



Sistemas Informáticos

Curso 2006-2007

ACIDE: A Configurable IDE

Diego Cardiel Freire

Juan José Ortiz Sánchez

Delfín Rupérez Cañas

Dirigido por:

Prof. Fernando Sáenz Pérez

Dpto. Sistemas Informáticos y Computación

Facultad de Informática
Universidad Complutense de
Madrid

ÍNDICE

Resumen del proyecto	4
Resumen en español	5
Resumen en inglés	5
Estado del arte	6
Estándares	10
Control de versiones	11
Estándar de la documentación	11
Estándares de código	12
Gestión de configuración.....	13
Gestión de riesgos.....	16
Análisis de riesgos	17
Estrategia gestión de riesgos.....	18
Plan de gestión de riesgos.....	20
Requisitos	23
Lista de casos de uso	27
Planificación	29
Iteración 1	30
Iteración 2	31
Iteración 3	31
Iteración 4	32
Iteración 5	32
Iteración 6	33
Carga de trabajo real durante el curso	34
Conclusiones.....	38
Objetivos cumplidos	39
Objetivos extra.....	40
Posibles ampliaciones.....	41
Bibliografía.....	43
Libros.....	44
Webs	44
Apéndice 1: Manual de usuario	47

Requisitos del sistema.....	48
Menú archivo	49
Menú edición	54
Menú ver.....	59
Gestor de proyectos	61
Configuración de menú.....	72
Configuración de la barra de herramientas.....	75
Configuración sintáctica	80
Configuración del idioma	84
Configuración léxica.....	86
Configuración de la consola	90
Menú ayuda	92
Apéndice 2: Diagramas UML.....	93
Apéndice 3: Lista de palabras clave	119
Información de contacto	121
Autorización	122

ACIDE

RESUMEN DEL PROYECTO

Versión: 5.0
Fecha: 05/07/2007

RESUMEN EN ESPAÑOL

El objetivo de este proyecto es el diseño e implementación de un entorno de desarrollo integrado (IDE, Integrated Development Enviroment) para lenguajes de programación.

El sistema debe ser lo suficientemente general como para parametrizarlo para distintos lenguajes. En particular es necesario contar con un editor semi-inteligente para el análisis interactivo (on the fly) de los programas que acepte distintas gramáticas.

Como instancia de IDE se parametrizará el sistema para el lenguaje lógico funcional con restricciones Toy (toy.sourceforge.net) y posiblemente para otros lenguajes, como Datalog (des.sourceforge.net). El sistema se implementará en Java.

RESUMEN EN INGLÉS

This project is aimed to provide a multiplatform configurable integrated development environment which can be configured in order to be used with any development system such as interpreters, compilers and database systems.

Features of this system include: project management, multfile editing, syntax colouring, and parsing on-the-fly (which informs of syntax errors when editing programs prior to the compilation).

End-users who can benefit from this system include: researchers that develop programming systems, developers, and database users. It is implemented in Java in order to be platform independent.

ACIDE

ESTADO DEL ARTE

Versión: 1.3
Fecha: 01/12/2006

ESTADO DEL ARTE

Actualmente es posible encontrar muchos entornos de desarrollo integrado en los que se puede elegir entre varias vistas y configuraciones diferentes para algunos de los lenguajes de programación existentes. Dado que nuestro proyecto versa precisamente sobre este tipo de herramientas hemos estado analizando algunas para poder hacernos una idea de qué es lo que tiene que ofrecer nuestra aplicación, qué interfaz puede ser el más adecuado, qué elementos nuevos se pueden incluir para hacer que este entorno de desarrollo incluya algo nuevo que todo lo que ofrecen los que ya existen...

Por una parte hemos analizado simplemente algunos editores de texto que permiten el resaltado de palabras dependiendo del lenguaje para el que tengamos configurado el editor. A este tipo de aplicaciones le faltan muchas de las funciones que perseguimos incluir en nuestra aplicación, pero no obstante nos sirven para darnos una idea de lo simple y lo conciso que debe ser un editor de código y además nos aporta ideas sobre el resaltado de palabras reservadas de un lenguaje de programación concreto. En particular nos hemos fijado en Crimson Editor (<http://www.crimsoneditor.com>) y JEdit (<http://www.jedit.org>).

Crimson Editor es un editor de texto bastante sencillo que permite seleccionar entre varios listados de palabras reservadas de diferentes lenguajes, así como la modificación de las mismas o la generación de un nuevo listado de palabras en caso de querer incluir un lenguaje nuevo. Por su parte JEdit ofrece las mismas características que Crimson y además permite configurar más cosas como la apariencia de los menús. Cuenta con la ventaja añadida de que es un programa de código abierto y que cualquiera puede modificarlo y mejorarlo, de forma que tiene muchos listados de palabras reservadas de lenguajes distintos y funcionalidades mejoradas.

En cuanto a los entornos de desarrollo integrado observamos un par de tipos: los que van enfocados únicamente a un lenguaje de programación en concreto, y los que tienen distintas configuraciones para varios lenguajes.

La principal ventaja de los primeros es que al dedicarse exclusivamente a uno de los lenguajes permite mucha más especialización y tiene herramientas más específicas, como puede ser un entorno gráfico para el diseño de componentes gráficas del lenguaje. En esta primera clase encontramos aplicaciones como JBuilder (<http://www.borland.com/jbuilder>), JCreator (<http://www.jcreator.com>), C++Builder (<http://www.borland.com/bcppbuilder>)... JBuilder y JCreator se fijan exclusivamente en Java como lenguaje de programación y ofrecen numerosas características para facilitar al desarrollador la codificación de programas. En este sentido JCreator es más parco en funcionalidad que JBuilder, que además de permitir al usuario programar los botones del interfaz a su gusto con las funciones que use más a menudo, tiene un interfaz gráfico para la creación de elementos de Swing y ofrece la opción de depuración que es parte muy importante a la hora de revisar que el programa funcione correctamente y reparar los fallos de ejecución tan difíciles de localizar. JBuilder es, además, de la misma casa que C++ Builder, y seguramente por esta razón son ambos tan parecidos: ofrecen las mismas prestaciones (o casi), sólo que para lenguajes distintos; que en el caso de C++ Builder casi sobra decir que es exclusivo de C++.

En el segundo grupo nos encontramos con aplicaciones como Eclipse (<http://www.eclipse.org>) que actualmente puede ser el máximo exponente de este grupo. En este entorno de desarrollo podemos encontrar herramientas de compilación, depuración y desarrollo para gran variedad de lenguajes: Java, C/C++, PHP, COBOL, lenguajes de programación web (HTML, JSP...), SQL...e incluso modelado con UML. El gran contratiempo de este programa es que todas estas funcionalidades se obtienen a través de una larga descarga de plugins que van añadiendo nuevas propiedades y que hay que estar actualizando de vez en cuando. Además sólo es posible configurarlo para aquellos lenguajes que los desarrolladores nos ofrezcan, debido a que no se pueden insertar o modificar a mano las características de cualquier otro lenguaje que no estén incluidas ya en la aplicación. Además hay que añadirle la complicación de los menús y la configuración de algunas de las propiedades (bien del propio Eclipse, o bien del proyecto en desarrollo). Aún así Eclipse es bastante completo, por todas las opciones que ofrece para la creación y desarrollo de proyectos (tanto situados en ubicaciones locales como remotas), así como las diferentes vistas de editores para cada uno de los lenguajes que oferta, y la interacción con programas externos como pueden ser los distintos compiladores necesarios para los lenguajes que incluye.

En este punto en el que nos encontramos, habiendo analizado las distintas posibilidades que la variedad nos ofrece, es cuando decidimos hasta qué punto queremos nuestra aplicación sea tan flexible como sea posible pero además que no sea complicada, ni enrevesada, ni demasiado específica para un lenguaje concreto. No podemos llegar al extremo de implementar algo demasiado sencillo pero con poca funcionalidad que provoque que al usuario le resulte poco útil el entorno, pero tampoco debemos desarrollar una aplicación que pretenda atender las especificidades de cada uno de los lenguajes de forma que sea desechada por ser de uso complicado. Por lo tanto, la generalidad y la simplicidad en su justa medida serán buenas consejeras para la implementación de nuestro entorno de desarrollo integrado.

ACIDE

ESTANDARES

Versión: 1.2
Fecha: 29/11/2006

ESTÁNDARES

Para el correcto desarrollo y tratamiento simultáneos de los distintos archivos (tanto de código como de documentación) que se manejarán a lo largo del trabajo se establecen los siguientes estándares:

Control de versiones

Para la numeración de las versiones de los diferentes documentos y archivos se deberá usar la siguiente nomenclatura: <Nombre_documento>_v<nº_iteración>.<nº_version>. La numeración de las versiones comienza en 0.

Estándar de documentación

El nombre que identificará a los documentos debe ser a la vez claramente identificativo y descriptivo de la información que contiene. Todos los documentos incluirán una portada como la mostrada en la página 3, en la que se muestra el nombre del documento, nombre del proyecto, versión del documento y fecha de última modificación (en formato dd/mm/aaaa). A continuación mostrará el índice con todos los contenidos del documento.

En cuanto a la presentación del texto en el documento será la siguiente:

- Fuente para todo el documento y las cabeceras: Times New Roman.
- Color de letra para todo el documento y las cabeceras: negra.
- Formato texto normal: tamaño 12, justificado.
- Formato títulos principales: tamaño 16, negrita, subrayado, centrado.
- Formato títulos secundarios: tamaños 14 y 12, negrita, subrayado, justificado.
- Dos líneas en blanco antes de cada título principal.
- Una línea en blanco antes de cada título secundario y después de cada título (principal y secundario) y punto y aparte.

- Tabulación inicial en la primera línea de cada párrafo.
- Interlineado de 1,5 líneas.
- Numeración de página en la parte inferior central.
- Encabezado con el texto “UCM. Sistemas Informáticos 2006/2007” en la parte superior izquierda y el nombre del documento en la parte superior derecha, ambos con letra Times New Roman, color negro, tamaño 8, cursiva.
- Para las listas de enumeraciones se usará la herramienta del editor de texto correspondiente (Microsoft Word)

Estándares de código

Para los archivos de código no será necesario incluir en el nombre la versión tal y como está explicado en el apartado de control de versiones debido a que se utilizará como herramienta de trabajo un CVS que implementa un sistema de control automático de versiones. Por contra la versión y fecha de última modificación (en formato dd/mm/aaaa) se deberá incluir al principio de cada clase como comentario de la misma.

Cada clase debe tener un nombre claramente explicativo y representativo, así como todas las variables y métodos que se usen para el desarrollo del código. El nombre de las clases deberá comenzar siempre con mayúscula, y el de las variables y métodos con minúscula. Aún así se deberá incluir antes de cada uno de estos tres elementos nombrados una breve explicación del propósito y función que desarrollan dentro del programa. Esta explicación será siempre en forma de comentarios de código colocados una línea antes que la declaración del elemento que describen.

Siguiendo estas indicaciones lo que debemos incluir antes de cada clase será:

```
/*
```

```
* Nombre de la clase: nombre
```

```
* Versión: v1.1
```

```
* Fecha última modificación: dd/mm/aaaa
```

```
* Descripción: descripción de la clase
```

```
*/
```

ACIDE

GESTION DE CONFIGURACION

Versión: 1.2
Fecha: 29/11/2006

GESTIÓN DE CONFIGURACIÓN

Todos los archivos, tanto de documentación como de código, son objeto de control de la gestión de configuración.

Para la nomenclatura de los documentos y clases se elegirán nombres claramente identificativos y descriptivos de la información que contienen. En cuanto a los archivos de documentación se incluirá la versión en el mismo nombre del archivo; y para los archivos de código no se hará en el nombre sino en el comentario previo a la declaración de la clase. La razón de que no se haga para los archivos de código es que se usará un CVS como herramienta de desarrollo que lleva incorporado un sistema de control automático de versiones.

Para controlar los cambios y no perder las versiones anteriores se guardarán siempre las versiones de todos los archivos de documentación hasta el final de la iteración en curso hasta establecer un documento final que será la versión definitiva de ese documento para esa iteración. Con los archivos de código no será necesario llevar este control debido al uso del CVS y su control automático de versiones.

Dado el reducido número de integrantes del grupo y la facilidad de coordinación en que se traduce, sumado a que proporciona una gran facilidad de revisión de cualquier información en todo momento, todos los miembros deberán tener una copia de los archivos. Así en el momento en el que un documento sea modificado se enviará por correo a los otros dos integrantes. De este modo todos tendrán una versión actualizada de los documentos en todo momento. Para los archivos de código (que estarán almacenados en el CVS y disco duro de cada uno de los componentes del grupo) se harán copias de seguridad todos los domingos (para evitar perder trabajo en caso de error del CVS) por parte de todos los miembros del grupo y se guardarán como “copia_aaaammdd.rar” donde aaaammdd se refiere al año (aaaa), mes (mm) y día (dd) escrito en formato numérico.

Para realizar cambios sobre un archivo que tengan que modificar varias personas antes de alcanzar su aspecto final se deberá establecer un turno de modificación para no

provocar pérdidas e inconsistencias de la información. De esta manera será necesario avisar a las personas correspondientes antes de modificar el archivo, y mientras no se reciba un nuevo aviso de que el archivo vuelve a estar disponible (adjuntando la nueva versión del archivo) no podrá ser modificado por nadie más.

