

La gramática original se transforma para eliminar la recursividad por la izquierda, resultando en:

- $S \rightarrow D S \mid \epsilon$
- $D \rightarrow T L ;$
- $T \rightarrow INT \mid FLOAT$

- $L \rightarrow ID \ L'$
- $L' \rightarrow , \ ID \ L' \mid \epsilon$

En este caso, la gramática ya está en forma LL(1) y no requiere refactorización adicional, ya que no presenta recursividad por la izquierda.

6.3.3. Paso 2: Construcción de conjuntos PRIMERO y SIGUIENTE

El segundo paso construye automáticamente los conjuntos PRIMERO y SIGUIENTE para cada símbolo no terminal. La figura 6.12 muestra estos conjuntos:

Símbolo	Conjunto Primero	Conjunto Siguiente
S	FLOAT ϵ INT	\$
D	FLOAT INT	FLOAT \$ INT
T	FLOAT INT	ID
L	ID	;
L'	ϵ ,	;

Figura 6.12: Paso 2: Conjuntos PRIMERO y SIGUIENTE

Los conjuntos calculados son:

Conjuntos PRIMERO:

- $\text{PRIMERO}(S) = \{ \text{INT}, \text{FLOAT}, \varepsilon \}$
- $\text{PRIMERO}(D) = \{ \text{INT}, \text{FLOAT} \}$
- $\text{PRIMERO}(T) = \{ \text{INT}, \text{FLOAT} \}$
- $\text{PRIMERO}(L) = \{ \text{ID} \}$
- $\text{PRIMERO}(L') = \{ , \varepsilon \}$

Conjuntos SIGUIENTE:

- $\text{SIGUIENTE}(S) = \{ \$ \}$
- $\text{SIGUIENTE}(D) = \{ \text{INT}, \text{FLOAT}, \$ \}$
- $\text{SIGUIENTE}(T) = \{ \text{ID} \}$
- $\text{SIGUIENTE}(L) = \{ ; \}$
- $\text{SIGUIENTE}(L') = \{ ; \}$

6.3.4. Paso 3: Construcción de la tabla predictiva

El tercer paso genera la tabla predictiva basada en los conjuntos PRIMERO y SIGUIENTE. La figura 6.13 muestra la tabla resultante:

6.3. SIMULACIÓN DEL ANÁLISIS SINTÁCTICO DESCENDENTE

The screenshot shows the SimAS 3.0 application window. At the top, there are three tabs: 'Menú' (selected), 'Editor', and 'Asistente Simulador'. The 'Asistente Simulador' tab is highlighted with a blue border. In the top right corner, there is a language selection dropdown set to 'Español'. Below the tabs, a purple header bar contains the text 'Tabla Predictiva'. Underneath this, a table titled 'Tabla de Análisis Predictivo' is displayed. The table has columns for Non-Terminal symbols (S, D, T, L, L') and terminals (INT, FLOAT, ID, \$). The entries in the table represent grammar rules:

No Terminal	;	INT	FLOAT	ID	,	\$
S		1. S → DS	1. S → DS			2. S → ε
D		3. D → TL;	3. D → TL;			
T		4. T → INT	5. T → FLO...			
L				6. L → ID L'		
L'		8. L' → ε			7. L' → , ID L'	

At the bottom of the window, there are several buttons: 'Cancelar' (Cancel) in red, 'Gramática' (Grammar) in orange, 'Primer' (First) with a left arrow, 'Anterior' (Previous) with a double-left arrow, 'Siguiente' (Next) with a double-right arrow, and 'Último' (Last) with a right arrow.