Las herramientas con las que se desarrollará el proyecto serán las siguientes:

- Eclipse como entorno de desarrollo.
- Microsoft Word para elaboración de la documentación.
- Rational Rose para el diseño de los diagramas UML.
- Microsoft Project para crear los diagramas de Gantt.
- CVS de www.berlios.de para el almacenamiento, sincronización y control de versiones del software

ACIDE

GESTION DE RIESGOS

Versión: 1.2
Fecha: 29/11/2006

GESTIÓN DE RIESGOS

La gestión de Riesgos se va a dividir en tres apartados:

- Análisis de Riesgos
- Estrategia Gestión de Riesgos
- Plan de Gestión de Riesgos

Análisis de riesgos

En este apartado se van a analizar los riesgos más relevantes que se pueden dar en el proyecto:

- **Los requisitos poco estables:** Como es entorno de desarrollo parametrizado para varios lenguajes puede haber dificultades en su integración debido a que no se cumplen algunos de los requisitos base de la aplicación.
- **Deficiencias en el personal:** Puede a que la aplicación se hace en equipo y de forma concurrente, la falta de uno de sus miembros, implica problemas en el desarrollo y retrasos en el proyecto.
- **Problemas en el desarrollo de las pruebas:** Cuando se testee (las pruebas) la aplicación, puede encontrarse deficiencias que retrasen el proyecto.
- **Incapacidad de completar el proyecto:** Puede darse que no haya tiempo de realizar todas las funciones del proyecto debido a su complejidad.
- **Realizar módulos innecesarios en la aplicación:** Esto es lo que se entiende como “chapado”. Puede pasar que se codifiquen módulos que no se necesiten en la aplicación, lo que implica pérdida de tiempo y falta de integración.
- **Desarrollo erróneo del interfaz de usuario:** Puede pasar se que realice un interfaz de usuario poco práctico, aunque es poco probable por la especificación de los requisitos.
- **Planificación:** Errores que pueden llevar a no completar el proyecto.
- **Problemas en la motivación:** Al surgir problemas desmotiven a los miembros del equipo en su trabajo.

- **Comunicación y cooperación:** Una mala comunicación y/o cooperación en el equipo dificulta el desarrollo de la planificación.
- **Actitud:** Al ser código libre, es positivo la corrección y mejora del código. Por eso no es bueno: “si funciona no lo toques”, “ya vale” conformismo que lleva a software de baja calidad.
- **Experiencia:** Al ser un proyecto nuevo y de esta envergadura se requiere una mayor precisión en las decisiones y en el trabajo.
- **Herramientas:** Aunque se conocen las herramientas (Eclipse, Rational Rose, Ms Project) suelen surgir tareas que al tener mayor dificultad entorpecen el desarrollo, por ejemplo: de ensamblaje, módulo de depuración, etc.
- **Diseño:** Un mal diseño conlleva una mala codificación y por tanto mala ejecución. Puede ocurrir que el diseño no se corresponda perfectamente con la implementación que se lleve a cabo, provocando dificultades en el momento de la verificación.
- **Pérdida de Información:** Retrasa el trabajo. Para evitarlo se utiliza un sistema de versiones concurrente (CVS) donde se podrá hacer copias de seguridad.

Estrategia gestión de riesgos

Se va a utilizar una estrategia preactiva. Ésta tiene las siguientes características:

- Comienza antes que los trabajos técnicos.
- Se identifican riesgos potenciales.
- Se evalúa probabilidad y consecuencia de los riesgos.
- Se priorizan los riesgos.
- Se produce un plan de gestión del riesgo.

El objetivo es evitar el riesgo pero también se proporcionan planes de contingencia (última solución)

A continuación se priorizan los riesgos, según su importancia, en orden descendente. La prioridad se encuentra entre paréntesis.

1) Pérdida de la información (Alta):

Si perdemos la información es como si no hubiésemos hecho nada, se considera un riesgo intolerable.

- Consecuencia: Catastrófico.
- Probabilidad: Remota.

2) Fallos en diseño (Medio):

- Consecuencia: Serio.
- Probabilidad: Ocasional

3) Requisitos poco estables (Medio):

Los requisitos pueden variar debido a las exigencias y problemas del proyecto. Aunque se hayan establecido unos requisitos en la planificación, estos pueden variar debido a la dificultad del proyecto.

- Consecuencia: De menor importancia.
- Probabilidad: Probable.

4) Baja del personal (Tolerable)

- Consecuencia: De menor importancia.
- Probabilidad: Remota.

5) Incapacidad de realizar el proyecto completa (Bajo):

Se intentará realizar el proyecto completo, si no fuese posible se realizarán sólo los requisitos obligatorios.

- Consecuencia: Seria.
- Probabilidad: Remota.

Plan de gestión de riesgos

Este plan servirá para la reducción, supervisión y gestión de cada riesgo que se pueda producir. Se intentará, llevar una gestión de riesgos lo más coherente posible y eficiente.

- **Tabla de riesgo del proyecto**

Se refleja los niveles de riesgo según Consecuencia y Probabilidad:

Consecuencia \ Probabilidad	Probable	Ocasional	Remota
Catastrófico	INTOLERABLE	INTOLERABLE	ALTO
Serio	ALTO	MEDIO	BAJO
De menor importancia	MEDIO	BAJO	TOLERABLE
Insignificante	BAJO	TOLERABLE	TOLERABLE

Consecuencia:

- Catastrófico: Más 10% Coste y más 10 % reducción de la funcionalidad en el producto.
- Serio: Menos 5% Coste y menos 5% reducción de la funcionalidad.
- De menor importancia: Menos 2% coste y menos 2% reducción de la funcionalidad en el producto.

- **Reducción, supervisión y gestión del riesgo**

Pérdida de la información:

- Reducción:** Se intentará salvar los datos con frecuencia. Además se intentará que la información se encuentre, por lo menos, en dos soportes distintos y como mínimo dos miembros del equipo dispongan de todas la partes del proyecto.
- Supervisión:** No hay una tarea concreta que gestione este riesgo. El equipo dispondrá de un sistema de versiones concurrente (CVS) donde se almacene la información que se genere.

- c) **Gestión:** Si se han perdido los datos se intentará recuperarlos con alguna herramienta apropiada. En caso de no poder recuperarlos, habrá que aumentar el esfuerzo para volver a realizar las partes del proyecto que se perdieron.

Fallos diseño:

- a) **Reducción:** Se intentarán realizar los diseños con la máxima eficacia posible. Además, tendrán que ser coherentes y de fácil implementación.
- b) **Supervisión:** Habrá una tarea específica de revisión de los diseños, en la cual se podrá verificar su validez, y en caso de contener errores, corregirlos.
- c) **Gestión:** En caso de producirse fallos importantes en el diseño que impidan la implementación, se volverán a repetir, lo que conlleva un aumento del esfuerzo por parte del equipo.

Requisitos poco estables:

- a) **Reducción:** Se intentará seguir todos los requisitos de forma ordenada y coherente.
- b) **Supervisión:** Se seguirá la planificación para completar todos los requisitos establecidos.
- c) **Gestión:** En caso de que cumpla este riesgo se intentará finalizar los requisitos que sean estables según la planificación adaptándolos a las variaciones que se encuentren en el transcurso del proyecto.

Baja del personal:

- a) **Reducción:** Para este riesgo no hay reducción.
- b) **Supervisión:** No hay una tarea concreta para este riesgo.
- c) **Gestión:** En caso de riesgo, se tendrá que aumentar considerablemente el esfuerzo personal para suplir al miembro que falte, e incluso redistribuir la organización en caso necesario.

Incapacidad de realizar la práctica completa:

- a) **Reducción:** Se intentará realizar un seguimiento de la planificación. Si es necesario se aumentará el esfuerzo.

- b) **Supervisión:** Para la supervisión se comprobarán las tareas en la planificación temporal.
- c) **Gestión:** Si el riesgo se convierte en una realidad, no se desarrollarán módulos de prioridad menor, por ejemplo: el módulo depurador.

ACIDE

REQUISITOS

Versión: 1.2
Fecha: 29/11/2006

REQUISITOS

Intentar definir las características de un sistema tan general es muy difícil y no están muy claras las barreras que se deben tener en cuenta para no caer en la especificidad de un lenguaje de programación en concreto.

El listado de los requisitos de este IDE está pensado para que pueda ser parametrizable para cualquier lenguaje de programación y sea así lo más general posible, en el sentido de que cada usuario pueda configurar todas (o la mayor parte) de las características de cada lenguaje y el entorno visual que tendrá disponible en cada caso.

Basándonos en estas premisas hemos identificado los siguientes requisitos para nuestro sistema:

- Editor de texto: es una parte fundamental y necesaria de todo IDE para poder visualizar y modificar los archivos de código.
- Menús: los menús serán lo más generalizados posible y luego tendrán partes configurables a partir de un archivo de texto que el propio usuario determinará. Pueden cambiar dependiendo del lenguaje con el que esté trabajando el IDE en cada momento.
- Iconos: accesos rápidos a las funciones más usadas y generales (comunes a todo tipo de lenguaje de programación).
- Interfaz Gráfica de Usuario (GUI) para la parametrización de todos los aspectos de los lenguajes y algunos aspectos de la parte visual del IDE.
- Explorador del proyecto: cuadro de texto a la izquierda del editor que permite visualizar elementos del proyecto.
- Números de línea: siempre útiles.
- Barra de estado: mostrará entre otras cosas la última entrada al log del programa e información útil al programador.
- Log: registro de todas las operaciones que se realizan en el transcurso de la utilización del programa. Inicialmente no estará visible a no ser que el propio

usuario indique lo contrario; y en tal caso se le mostrará esta información en una ventana aparte.

- Salida (o consola): cuadro de texto situado debajo del cuadro de edición en el que se muestra la salida de los programas desarrollados por el usuario, mensajes de error, avisos...
- Archivos de configuración y parametrización: el sistema dispone de una serie de archivos de texto en los que se guarda la configuración del IDE para cada uno de los lenguajes de programación parametrizados por el usuario, así como los parámetros de cada uno de ellos. Dentro de éstos últimos se incluyen los que contienen las diferentes palabras reservadas, las gramáticas, extensiones de archivo...
- Reconocimiento de las palabras reservadas de cada uno de los lenguajes: se marcan con distintos colores cada una de las palabras reservadas del lenguaje de programación activo, que previamente el usuario debe haber insertado en un archivo de configuración.
- Reconocimiento de los errores de sintaxis: señala las instrucciones que no se correspondan con la gramática del lenguaje y que tengan algún error de sintaxis con ayuda de una gramática proporcionada, configurada y salvada en un archivo previamente por el usuario.
- Creación de proyectos: para poder crear proyectos de cada uno de los lenguajes usados.
- Creación archivos relativos al lenguaje y proyecto que se está desarrollando con ayuda de las extensiones parametrizadas para el lenguaje activo.
- Guardar y cargar la configuración del IDE establecida por el usuario para un lenguaje de programación en concreto.
- Plantillas de código: el usuario podrá definir distintas plantillas de texto que podrá insertar en su código. Será especialmente útil para partes de código que se tengan que escribir muchas veces.
- Carga de la configuración previamente definida por el usuario de un lenguaje en particular para cuando abramos un archivo o proyecto que pertenezca a ese lenguaje.
- Intérprete y compilador: al tratarse de un sistema muy general deben existir ambos para cualquier posible lenguaje que se parametrize en el IDE.

- Ejecución: posibilidad de ejecución para un programa desde el IDE.
- Ayuda: para cualquier problema que el usuario pueda encontrar con el uso del IDE.
- Depurador: herramienta para poder encontrar y resolver errores del código en ejecución.

ACIDE

LISTA DE CASOS DE USO

Versión: 1.0
Fecha: 14/11/2006

LISTA DE CASOS DE USO

- crear / guardar / abrir un archivo.
- operaciones sobre el texto: copiar / pegar / deshacer / rehacer / buscar / seleccionar / reemplazar.
- uso de iconos como enlaces rápidos.
- activar / desactivar la numeración de líneas.
- uso de asistente para crear plantillas de código / clases / proyectos.
- navegación mediante el explorador del proyecto.
- consulta de la ayuda del IDE.
- configuración de los menús desplegados mediante archivo de texto.
- cambio del lenguaje activo.
- configuración de las palabras reservadas de un lenguaje.
- configuración de la gramática de un nuevo lenguaje.
- activar / desactivar la interpretación / compilación en tiempo real del código.
- activar / desactivar el resaltado de sintaxis del código.
- ejecución del código.
- ejecución con parámetros del código.
- depuración del código.
- consulta del log generado por el IDE.
- consulta de la salida de la ejecución del programa.

ACIDE

PLANIFICACION

Versión: 1.2
Fecha: 29/11/2006

PLANIFICACIÓN

En cuanto al desarrollo del trabajo lo dividiremos por iteraciones de duración variable dependiendo de los objetivos asignados a cada una de ellas. En general todas durarán entre 3 y 4 semanas e irán marcadas por una fecha de comienzo y finalización, para la cual deben estar terminados y conseguidos los objetivos fijados.

Hemos contado con periodos en los que no podemos trabajar tan a fondo en el proyecto debido a exámenes, periodos festivos...Así partimos del lunes 13 de noviembre de 2006 hasta el jueves 31 de mayo de 2007, y contamos con los periodos festivos como media semana de trabajo por una semana completa de 7 días. En cuanto a las fechas de exámenes hemos decidido excluirlas de la planificación por la falta de tiempo.

Nuestro plan de fase queda dividido en las siguientes iteraciones:

- **Iteración 1:**

Desde el lunes 13 de noviembre hasta el domingo 3 de diciembre de 2006 (3 semanas).

Básicamente esta iteración se centra en los aspectos gráficos del IDE y en la implementación de las opciones sencillas y comunes del menú. En cuanto a la implementación de la parametrización queda pendiente para la segunda iteración.

Esta iteración incluye:

1. Editor de texto.
2. Menús y operaciones que ofrecen.
3. Iconos de las opciones más comunes y usadas.
4. GUI de la parametrización.
5. Explorador del proyecto.
6. Números de línea.
7. Log.
8. Salida.

- **Iteración 2:**

Desde el lunes 4 de diciembre hasta el domingo 31 de diciembre de 2006 (4 semanas).

En esta iteración se implementa la GUI de parametrización que quedó especificada en la anterior iteración.

Esta iteración incluye:

1. Archivos de configuración y parametrización del proyecto: crearlos, cargarlos y salvarlos.
2. Almacenamiento en un archivo de las palabras reservadas de un lenguaje de programación.
3. Almacenamiento en un archivo de una gramática correspondiente a un lenguaje de programación.
4. Almacenamiento de un listado de extensiones de archivo de un lenguaje de programación para identificarlos como pertenecientes a un lenguaje determinado.
5. Creación de proyectos de cada uno de los lenguajes parametrizados con ayuda de la información previamente introducida por el usuario.
6. Creación de los tipos de archivo de entre los especificados previamente por el usuario como pertenecientes al lenguaje de programación.
7. Cargar y salvar configuraciones definidas por el usuario en un archivo de texto.
8. Parametrización de localización del compilador y las librerías (en caso de ser necesario) a utilizar por el lenguaje de programación.
9. Carga automática de la configuración de un lenguaje al cargar un archivo.
10. Creación y guardado de plantillas de código escrito por el usuario para poder reutilizar código de forma rápida.

- **Iteración 3:**

Desde el lunes 1 de enero hasta el domingo 28 de enero de 2007 (4 semanas), y desde el lunes 26 de febrero hasta el domingo 11 de marzo de 2007 (2 semanas).

A esta iteración la hemos concedido más tiempo debido a que los objetivos fijados son algo más complejos que los de iteraciones anteriores, sumado a que las fechas de exámenes están a mitad de la iteración y con mucha probabilidad disminuya el ritmo de trabajo.

En esta iteración se implementará el reconocimiento en tiempo real de palabras y gramáticas del texto que se encuentre en el cuadro de edición de texto en ese momento. Esto incluye las palabras reservadas introducidas en el archivo de configuración correspondiente al lenguaje activo, así como el reconocimiento de errores de sintaxis del lenguaje con la ayuda de la gramática especificada por el usuario y su parser asociado.

NOTA: antes de su implementación en tiempo real usaremos un botón para activar el mecanismo de reconocimiento.

- **Iteración 4:**

Desde el lunes 12 de marzo hasta el domingo 1 de abril de 2007 (3 semanas). En esta iteración se incluirán en el sistema un mecanismo para la interpretación y para la compilación y que se usarán selectivamente dependiendo del lenguaje de programación activo. También se añade la posibilidad de ejecución del código.

Esta iteración incluye:

1. Intérprete.
2. Compilador.
3. Ejecución.

- **Iteración 5:**

Desde el lunes 2 de abril hasta el domingo 22 de abril de 2007 (3 semanas).

En esta iteración hemos contado con las fechas de Semana Santa y por eso le dedicamos una semana más de lo que inicialmente debería ocuparnos.

Esta iteración estará dedicada a las pruebas del código desarrollado hasta el momento, que aunque al final de cada iteración será probado contamos con que quedarán bugs pendientes o no localizados desde un momento inicial y que pueden ir apareciendo a lo largo del desarrollo. Para ello también

realizaremos pruebas más exhaustivas. Además se observa la posibilidad de realizar mejoras en la aplicación.

- **Iteración 6:**

Desde el lunes 23 de abril hasta el domingo 31 de mayo de 2007 (5 semanas y 4 días).


















Esta iteración es la más larga porque consideramos que el depurador es la parte posiblemente más difícil de implementar de la aplicación y además nos concede un colchón de tiempo por si en alguna de las anteriores iteraciones hemos tenido la necesidad de alargar su duración a causa de algún contratiempo surgido o riesgo no detectado.

OBSERVACION: En ninguna de las iteraciones se ha incluido un apartado de realización de la memoria o ayuda del IDE debido a que se realizará a lo largo de todo el desarrollo de la aplicación. Dentro de cada plan de trabajo se considera que se debe tanto implementar como documentar.

A continuación mostraremos capturas de pantalla de la herramienta MS Project donde podemos ver cuál ha sido la carga de trabajo real a lo largo del desarrollo.

Id		Nombre de tarea	Comienzo	Fin	Nombres de los recursos
1		Iteración 1	lun 13/11/06	mar 28/11/06	
2		Análisis GUI	lun 13/11/06	lun 13/11/06	
3		Editor de Texto y Numeros de linea	lun 13/11/06	lun 13/11/06	Diego
4		Menús e iconos	lun 13/11/06	lun 13/11/06	Delfin
5		GUI parametrización	lun 13/11/06	lun 13/11/06	Juanjo
6		Explorador del proyecto	lun 13/11/06	lun 13/11/06	Diego
7		Log	lun 13/11/06	lun 13/11/06	Juanjo
8		Salida	lun 13/11/06	lun 13/11/06	Diego
9		Diseño GUI	mié 15/11/06	mié 15/11/06	
10		Editor de Texto y Numeros de linea	mié 15/11/06	mié 15/11/06	Diego
11		Menús e iconos	mié 15/11/06	mié 15/11/06	Delfin
12		GUI parametrización	mié 15/11/06	mié 15/11/06	Juanjo
13		Explorador del proyecto	mié 15/11/06	mié 15/11/06	Diego
14		Log	mié 15/11/06	mié 15/11/06	Juanjo
15		Salida	mié 15/11/06	mié 15/11/06	Diego
16		Implementación GUI	lun 20/11/06	lun 20/11/06	
17		Editor de Texto y Numeros de linea	lun 20/11/06	lun 20/11/06	Diego
18		Menús e iconos	lun 20/11/06	lun 20/11/06	Delfin
19		GUI parametrización	lun 20/11/06	lun 20/11/06	Juanjo
20		Explorador del proyecto	lun 20/11/06	lun 20/11/06	Diego
21		Log	lun 20/11/06	lun 20/11/06	Juanjo
22		Salida	lun 20/11/06	lun 20/11/06	Diego
23		Pruebas GUI	lun 27/11/06	mar 28/11/06	
24		Editor de Texto y Numeros de linea	lun 27/11/06	lun 27/11/06	Diego
25		Menús e iconos	lun 27/11/06	mar 28/11/06	Delfin
26		GUI parametrización	lun 27/11/06	mar 28/11/06	Juanjo
27		Explorador del proyecto	lun 27/11/06	mar 28/11/06	Diego
28		Log	lun 27/11/06	mar 28/11/06	Juanjo
29		Salida	lun 27/11/06	mar 28/11/06	Diego
30		Iteración 2	lun 04/12/06	vie 29/12/06	
31		Análisis	lun 04/12/06	jue 07/12/06	
32		Lenguajes GUI	lun 04/12/06	jue 07/12/06	Juanjo
33		L Palabras reservadas + coloreado	lun 04/12/06	jue 07/12/06	Diego
34		L Gramáticas BNF (Guardar reglas)	lun 04/12/06	jue 07/12/06	Juanjo
35		L Cargar tipos de archivo asociados para explor	lun 04/12/06	jue 07/12/06	Diego
36		L Activar la opción de menu correspondiente (cc	lun 04/12/06	jue 07/12/06	Juanjo
37		Proyecto GUI	lun 04/12/06	jue 07/12/06	Juanjo
38		P nombre	lun 04/12/06	jue 07/12/06	Delfin
39		P extension (*.pide)	lun 04/12/06	jue 07/12/06	Delfin
40		P Path del compilador o interprete y librerias	lun 04/12/06	jue 07/12/06	Delfin
41		P Configuración	lun 04/12/06	jue 07/12/06	Delfin
42		Plantillas de código	lun 04/12/06	jue 07/12/06	Diego
43		Lenguajes Nuevo, Guardar, Abrir	lun 04/12/06	jue 07/12/06	Diego
44		Proyecto Nuevo, Guardar, Abrir y Cargar Autom	lun 04/12/06	jue 07/12/06	Delfin
45		Plantillas Nuevo, Guardar, Abrir	lun 04/12/06	jue 07/12/06	Diego
46		Multidioma	lun 04/12/06	jue 07/12/06	Diego
47		Buscar y Reemplazar	lun 04/12/06	jue 07/12/06	Delfin
48		Deshacer y Rehacer	lun 04/12/06	jue 07/12/06	Delfin
49		Configuración menus Nuevo, Guardar, Abrir y C	lun 04/12/06	jue 07/12/06	Juanjo
50		Diseño	vie 08/12/06	dom 10/12/06	
51		Lenguajes GUI	vie 08/12/06	dom 10/12/06	Juanjo
52		L Palabras reservadas + coloreado	vie 08/12/06	dom 10/12/06	Diego

Id		Nombre de tarea	Comienzo	Fin	Nombres de los recursos
53		L Gramáticas BNF (Guardar reglas)	vie 08/12/06	dom 10/12/06	Juanjo
54		L Cargar tipos de archivo asociados para explor	vie 08/12/06	dom 10/12/06	Diego
55		L Activar la opción de menu correspondiente (cc	vie 08/12/06	dom 10/12/06	Juanjo
56		Proyecto GUI	vie 08/12/06	dom 10/12/06	Juanjo
57		P nombre	vie 08/12/06	dom 10/12/06	Delfin
58		P extension (*.pide)	vie 08/12/06	dom 10/12/06	Delfin
59		P Path del compilador o interprete y librerias	vie 08/12/06	dom 10/12/06	Delfin
60		P Configuración	vie 08/12/06	dom 10/12/06	Delfin
61		Plantillas de código	vie 08/12/06	dom 10/12/06	Diego
62		Lenguajes Nuevo, Guardar, Abrir	vie 08/12/06	dom 10/12/06	Diego
63		Proyecto Nuevo, Guardar, Abrir y Cargar Auton	vie 08/12/06	dom 10/12/06	Delfin
64		Plantillas Nuevo, Guardar, Abrir	vie 08/12/06	dom 10/12/06	Diego
65		Multidioma	vie 08/12/06	dom 10/12/06	Diego
66		Buscar y Reemplazar	vie 08/12/06	dom 10/12/06	Delfin
67		Deshacer y Rehacer	vie 08/12/06	dom 10/12/06	Delfin
68		Configuración menus Nuevo, Guardar, Abrir y C	vie 08/12/06	dom 10/12/06	Juanjo
69		Implementacion	lun 11/12/06	dom 24/12/06	
70		Lenguajes GUI	lun 11/12/06	dom 24/12/06	Juanjo
71		L Palabras reservadas + coloreado	lun 11/12/06	dom 24/12/06	Diego
72		L Gramáticas BNF (Guardar reglas)	lun 11/12/06	dom 24/12/06	Juanjo
73		L Cargar tipos de archivo asociados para explor	lun 11/12/06	dom 24/12/06	Diego
74		L Activar la opción de menu correspondiente (cc	lun 11/12/06	dom 24/12/06	Juanjo
75		Proyecto GUI	lun 11/12/06	dom 24/12/06	Juanjo
76		P nombre	lun 11/12/06	dom 24/12/06	Delfin
77		P extension (*.pide)	lun 11/12/06	dom 24/12/06	Delfin
78		P Path del compilador o interprete y librerias	lun 11/12/06	dom 24/12/06	Delfin
79		P Configuración	lun 11/12/06	dom 24/12/06	Delfin
80		Plantillas de código	lun 11/12/06	dom 24/12/06	Diego
81		Lenguajes Nuevo, Guardar, Abrir	lun 11/12/06	dom 24/12/06	Diego
82		Proyecto Nuevo, Guardar, Abrir y Cargar Auton	lun 11/12/06	dom 24/12/06	Delfin
83		Plantillas Nuevo, Guardar, Abrir	lun 11/12/06	dom 24/12/06	Diego
84		Multidioma	lun 11/12/06	dom 24/12/06	Diego
85		Buscar y Reemplazar	jue 21/12/06	dom 24/12/06	Delfin
86		Deshacer y Rehacer	jue 21/12/06	dom 24/12/06	Delfin
87		Configuración menus Nuevo, Guardar, Abrir y C	lun 11/12/06	dom 24/12/06	Juanjo
88		Pruebas	lun 25/12/06	vie 29/12/06	
89		Lenguajes GUI	lun 25/12/06	vie 29/12/06	Juanjo
90		L Palabras reservadas + coloreado	lun 25/12/06	vie 29/12/06	Diego
91		L Gramáticas BNF (Guardar reglas)	lun 25/12/06	vie 29/12/06	Juanjo
92		L Cargar tipos de archivo asociados para explor	lun 25/12/06	vie 29/12/06	Diego
93		L Activar la opción de menu correspondiente (cc	lun 25/12/06	vie 29/12/06	Juanjo
94		Proyecto GUI	lun 25/12/06	vie 29/12/06	Juanjo
95		P nombre	lun 25/12/06	vie 29/12/06	Delfin
96		P extension (*.pide)	lun 25/12/06	vie 29/12/06	Delfin
97		P Path del compilador o interprete y librerias	lun 25/12/06	vie 29/12/06	Delfin
98		P Configuración	lun 25/12/06	vie 29/12/06	Delfin
99		Plantillas de código	lun 25/12/06	vie 29/12/06	Diego
100		Lenguajes Nuevo, Guardar, Abrir	lun 25/12/06	vie 29/12/06	Diego
101		Proyecto Nuevo, Guardar, Abrir y Cargar Auton	lun 25/12/06	vie 29/12/06	Delfin
102		Plantillas Nuevo, Guardar, Abrir	lun 25/12/06	vie 29/12/06	Diego
103		Multidioma	lun 25/12/06	vie 29/12/06	Diego
104		Configuración menus Nuevo, Guardar, Abrir y C	lun 25/12/06	vie 29/12/06	Juanjo