Figura 6.13: Paso 3: Tabla predictiva

6.3.5. Paso 4: Funciones de error

El cuarto paso permite configurar funciones de error para el manejo de cadenas incorrectas. La figura 6.14 muestra la configuración de funciones de error:

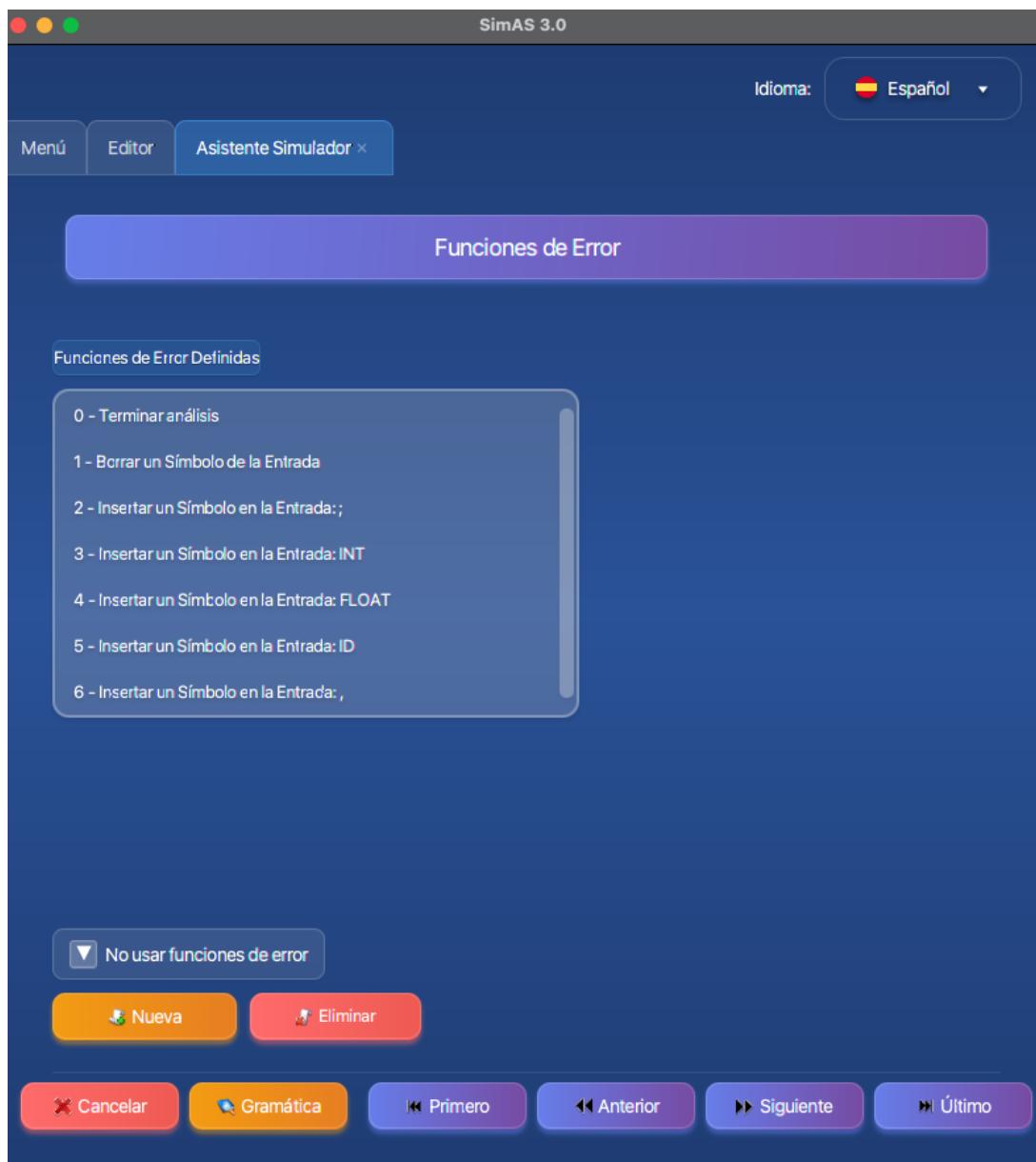


Figura 6.14: Paso 4: Configuración de funciones de error

La figura 6.15 muestra el panel auxiliar para añadir nuevas funciones de error:

6.3. SIMULACIÓN DEL ANÁLISIS SINTÁCTICO DESCENDENTE

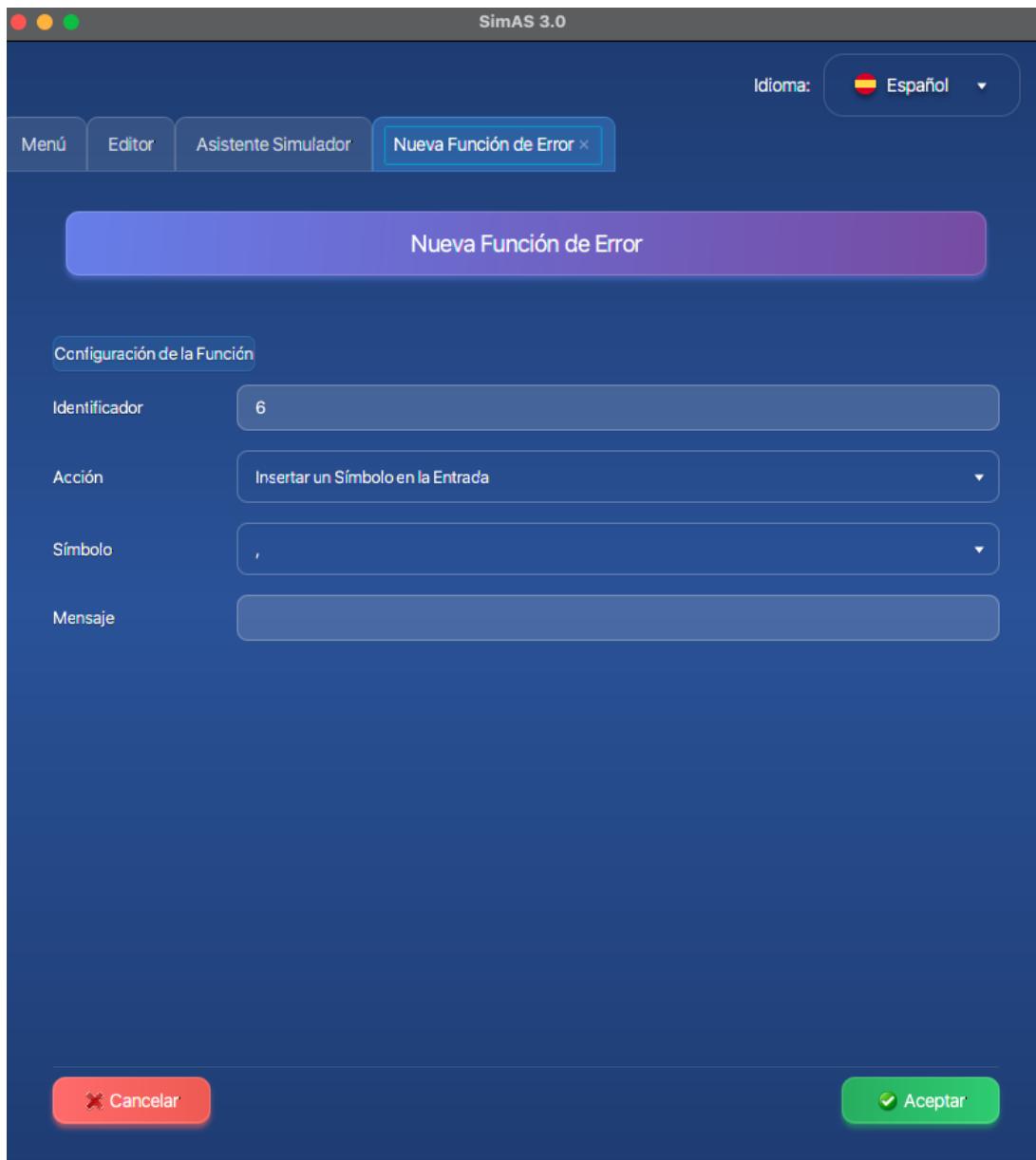


Figura 6.15: Panel auxiliar para añadir funciones de error

6.3.6. Paso 5: Tabla predictiva completa

El quinto paso presenta la tabla predictiva completa con todas las funciones de error configuradas. La figura 6.16 muestra esta tabla final:

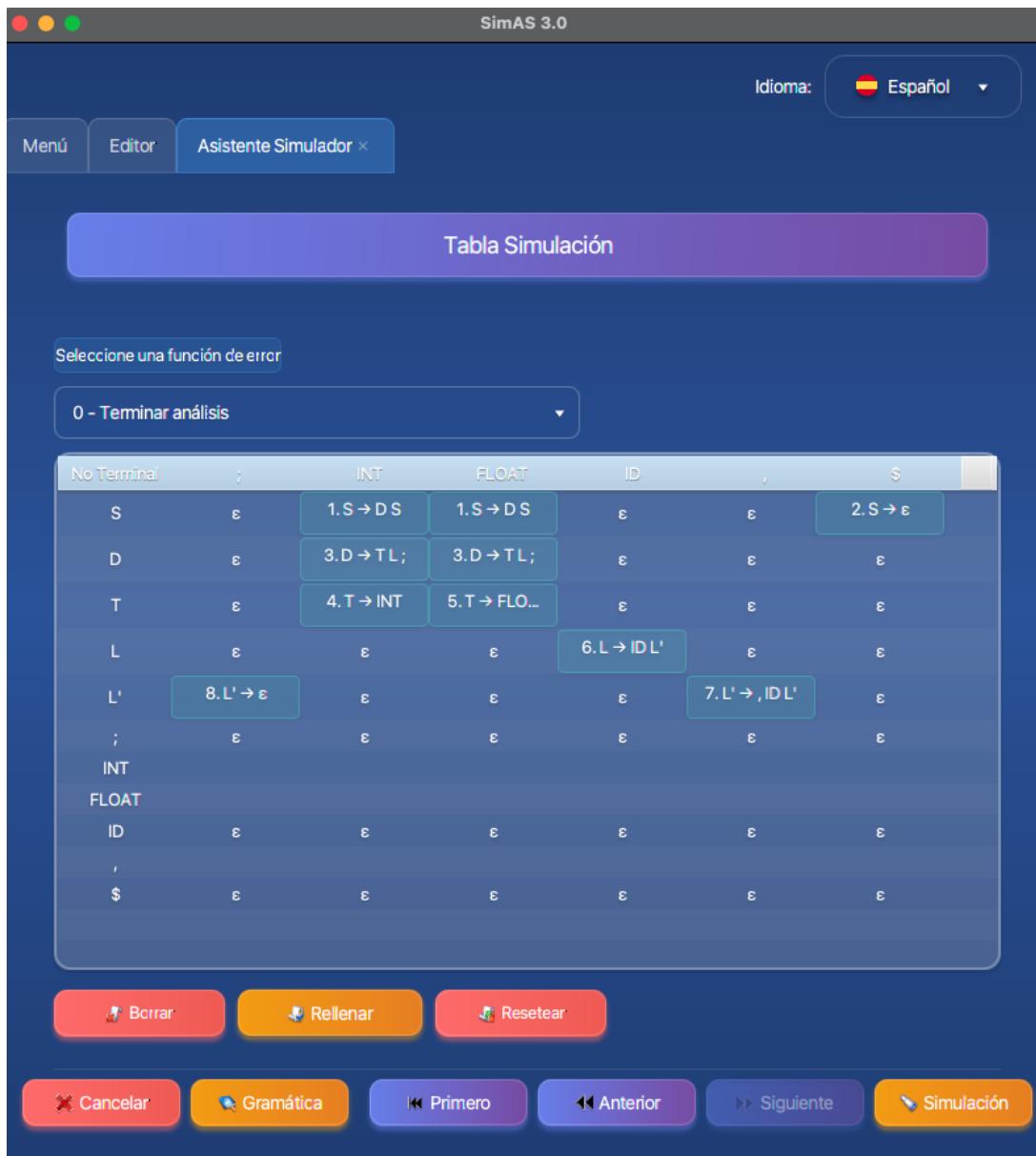


Figura 6.16: Paso 5: Tabla predictiva completa

Una vez completada la configuración de la tabla predictiva, se puede proceder al simulador principal haciendo clic en el botón "Simulación". Desde el simulador principal se accede a la simulación interactiva.