Id		Nombre de tarea	Comienzo	Fin	Nombres de los recursos
105		Buscar y Reemplazar	vie 29/12/06	vie 29/12/06	Delfin
106		Deshacer y Rehacer	vie 29/12/06	vie 29/12/06	Delfin
107		Iteración 3	lun 01/01/07	dom 11/03/07	
108		Análisis	lun 01/01/07	dom 07/01/07	
109		reconocimiento de palabras en tiempo real	lun 01/01/07	dom 07/01/07	Diego
110		reconocimiento de errores en la sintaxis	lun 01/01/07	dom 07/01/07	Juanjo, Delfin y Diego
111		acoplamiento de parser para reconocimiento de	lun 01/01/07	dom 07/01/07	Juanjo, Delfin y Diego
112		Diseño	lun 08/01/07	dom 14/01/07	
113		reconocimiento de palabras en tiempo real	lun 08/01/07	dom 14/01/07	Diego
114		reconocimiento de errores en la sintaxis	lun 08/01/07	dom 14/01/07	Juanjo, Delfin y Diego
115		acoplamiento de parser para reconocimiento de	lun 08/01/07	dom 14/01/07	Juanjo, Delfin y Diego
116		Implementación 1	lun 15/01/07	dom 28/01/07	
117		reconocimiento de palabras en tiempo real	lun 15/01/07	dom 28/01/07	Diego
118		reconocimiento de errores en la sintaxis	lun 15/01/07	dom 28/01/07	Juanjo, Delfin y Diego
119		acoplamiento de parser para reconocimiento de	lun 15/01/07	dom 28/01/07	Juanjo, Delfin y Diego
120		Implementación 2	lun 26/02/07	dom 04/03/07	
121		reconocimiento de palabras en tiempo real	lun 26/02/07	dom 04/03/07	Diego
122		reconocimiento de errores en la sintaxis	lun 26/02/07	dom 04/03/07	Juanjo, Delfin y Diego
123		acoplamiento de parser para reconocimiento de	lun 26/02/07	dom 04/03/07	Juanjo, Delfin y Diego
124		Pruebas	lun 05/03/07	dom 11/03/07	
125		reconocimiento de palabras en tiempo real	lun 05/03/07	dom 11/03/07	Diego
126		reconocimiento de errores en la sintaxis	lun 05/03/07	dom 11/03/07	Juanjo, Delfin y Diego
127		acoplamiento de parser para reconocimiento de	lun 05/03/07	dom 11/03/07	Juanjo, Delfin y Diego
128		Iteración 4	lun 12/03/07	jue 26/04/07	
129		Análisis	lun 12/03/07	jue 26/04/07	
130		Análisis Sintáctico	lun 12/03/07	jue 26/04/07	Juanjo y Delfin
131		Casos de Uso	lun 12/03/07	jue 26/04/07	Juanjo, Delfin y Diego
132		Recocimiento de comentarios	lun 12/03/07	jue 26/04/07	Diego
133		Archivos XML para tokens del lenguaje	lun 12/03/07	jue 26/04/07	Diego
134		Historial de comandos	lun 12/03/07	jue 26/04/07	Diego
135		Diseño	lun 12/03/07	jue 26/04/07	
136		Análisis Sintáctico	lun 12/03/07	jue 26/04/07	Juanjo y Delfin
137		Casos de Uso	lun 12/03/07	jue 26/04/07	Juanjo, Delfin y Diego
138		Recocimiento de comentarios	lun 12/03/07	jue 26/04/07	Diego
139		Archivos XML para tokens del lenguaje	lun 12/03/07	jue 26/04/07	Diego
140		Historial de comandos	lun 12/03/07	jue 26/04/07	Diego
141		Implementación	lun 12/03/07	jue 26/04/07	
142		Análisis Sintáctico	lun 12/03/07	jue 26/04/07	Juanjo y Delfin
143		Casos de Uso	lun 12/03/07	jue 26/04/07	Juanjo, Delfin y Diego
144		Recocimiento de comentarios	lun 12/03/07	jue 26/04/07	Diego
145		Archivos XML para tokens del lenguaje	lun 12/03/07	jue 26/04/07	Diego
146		Historial de comandos	lun 12/03/07	jue 26/04/07	Diego
147		Pruebas	lun 12/03/07	jue 26/04/07	
148		Análisis Sintáctico	lun 12/03/07	jue 26/04/07	Juanjo y Delfin
149		Casos de Uso	lun 12/03/07	jue 26/04/07	Juanjo, Delfin y Diego
150		Recocimiento de comentarios	lun 12/03/07	jue 26/04/07	Diego
151		Archivos XML para tokens del lenguaje	lun 12/03/07	jue 26/04/07	Diego
152		Historial de comandos	lun 12/03/07	jue 26/04/07	Diego
153		Iteración 5	vie 27/04/07	vie 06/07/07	
154		Análisis	vie 27/04/07	vie 06/07/07	
155		Corrección de errores	vie 27/04/07	vie 06/07/07	Juanjo, Delfin y Diego
156		Análisis Sintáctico	vie 27/04/07	vie 06/07/07	Juanjo

Id		Nombre de tarea	Comienzo	Fin	Nombres de los recursos
157		Gestión de proyectos	vie 27/04/07	vie 06/07/07	Delfin
158		Ampliación de menus	vie 27/04/07	vie 06/07/07	Juanjo
159		pareado de corchetes parentesis y llaves	vie 27/04/07	vie 06/07/07	Diego
160		Doble ventana	vie 27/04/07	vie 06/07/07	Diego
161		Barra de estado	vie 27/04/07	vie 06/07/07	Delfin
162		Análisis Léxico de forma incremental	vie 27/04/07	vie 06/07/07	Diego
163		Memoria	vie 27/04/07	vie 06/07/07	Juanjo, Delfin y Diego
164		Diseño	vie 27/04/07	vie 06/07/07	
165		Corrección de errores	vie 27/04/07	vie 06/07/07	Juanjo, Delfin y Diego
166		Análisis Sintáctico	vie 27/04/07	vie 06/07/07	Juanjo
167		Gestión de proyectos	vie 27/04/07	vie 06/07/07	Delfin
168		Ampliación de menus	vie 27/04/07	vie 06/07/07	Juanjo
169		pareado de corchetes parentesis y llaves	vie 27/04/07	vie 06/07/07	Diego
170		Doble ventana	vie 27/04/07	vie 06/07/07	Diego
171		Barra de estado	vie 27/04/07	vie 06/07/07	Delfin
172		Análisis Léxico de forma incremental	vie 27/04/07	vie 06/07/07	Diego
173		Memoria	vie 27/04/07	vie 06/07/07	Juanjo, Delfin y Diego
174		Implementación	vie 27/04/07	vie 06/07/07	
175		Corrección de errores	vie 27/04/07	vie 06/07/07	Juanjo, Delfin y Diego
176		Análisis Sintáctico	vie 27/04/07	vie 06/07/07	Juanjo
177		Gestión de proyectos	vie 27/04/07	vie 06/07/07	Delfin
178		Ampliación de menus	vie 27/04/07	vie 06/07/07	Juanjo
179		pareado de corchetes parentesis y llaves	vie 27/04/07	vie 06/07/07	Diego
180		Doble ventana	vie 27/04/07	vie 06/07/07	Diego
181		Barra de estado	vie 27/04/07	vie 06/07/07	Delfin
182		Análisis Léxico de forma incremental	vie 27/04/07	vie 06/07/07	Diego
183		Memoria	vie 27/04/07	vie 06/07/07	Juanjo, Delfin y Diego
184		Pruebas	vie 27/04/07	vie 06/07/07	
185		Corrección de errores	vie 27/04/07	vie 06/07/07	Juanjo, Delfin y Diego
186		Análisis Sintáctico	vie 27/04/07	vie 06/07/07	Juanjo
187		Gestión de proyectos	vie 27/04/07	vie 06/07/07	Delfin
188		Ampliación de menus	vie 27/04/07	vie 06/07/07	Juanjo
189		pareado de corchetes parentesis y llaves	vie 27/04/07	vie 06/07/07	Diego
190		Doble ventana	vie 27/04/07	vie 06/07/07	Diego
191		Barra de estado	vie 27/04/07	vie 06/07/07	Delfin
192		Análisis Léxico de forma incremental	vie 27/04/07	vie 06/07/07	Diego
193		Memoria	vie 27/04/07	vie 06/07/07	Juanjo, Delfin y Diego

ACIDE

CONCLUSIONES

Versión: 1.0
Fecha: 05/07/2007

OBJETIVOS CUMPLIDOS

De los objetivos que nos propusimos al comenzar este proyecto hemos conseguido cumplir los siguientes:

- Editor de texto
- Menús
- Barra de herramientas
- Interfaz Gráfica de Usuario (GUI) para la parametrización
- Explorador del proyecto
- Números de línea
- Barra de estado
- Log
- Consola
- Archivos de configuración y parametrización
- Reconocimiento de las palabras reservadas
- Creación de proyectos
- Guardar y cargar configuraciones
- Intérprete y compilador
- Ejecución
- Ayuda

Cabe mencionar que uno de los objetivos que teníamos pensado terminar ha quedado pendiente por falta de tiempo:

- Reconocimiento de los errores de sintaxis

OBJETIVOS EXTRA

Al margen de los objetivos principales para el desarrollo de este proyecto hemos implementado algunas funcionalidades adicionales que nos parecen muy útiles:

- Comunicación con programas externos integrada en el propio entorno (a nivel de hilos)
- Configuración en archivos XML
- Búsqueda de expresiones regulares
- Doble vista del documento
- Impresión de documentos
- Pareado de paréntesis, corchetes y llaves

ACIDE

POSIBLES AMPLIACIONES

Versión: 5.0
Fecha: 05/07/2007

LISTA DE POSIBLES MEJORAS

- Historial de comandos
- Listas de archivos y proyectos recientes
- Multiselección de archivos para abrirlos
- Modificar tipo de letra del editor
- Añadir/Quitar comentario. Se seleccionan las líneas y con el menú contextual se decide añadir el símbolo de comentario (definido en el léxico) o quitarlo
- Aceptar arrastrar archivos y proyectos, y soltar en ACIDE
- Sangrado automático
- Modificar el tipo de fuente del léxico por grupos (para aplicarlo a categorías sintácticas diferentes, como los tipos, funciones, etc.)
- Organizar la barra de herramientas por secciones
- Comandos de la barra de herramientas: agregar opción para solicitar parámetro escrito, archivo o ruta seleccionados. Por ejemplo: comando cd seguido de una ruta.
- Diccionario de idioma
- Macros de comandos
- Obtención automática del léxico a partir de la gramática

ACIDE

BIBLIOGRAFIA

Versión: 5.0
Fecha: 05/07/2007

BIBLIOGRAFÍA

LIBROS

- JAVA2 Manual de usuario y Tutorial. Agustín Froufe. Editorial Ra-Ma Ed. 2000
- Piensa en Java. Bruce Eckel. Editorial PEARSON Prentice Hall 2ºEd.

WEBS

Generales

- <http://forum.java.sun.com/index.jspa>

Layouts

- <http://java.sun.com/docs/books/tutorial/uiswing/layout/using.html>

look & feel

- <http://java.sun.com/docs/books/tutorial/uiswing/learn/example2.html>

Direcciones Swing

- <http://www.programacion.com/java/tutorial/swing/>
- <http://java.sun.com/docs/books/tutorial/uiswing/components/button.html>
- <http://java.sun.com/javase/6/docs/api/javax/swing/JCheckBox.html>
- <http://java.sun.com/docs/books/tutorial/uiswing/layout/grid.html>
- <http://java.sun.com/docs/books/tutorial/uiswing/layout/using.html#choosing>
- <http://java.sun.com/docs/books/tutorial/uiswing/components/label.html>
- <http://java.sun.com/j2se/1.4.2/docs/api/javax/swing/JLabel.html>
- <http://java.sun.com/j2se/1.5.0/docs/api/javax/swing/JTextArea.html>
- <http://java.sun.com/docs/books/tutorial/uiswing/layout/flow.html>
- <http://java.sun.com/docs/books/tutorial/uiswing/layout/using.html>
- <http://java.sun.com/j2se/1.4.2/docs/api/java/awt/GridLayout.html>
- <http://java.sun.com/j2se/1.4.2/docs/api/java/awt/BorderLayout.html>

Pestañas

- <http://www.itapizaco.edu.mx/paginas/JavaTut/froufe/parte14/cap14-14.html>
- <http://www.programacion.com/java/tutorial/swing/23/>

JFileChooser

- <http://www.javahispano.org/articles.article.action?id=13>
- <http://www.javahispano.org/articles.article.action?id=14>

log4j

- <http://java.sun.com/j2se/1.4.2/docs/guide/util/logging/>
- <http://www.javahispano.org/articles.article.action?id=9>
- <http://www.javahispano.org/articles.article.action?id=10>
- <http://www.vimeworks.com/~mauricio/manualLog4J.pdf>
- <http://logging.apache.org/log4j/docs/>
- <http://logging.apache.org/log4j/docs/manual.html>

Otras aplicaciones relacionadas

- http://es.wikipedia.org/wiki/Entorno_integrado_de_desarrollo
- <http://www.kdevelop.org/>
- <http://anjuta.sourceforge.net/>
- <http://www.jcreator.com/>
- <http://packages.qa.debian.org/c/crimson.html>
- http://www.borland.com/downloads/download_jbuilder.html
- <http://www.eclipse.org>
- <http://www.crimsoneditor.com/>
- <http://geany.uvena.de/Main/HomePage>
- <http://www.jedit.org>

Gramáticas en BNF

- www.lcc.uma.es/~vicente/docencia/
- <http://www.dsic.upv.es/asignaturas/facultad/prg/pav/pracs/kkk/gramatica>

Parsing

- <http://www.lcc.uma.es/~galvez/theme/cup/gramaticacup/genparsercup.html>
- <http://coleweb.dc.fi.udc.es/cole/library/html/Alo94/node48.html>
- <http://www.microsoft.com/spain/technet/itsshowtime/session1.aspx?videoid=265>

antlr

- <http://www.antlr.org/>
- <http://www.lsi.us.es/~troyano/documentos/guia.pdf>
- http://pablobm.com/blog/libro_gratuito_en_espanol_sobre_antlr?PHPSESSID=69e3e42c0a9f38b7ea3c79693c90a2a0
- <http://www.suigeneris.org/kb/display/UCABTI/Recomendaciones+sobre+ANTLR>
- <http://www.jguru.com/faq/ANTLR>
- <http://www.cs.usfca.edu/~parrt/course/652/lectures/antlr.html>

ACIDE

APÉNDICE 1: MANUAL DE USUARIO

Versión: 5.0
Fecha: 05/07/2007

REQUISITOS DEL SISTEMA

Usuario

Esta aplicación no requiere una configuración especial para el sistema ni unos requisitos especiales dado que se proporciona un archivo autoejecutable con el que se puede usar la herramienta sin ningún tipo de problema.

El único requisito exigido es tener instalada la máquina virtual de Java (JVM) dado que el propio archivo comprimido de Java la necesita para poder ejecutarse. Para poder instalar la JVM es necesario bajarse de la página de Sun (<http://www.java.com/es/download/manual.jsp>) el instalador correspondiente a alguna de las versiones de Java Runtime Environment (JRE) iguales o posteriores a la versión 1.5. Con esta simple instalación se podrá trabajar con el entorno de desarrollo, pero no será posible trabajar con configuraciones sintácticas.

Para poder usar ACIDE junto con una configuración sintáctica adecuada para el lenguaje con el que se quiera trabajar es necesario tener instaladas también dos herramientas más de java: la de compilación (javac.exe) y la de creación y manipulación de archivos comprimidos de java (jar.exe). Estas dos características vienen incluidas en el archivo de instalación de Java Development Kit (JDK) que de nuevo podremos descargar de la página de Sun (<http://www.java.com/es/download/manual.jsp>) y que también viene proporcionado en el CD. Por supuesto siempre tenemos que trabajar con versiones posteriores a JDK 1.5.

Desarrollador

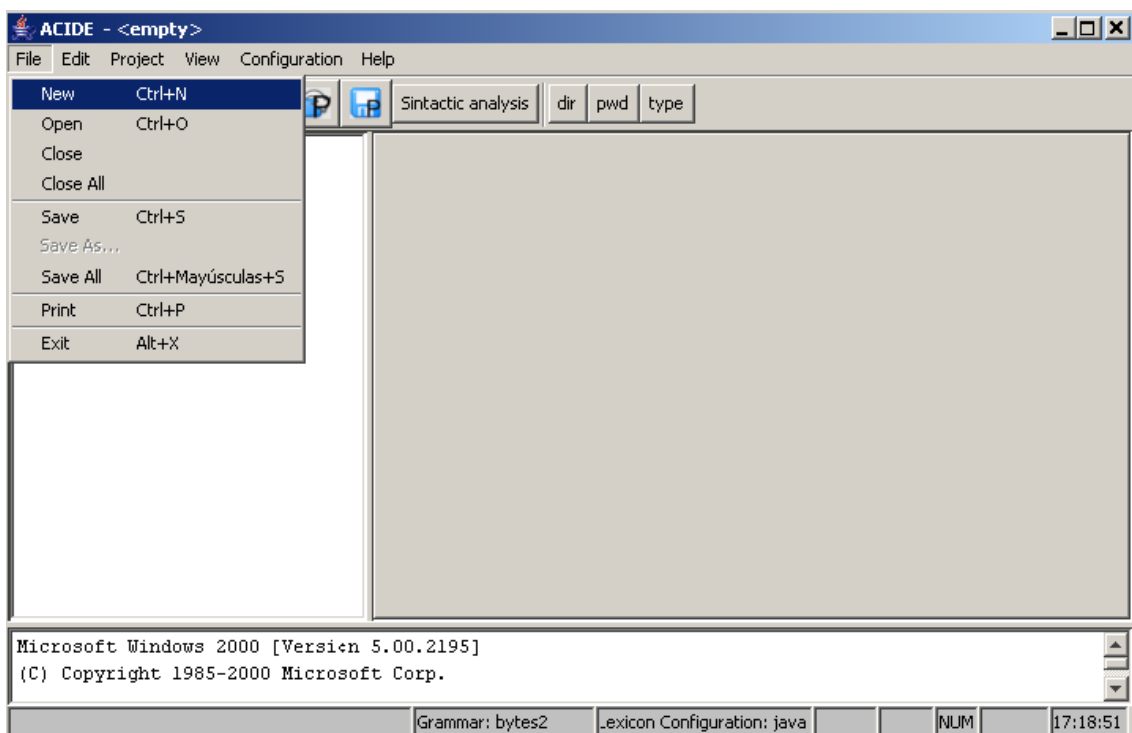
En el caso de ser desarrollador de la aplicación será necesario tener instalado como requisito obligatorio el JDK (versión posterior posterior a 1.5) y una herramienta de desarrollo de código a gusto del programador para la edición de código.

MENÚ ARCHIVO

Opción Archivo (File):

En la opción de Archivo se encuentran las opciones básicas del editor de texto: Nuevo, Abrir, Cerrar, Imprimir...

Como se puede comprobar tiene accesos directos.



El menú File: Dispone de las siguientes submenús:

- **Nuevo (New):** Crea un nuevo documento.
- **Abrir (Open):** Abre un documento anteriormente almacenado.
- **Cerrar (Close):** Cierra un documento abierto, y pregunta si se quiere guardar, si ha sido modificado anteriormente.
- **Cerrar Todo (Close All):** Cierra todos los documentos abiertos, pregunta si se quiere guardar si ha sido modificado anteriormente.

- **Guardar (Save):** Guarda un documento anteriormente salvado, en la misma ruta establecida.
- **Salvar como (Save As):** Guarda un documento en un path diferente.
- **Guardar Todo (Save All):** Guarda todos los documentos abiertos.
- **Imprimir (Print):** Imprime un documento.
- **Salir (Exit):** Sale de la aplicación

Opción Imprimir (Print):

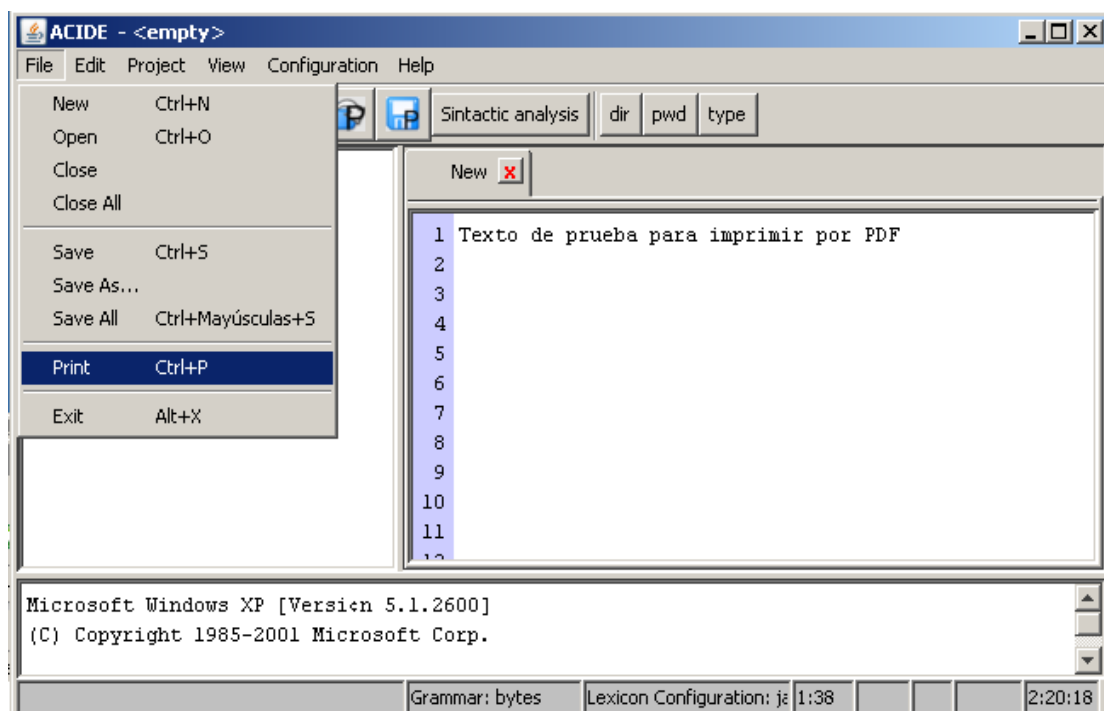
Los pasos son los siguientes:

Se crea un nuevo documento o se abre uno existente. Se puede imprimir por impresora o en formato PDF (En pantalla).

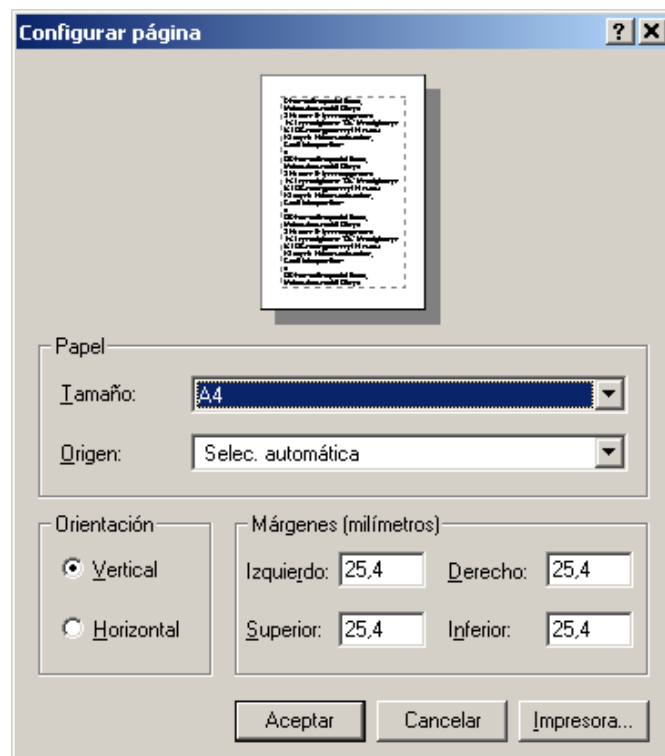
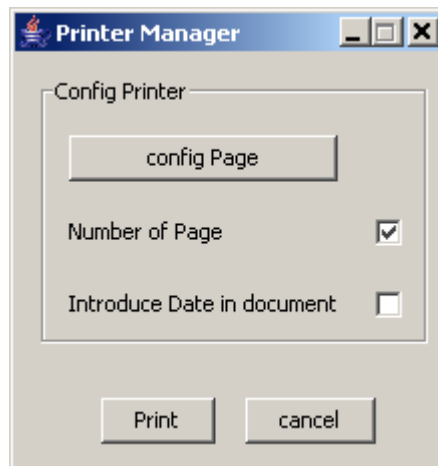
En este caso se ha creado un documento nuevo, el cual se va a imprimir.

- Se selecciona la opción de Print del menú File

En el ejemplo siguiente vemos cómo funciona el imprimir un documento en formato PDF.

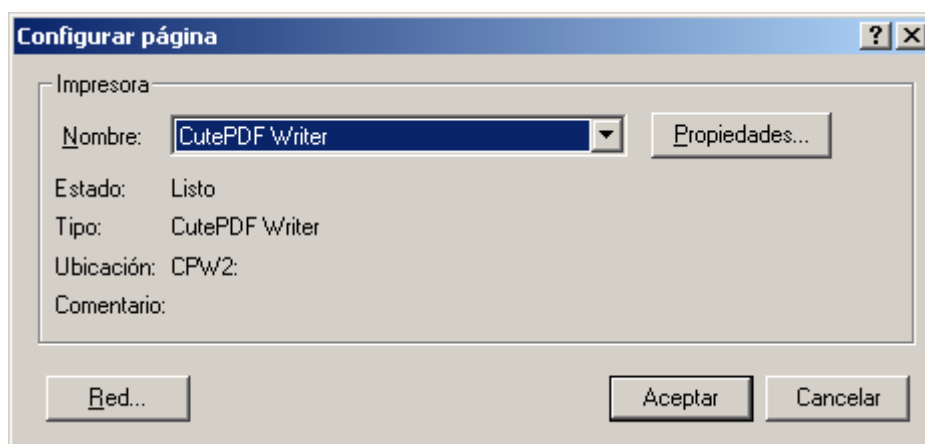


Para elegir las configuraciones de impresión se pulsa sobre el botón configuración de Página (**config Page**)



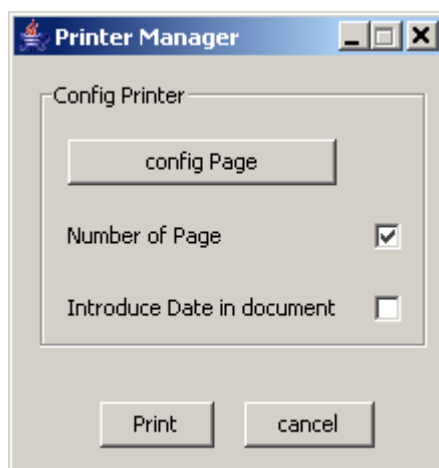
Se eligen las opciones de impresión apropiadas: tamaño, origen del papel, orientación (Vertical u Horizontal) y los márgenes.

Para seleccionar la impresión por PDF se usa la opción de impresora. Al pulsarla se muestra la siguiente ventana.



En esta ventana se elige el modo de impresión, en este caso CutePDF Writer. Después de pulsa en aceptar.

Después se selecciona lo que se quiera imprimir: número de página y/o fecha del documento.

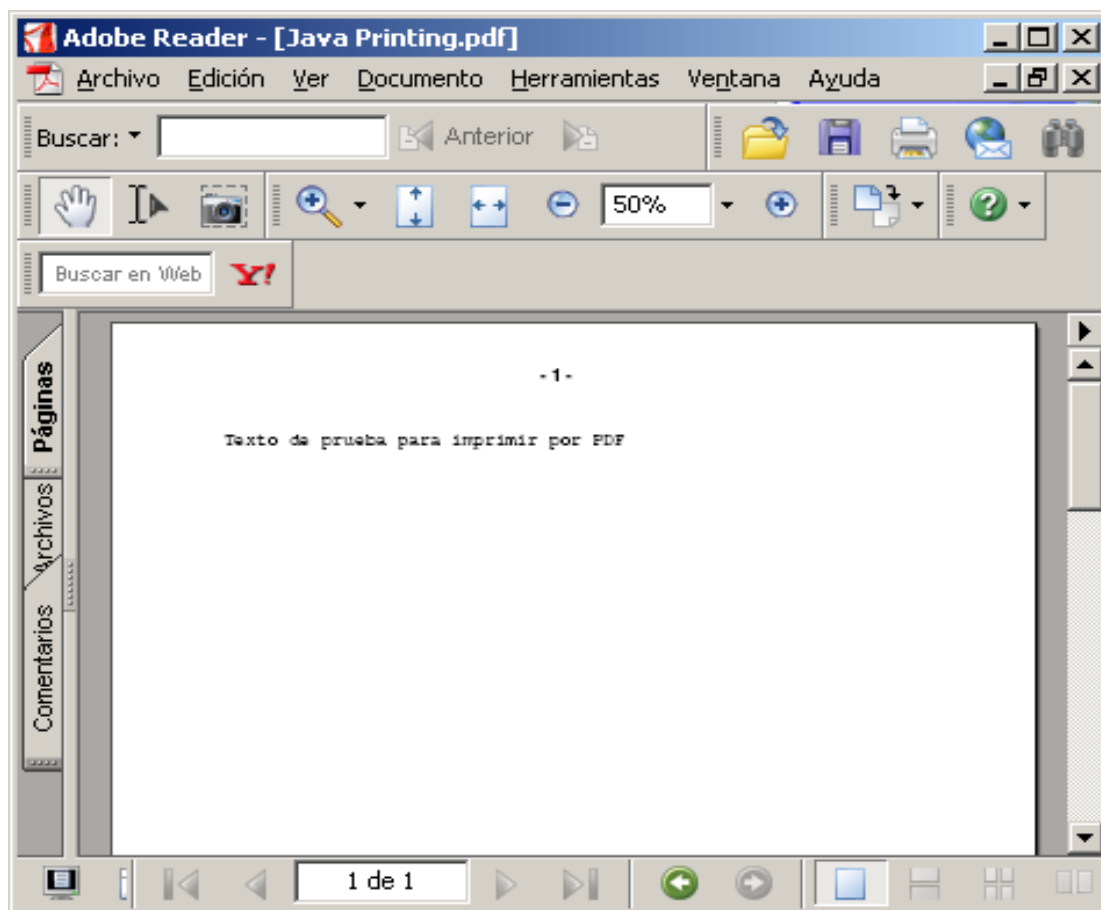


En este caso se selecciona el Número de Página.