6.3.7. Simulador principal

El simulador principal muestra un resumen completo de la configuración realizada, incluyendo las producciones modificadas, las funciones de error configuradas y la tabla predictiva completa. La figura 6.17 muestra la interfaz del simulador principal:

6.3. SIMULACIÓN DEL ANÁLISIS SINTÁCTICO DESCENDENTE

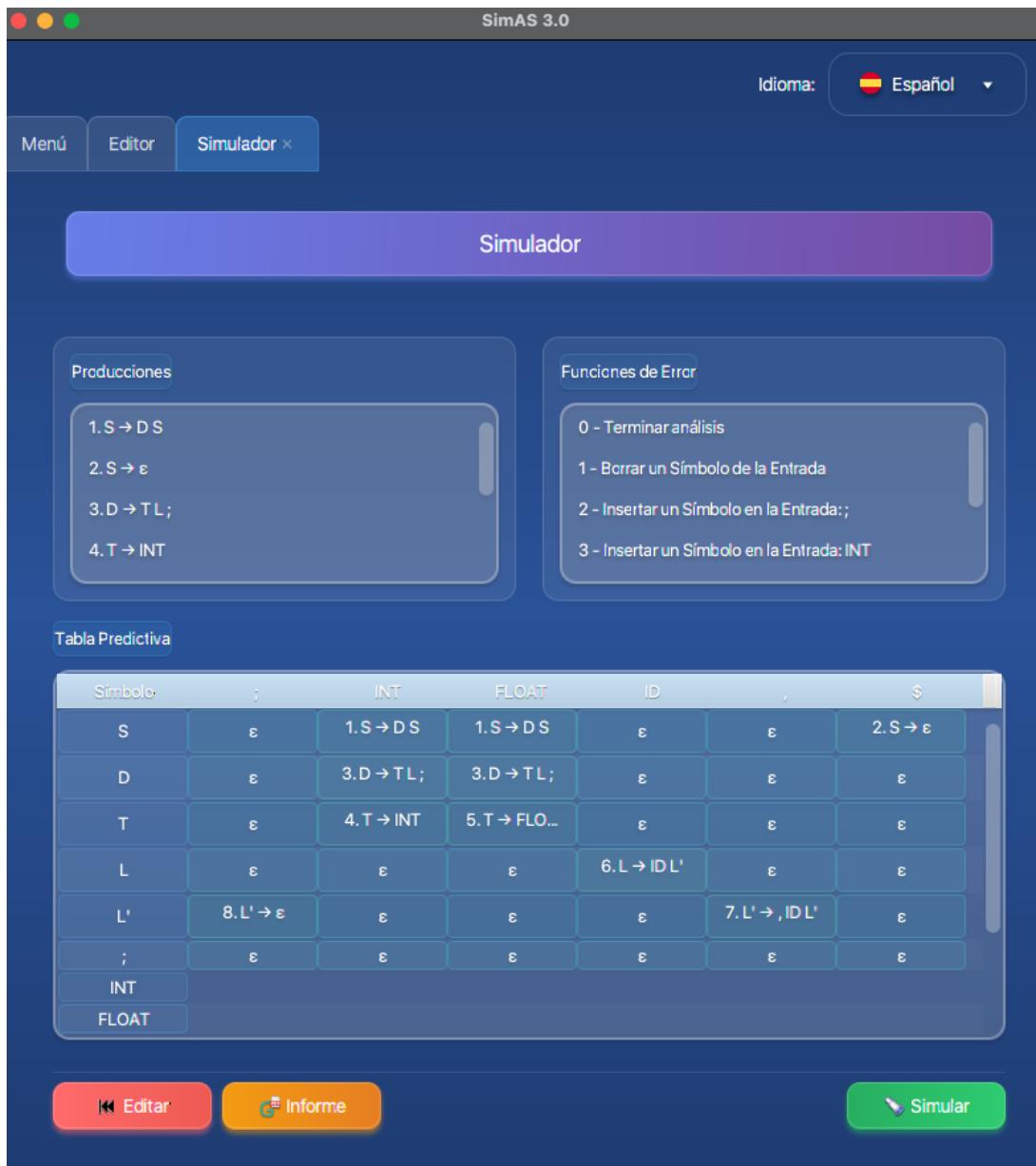


Figura 6.17: Simulador principal con resumen de configuración

Desde este simulador se puede generar un informe PDF con toda la información y acceder a la simulación interactiva.