Finalmente se pulsa en Imprimir (Print).

Nos preguntará con qué nombre y dónde queremos guardar el PDF.

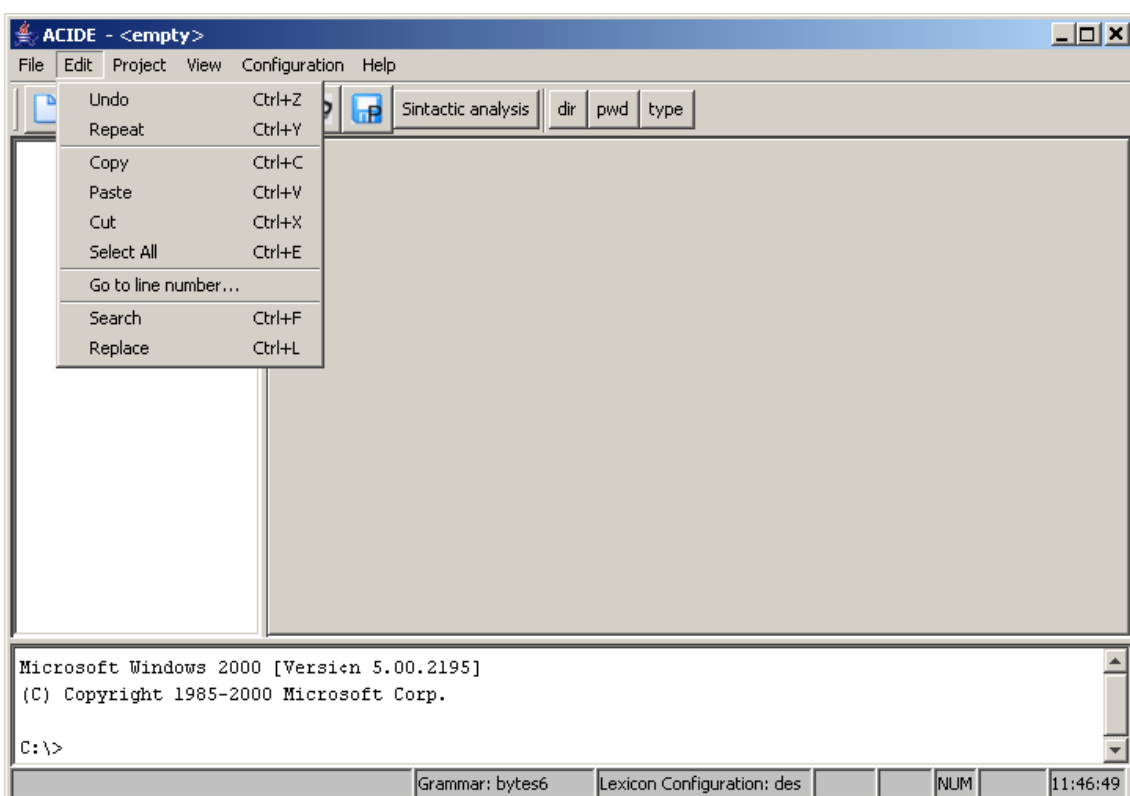
El resultado del PDF es el siguiente:



MENÚ EDICIÓN

Todo editor de texto necesita unas opciones de edición para el propio texto que contienen, así nuestro entorno de desarrollo proporciona algunas opciones bastante útiles en este sentido.

La apariencia y localización de estas utilidades se muestran a continuación en la siguiente imagen:



El primer grupo de operaciones lo conforman las opciones de deshacer y rehacer (cada una con sus teclas abreviadas correspondientes). La función de estas opciones son deshacer los últimos cambios realizados por el usuario en el archivo o bien rehacer algún cambio deshecho previamente en caso de arrepentimiento.

A continuación nos encontramos con las opciones de:

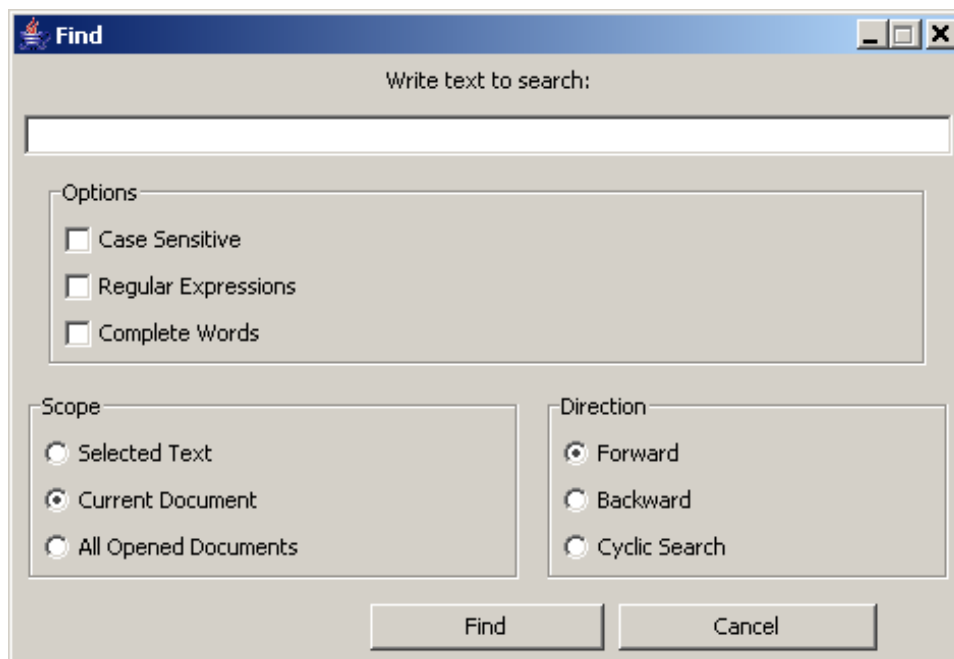
- Copiar (Copy): copia el texto seleccionado en el editor al portapapeles.

- Pegar (Paste): pega un texto almacenado en el portapapeles en la posición en la que tengamos colocado el cursor.
- Cortar (Cut): corta el texto seleccionado en el editor al portapapeles (es decir, lo copia al portapapeles y lo elimina del editor).
- Seleccionar todo (Select All): selecciona todo el texto contenido en el editor para realizar cualquiera de las operaciones anteriores.

Seguidamente tenemos una opción aislada: Ir a número de línea. Al seleccionarla podremos introducir el número de línea del documento en foco que queremos que se muestre en el editor y posicionar el cursor en esa línea para continuar la edición desde ese punto.

Por último el menú de edición muestra otras dos opciones más, que tienen una funcionalidad algo más extendida y completa y por ello poseen una interfaz aparte para poder realizar las operaciones correctamente. Por ello les dedicamos dos subcapítulos aparte.

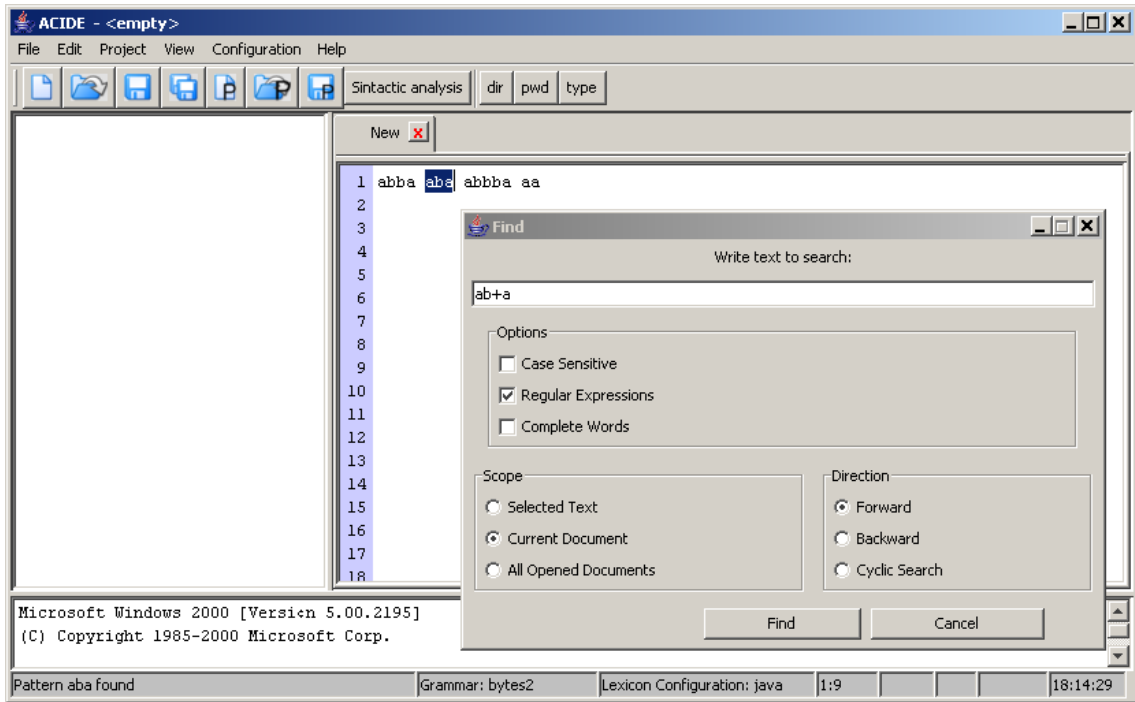
Buscar (Search):



La opción de buscar dispone de varias opciones:

- Cuadro de texto (Text Field): Es aquí donde se introduce el texto a buscar
- Opciones (Options):
 - Sensible a Mayúsculas/Minúsculas (Case Sensitive) Esta opción se utiliza para buscar cadenas sin tener en cuenta a las Mayúsculas/Minúsculas o teniendo en cuenta.
 - Expresiones Regulares (Regular Expressions): Busca expresiones regulares asociadas a un patrón de búsqueda.
 - Palabras completas (Complete Words). Busca sólo palabras completas.
- Alcance (Scope):
 - Texto Seleccionado (Selected Text): Busca en un texto seleccionado.
 - Documento actual (Current Document): Busca en una posición determinada del documento activo.
 - Todos los documentos abiertos (All Opened Documents): Busca en todos los ficheros abiertos.
- Dirección (Direction)
 - Hacia adelante (Forward): Busca desde el cursor actual hasta el final del documento.
 - Hacia atrás (Backward): Busca desde la posición del cursor actual hasta principio de documento.
 - Cíclico (Cyclic Search): Busca desde la posición actual del cursor hasta el final del documento, y empieza desde el principio hasta la posición de partida.

En el siguiente ejemplo se buscan cadenas asociadas al patrón `ab+a` (a seguido de al menos una b o más y terminado con otra a).

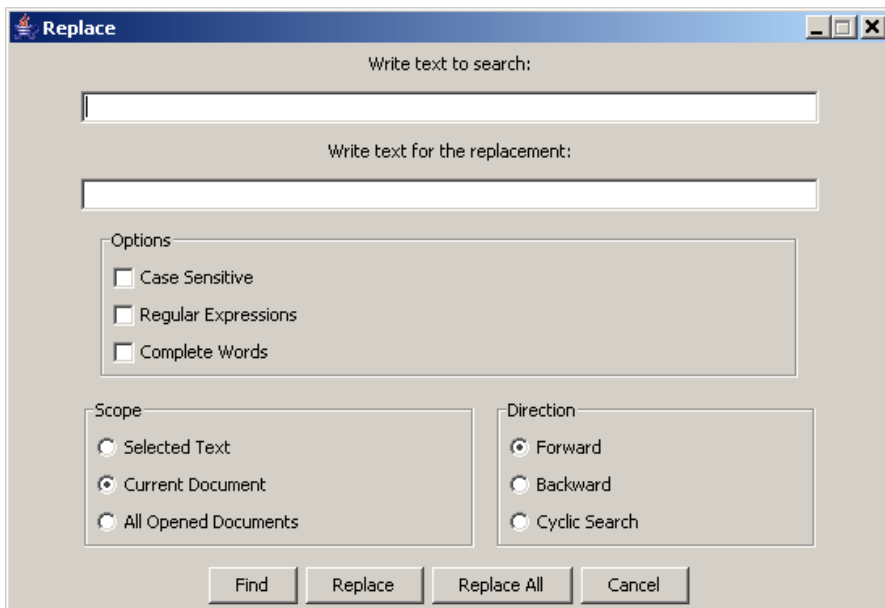


Replace:

Dispone de dos cuadros de texto:

El primero corresponde a buscar un texto para su posible reemplazo.

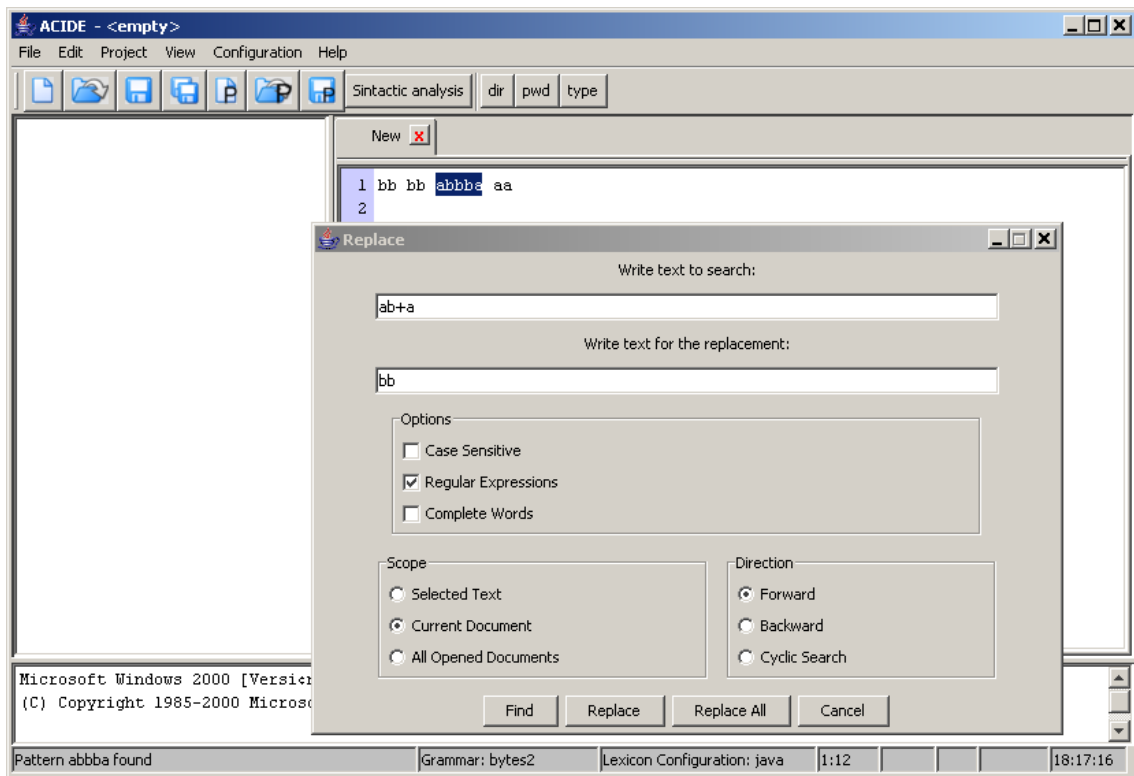
El segundo corresponde a la cadena por la que se reemplaza.



Dispone de las mismas opciones que el buscar.

Además se pueden reemplazar una o todas las cadenas.

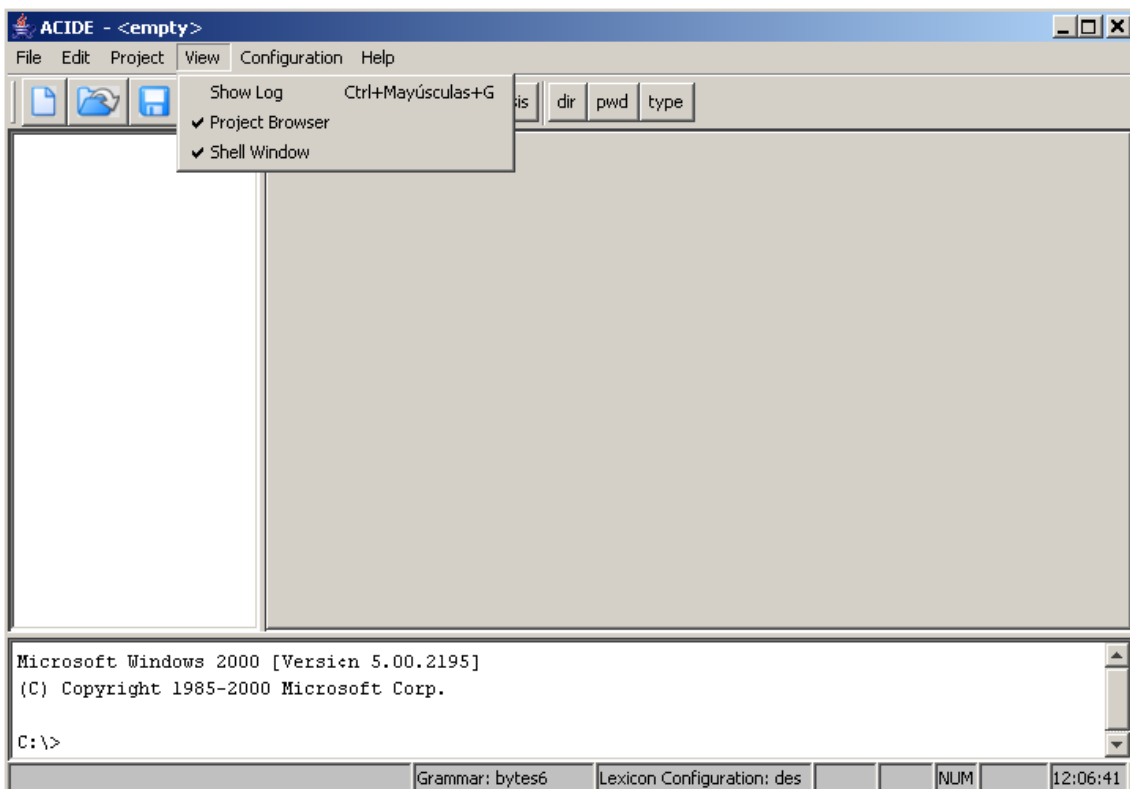
En el ejemplo siguiente se reemplaza el patrón `ab+a` por la cadena `bb`.



MENÚ VER

Otro de los menús que ofrece el entorno es el que controla las partes visibles de la ventana principal de la aplicación así como la visualización del log de las operaciones realizadas por el propio programa.

Así nos encontraremos con el siguiente menú al seleccionar el menú Ver (View):



Las tres opciones que se muestran son:

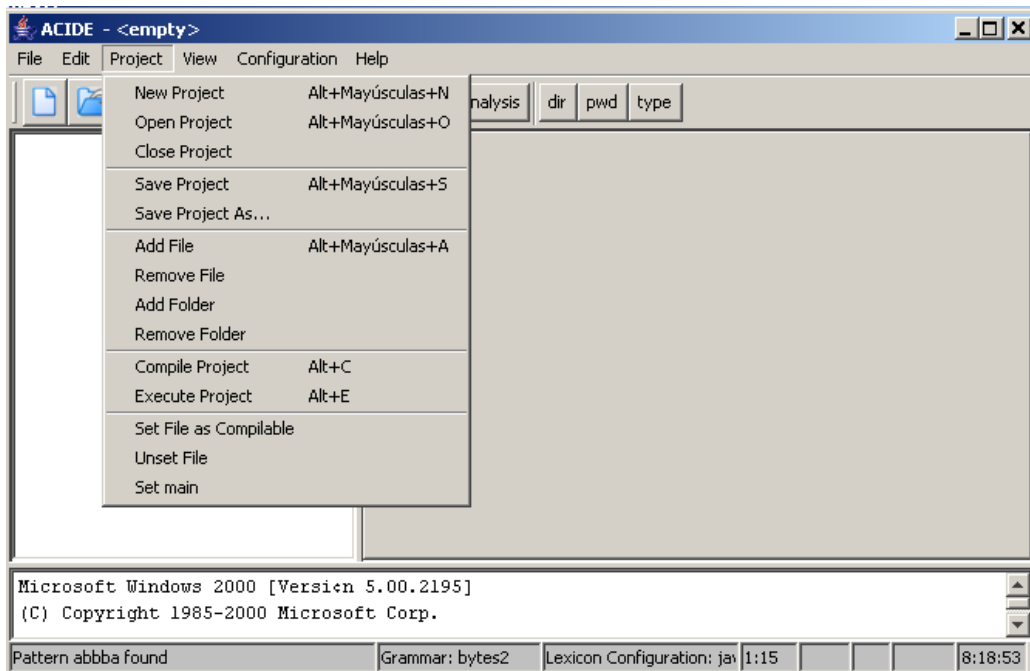
- **Mostrar Log (Show Log):** muestra en una nueva pestaña del editor el log de la ejecución actual del programa. En él están listadas cada una de las operaciones que realiza el entorno para cada una de las acciones que realiza el usuario con el mismo. En caso de ser necesario acceder a logs de ejecuciones anteriores podremos hacerlo abriéndolas como si fueran un archivo normal desde el directorio de instalación de ACIDE, en la subcarpeta log.
- **Explorador de proyecto (Project Browser):** muestra u oculta el explorador de proyectos situado en la parte izquierda de la ventana.

- Ventana de consola (Shell Window): muestra u oculta la ventana de consola situada en la parte inferior de la ventana.

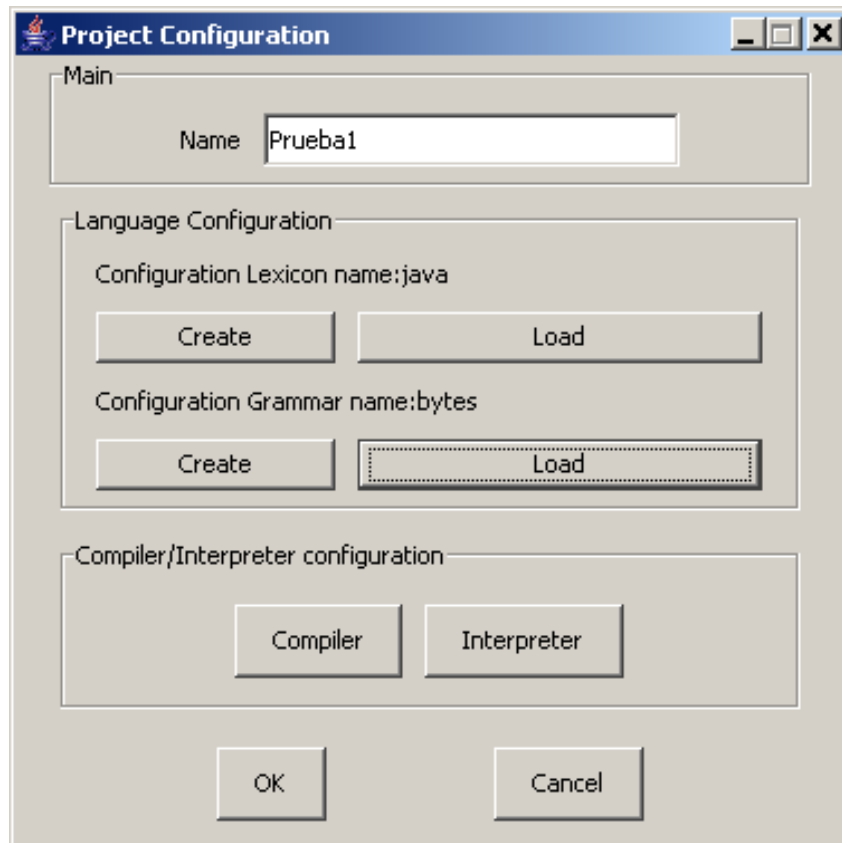
GESTOR DE PROYECTOS

Proyecto (Project):

Al marcar la opción del menú se despliegan las opciones del Gestor de proyectos.



Al marcar la opción del menú Nuevo Proyecto (Nuevo Project) aparece la siguiente ventana:



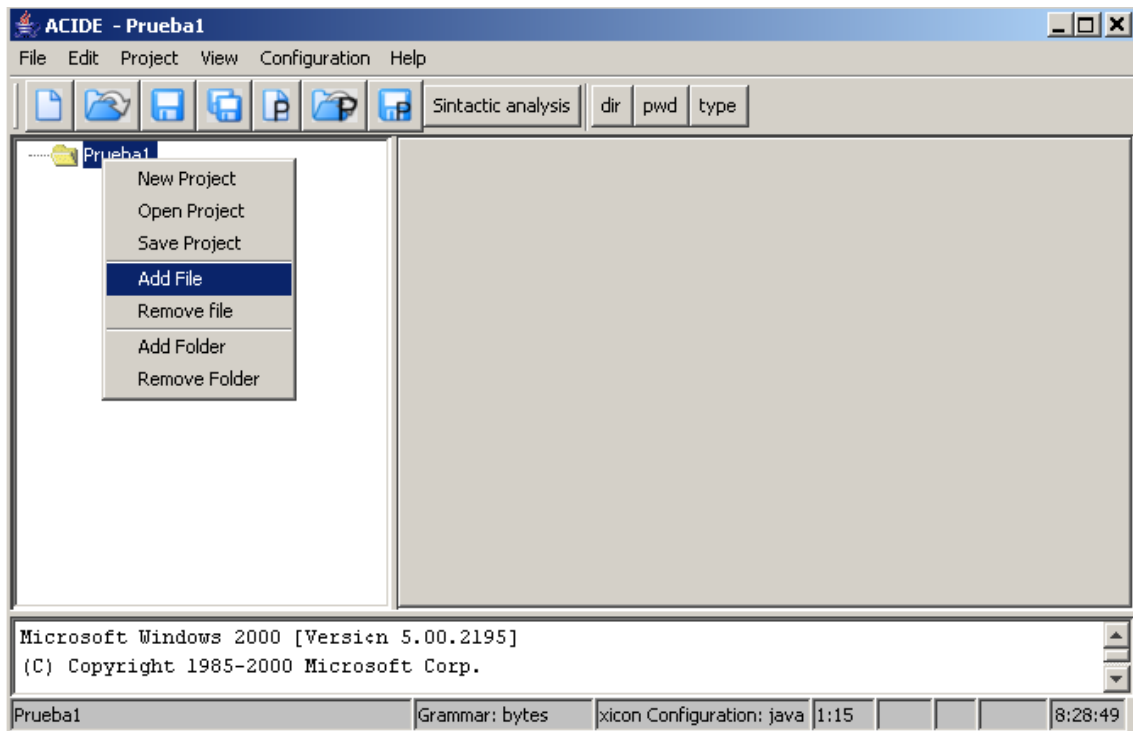
Las opciones de la ventana anterior son las siguientes:

- Nombre (Name): Indica el nombre del proyecto.
- Configuración del lenguaje (Lenguaje Configuration)
 - Configuración del léxico: Crear, cargar una existente.
 - Configuración de la gramática : Crear, cargar una existente.
- Opciones de compilador /Interprete (Compiler/Interpreter configuration)
 - Compilación (Compiler)
 - Intérprete (Interpreter)

Una vez creado el proyecto, en este caso Prueba1, se puede: añadir/eliminar ficheros y/o carpetas.