6.3.8. Simulación interactiva

Desde el simulador principal, haciendo clic en el botón "Simular", se accede a la simulación interactiva. La figura 6.18 muestra la introducción de una cadena de entrada:



Figura 6.18: Introducción de cadena de entrada para simulación

Para este ejemplo, utilizaremos la cadena "INT x, y;" que representa la declaración de dos variables enteras.

6.3.9. Derivación de la simulación

La figura 6.19 muestra la derivación paso a paso del análisis sintáctico:

6.3. SIMULACIÓN DEL ANÁLISIS SINTÁCTICO DESCENDENTE

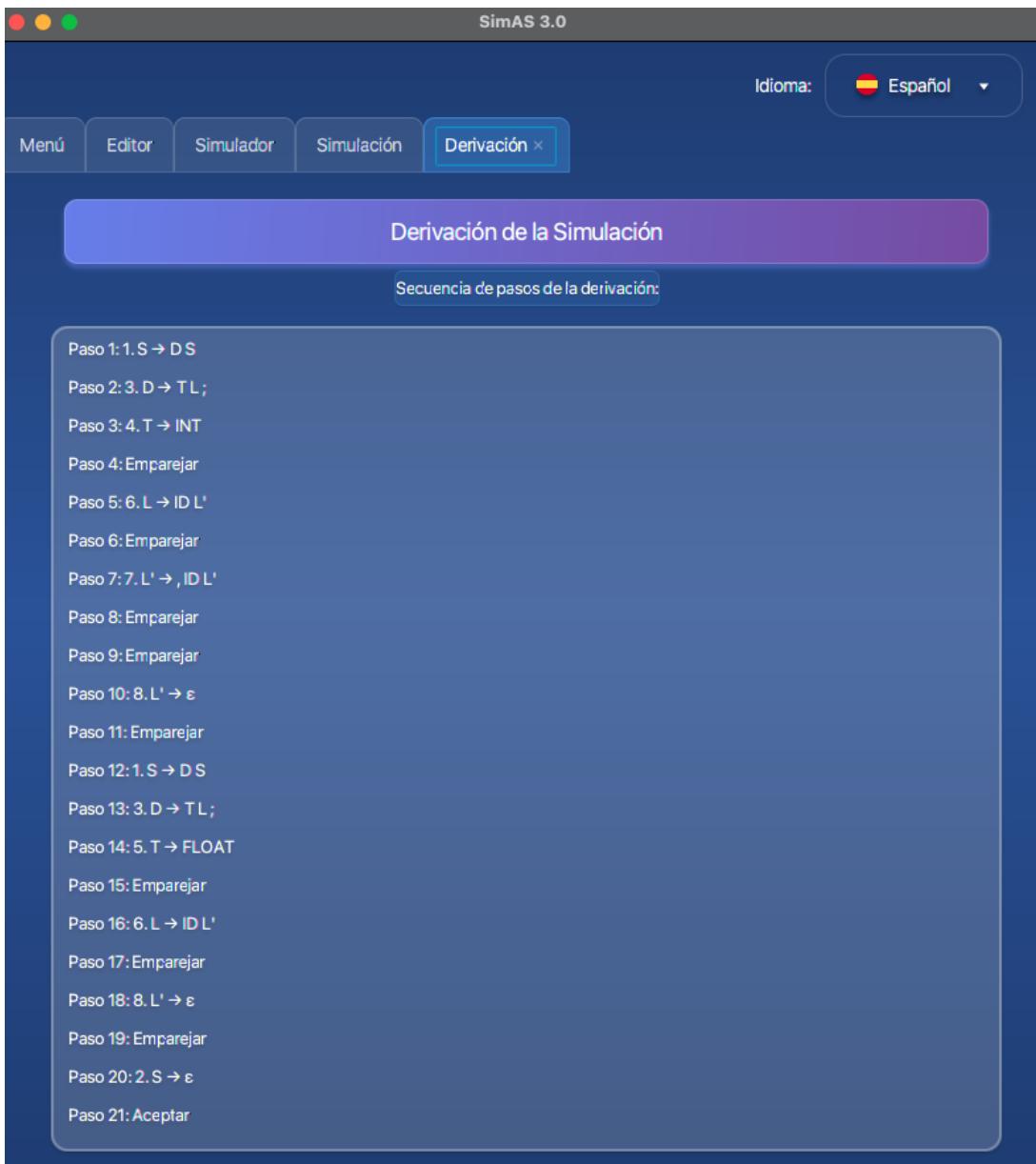


Figura 6.19: Derivación paso a paso del análisis sintáctico

La derivación muestra cómo se aplican las reglas de producción para reconocer la cadena de entrada, comenzando con el símbolo inicial S y aplicando las reglas correspondientes según la tabla predictiva.

6.3.10. Árbol de análisis sintáctico

La figura 6.20 muestra el árbol de análisis sintáctico generado:

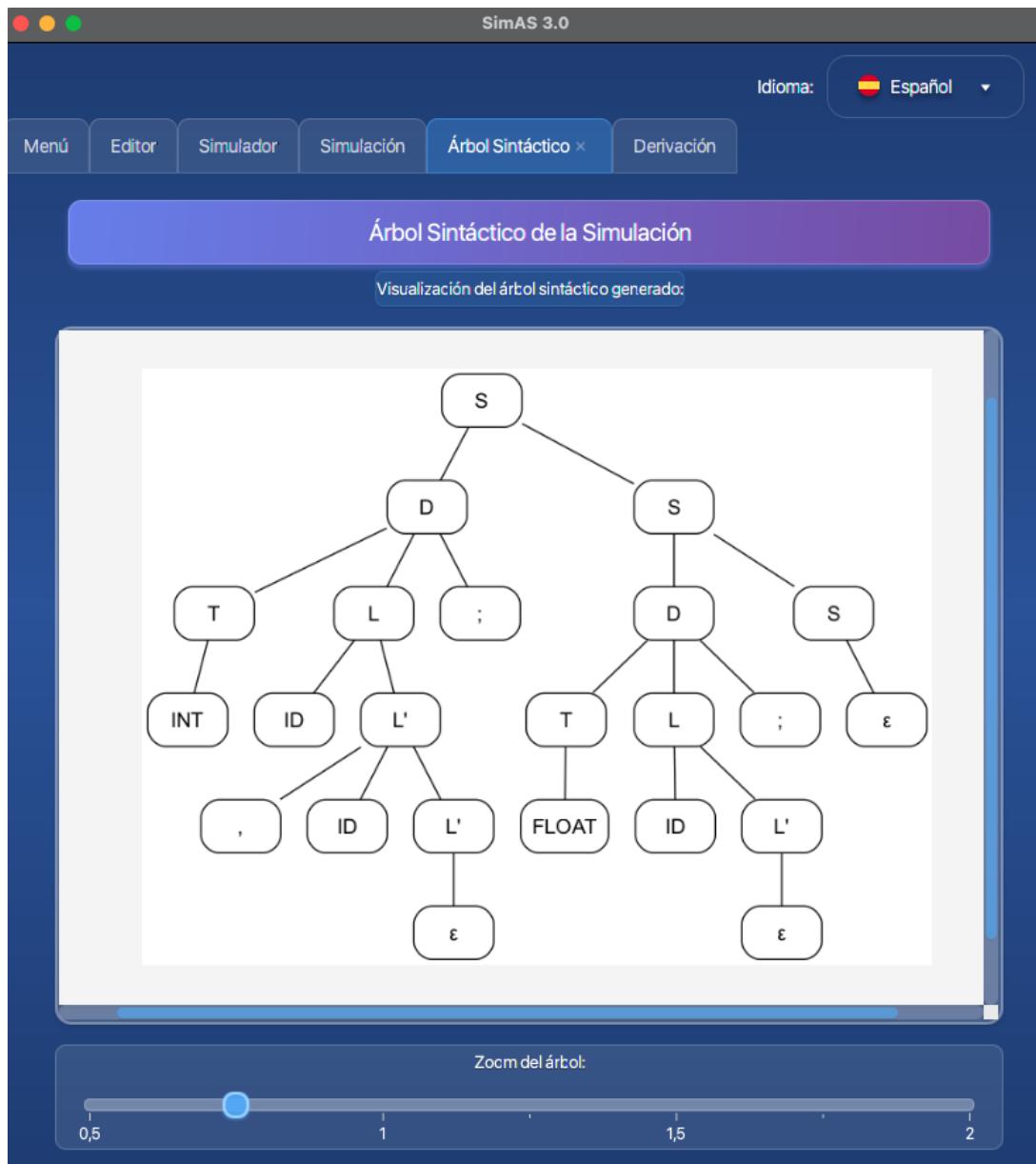


Figura 6.20: Árbol de análisis sintáctico

El árbol muestra la estructura jerárquica de la declaración, donde se puede observar cómo se organizan los elementos: el tipo de dato (INT), seguido de la lista de identificadores (x, y) y el punto y coma final.

6.4. Conclusión del ejemplo

Este ejemplo práctico ha demostrado el flujo completo de trabajo con SimAS 3.0:

1. **Creación de gramática:** Se utilizó el editor guiado para crear una gramática de declaraciones de variables.
2. **Validación:** Se verificó que la gramática estuviera correctamente definida.

3. **Refactorización:** Se confirmó que la gramática ya estaba en forma LL(1) sin recursividad por la izquierda.
4. **Construcción de tablas:** Se generaron automáticamente los conjuntos PRIMERO y SIGUIENTE, y la tabla predictiva.
5. **Configuración de errores:** Se establecieron funciones de error para el manejo de cadenas incorrectas.
6. **Simulación:** Se realizó un análisis sintáctico completo de una declaración de variables.
7. **Visualización:** Se generaron la derivación y el árbol de análisis sintáctico.

Este proceso ilustra cómo SimAS 3.0 facilita el aprendizaje y comprensión de los conceptos fundamentales del análisis sintáctico descendente, proporcionando herramientas visuales e interactivas que hacen accesible este tema complejo de la teoría de lenguajes formales.

Bibliografía

- [1] Aho, A. V., Lam, M. S., Sethi, R. y Ullman, J. D. *Compiladores: Principios, Técnicas y Herramientas. Segunda edición.* México: Pearson Educación, 2008.
- [2] Eclipse Adoptium. *Eclipse Adoptium - OpenJDK (alternativa de código abierto).* [En línea. Última consulta: 15-12-2024]. URL: <https://adoptium.net/>.
- [3] GeeksforGeeks. *First and Follow Sets.* [En línea. Última consulta: 15-12-2024]. URL: <https://www.geeksforgeeks.org/first-set-in-syntax-analysis/>.
- [4] iText Group. *iText.* [En línea. Última consulta: 17-9-2025]. URL: <https://itextpdf.com/>.
- [5] Oracle. *Java.* [En línea. Última consulta: 17-9-2025]. URL: <https://www.java.com/es>.
- [6] Oracle. *JavaFX.* [En línea. Última consulta: 15-12-2024]. URL: <https://openjfx.io/>.
- [7] Wikipedia. *Forma de Backus-Naur.* [En línea. Última consulta: 15-12-2024]. URL: https://es.wikipedia.org/wiki/Forma_de_Backus-Naur.
- [8] Wikipedia. *LL parser.* [En línea. Última consulta: 15-12-2024]. URL: https://en.wikipedia.org/wiki/LL_parser.
- [9] XML Tutorial. *W3 School.* [En línea. Última consulta: 15-12-2024]. URL: <https://www.w3schools.com/xml/>.