Para añadir un fichero se utiliza la opción añadir Fichero (Add File) del menú contextual o desde el menú de Project.

NOTA: al añadir un fichero al proyecto debemos estar colocados en una carpeta, en cambio para eliminar un fichero debemos marcar el fichero a eliminar.

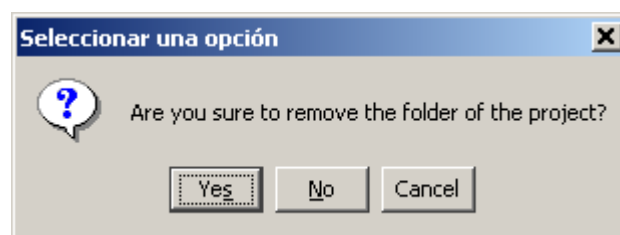


Igualmente para añadir una carpeta, debemos marcar la carpeta en la cual queremos crear la subcarpeta.

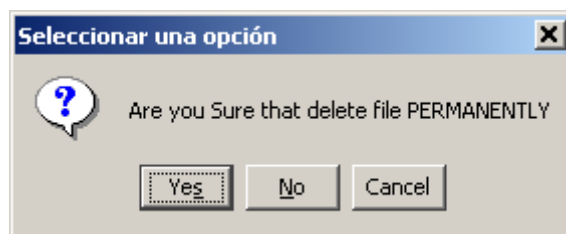
Para eliminar la subcarpeta debemos seleccionar la carpeta a eliminar.



Cuando se elimina una carpeta nos da un mensaje de confirmación:



Si además queremos borrar físicamente todos los ficheros asociados a esa carpeta, nos muestra otro mensaje de confirmación para mayor seguridad de la operación.

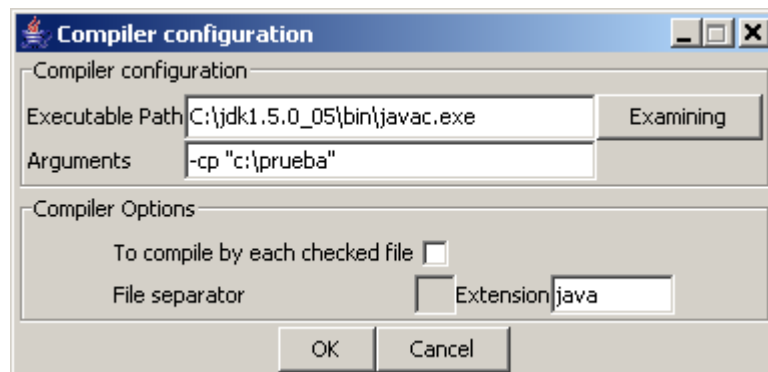
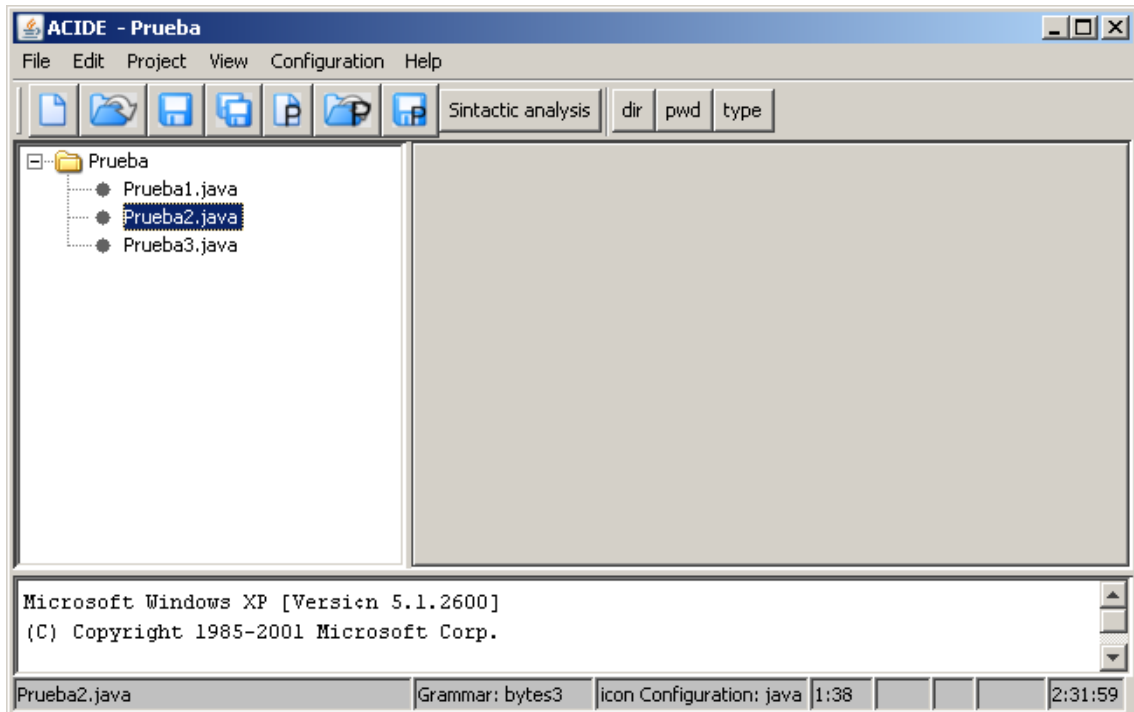


NOTA: El funcionamiento del eliminar carpetas se comporta de forma semejante al de eliminar ficheros.

Para guardar un proyecto debemos indicarle la extensión “acidePrj”, extensión que utiliza Acide para guardar los proyectos.

Ejemplo de compilación: (desde Gestor de Proyectos)

a) Compilar según extensión.

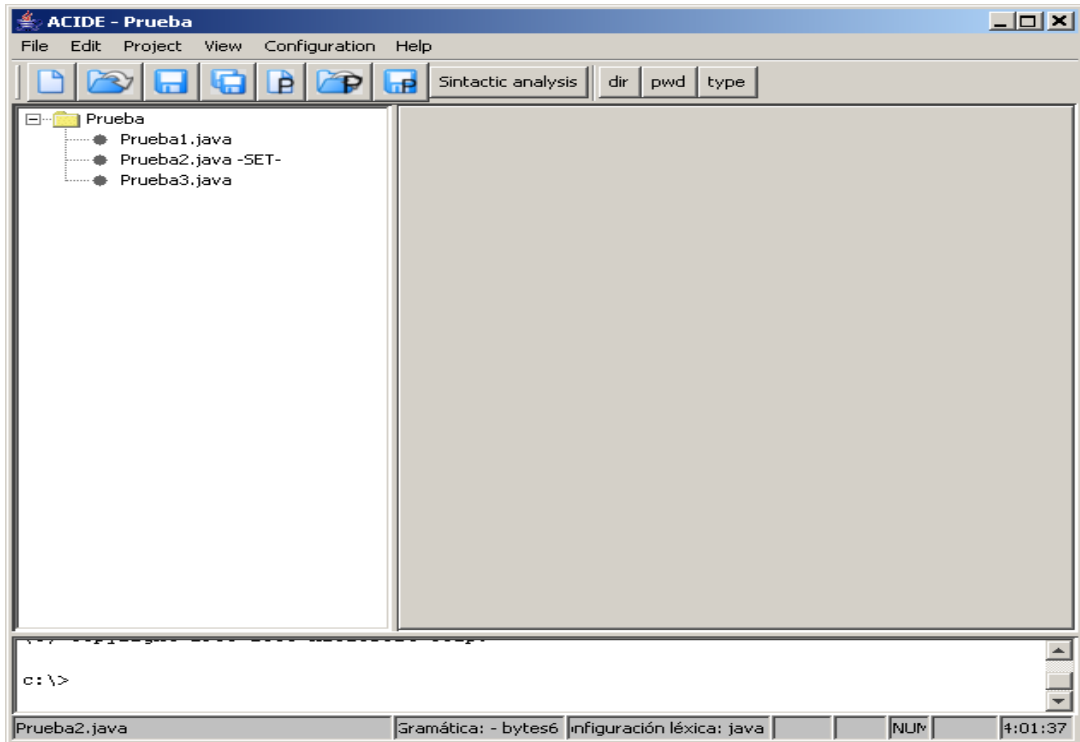


En la figura anterior se desactiva la opción de compilar para cada fichero marcado y se elige la extensión de java para compilar.

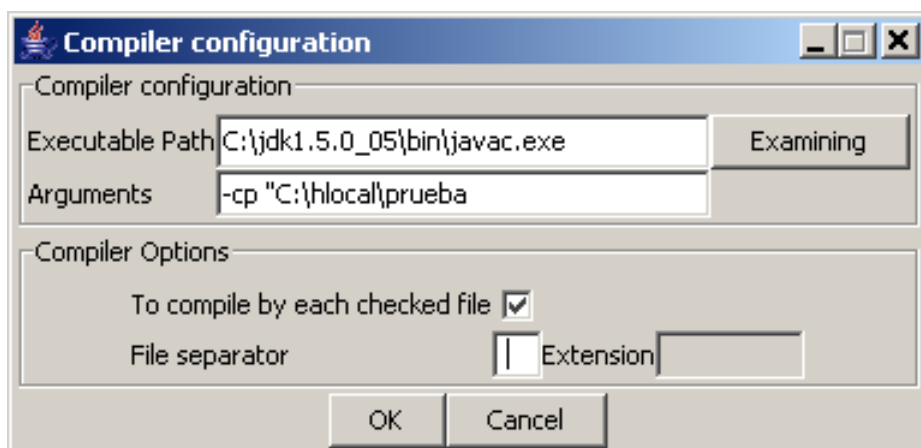
Finalmente se compila desde la opción compile Project del menú Project.

b) Compilar según “ficheros marcados para compilar”.

Para marcar un fichero se usa la opción de marcar fichero en la opción Menú proyecto.



En el la siguiente figura se muestran las opciones de compilación:

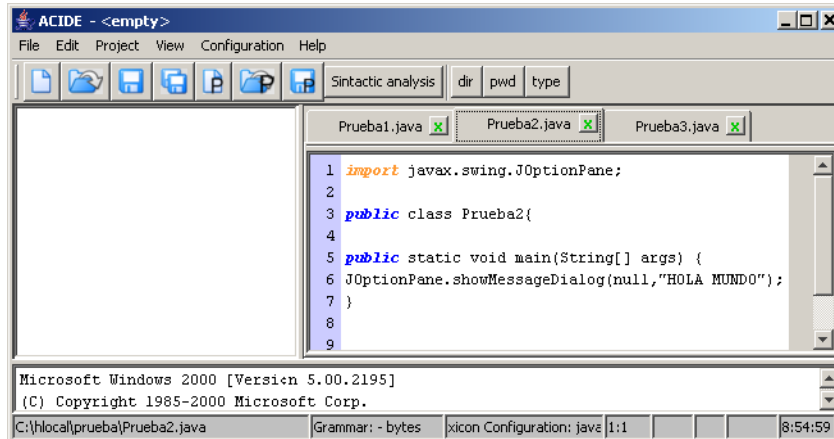


Como se puede comprobar se selecciona la opción de compilar para cada fichero marcado y se usa un espacio en blanco para el separador de ficheros.

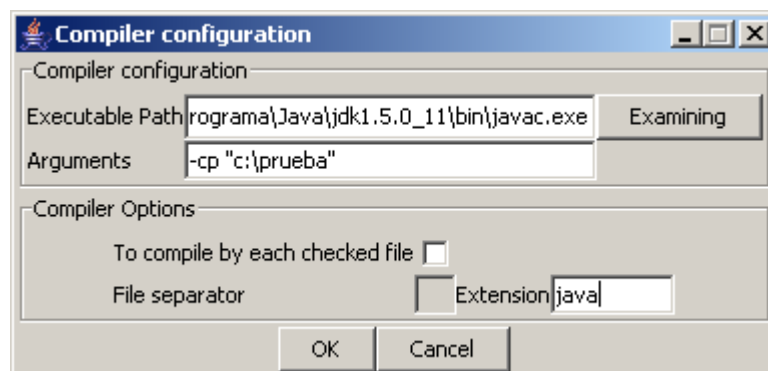
Después se compila en la opción del menú Project de compile Project

Ejemplo de compilación (desde configuración <Empty>)

a) Compilar según extensión.



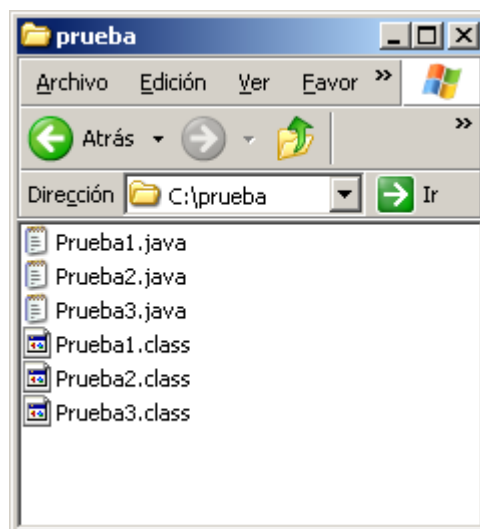
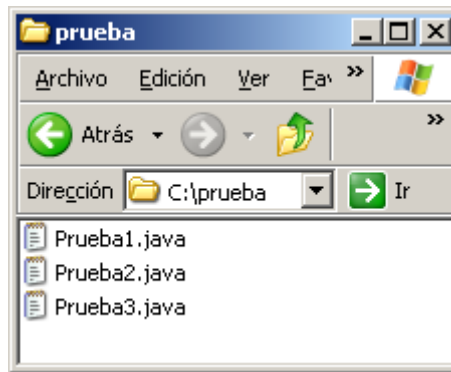
Una vez abiertos los ficheros se elige la opción de Compiler del menú Configuración. Con este proceso se muestra la siguiente ventana:



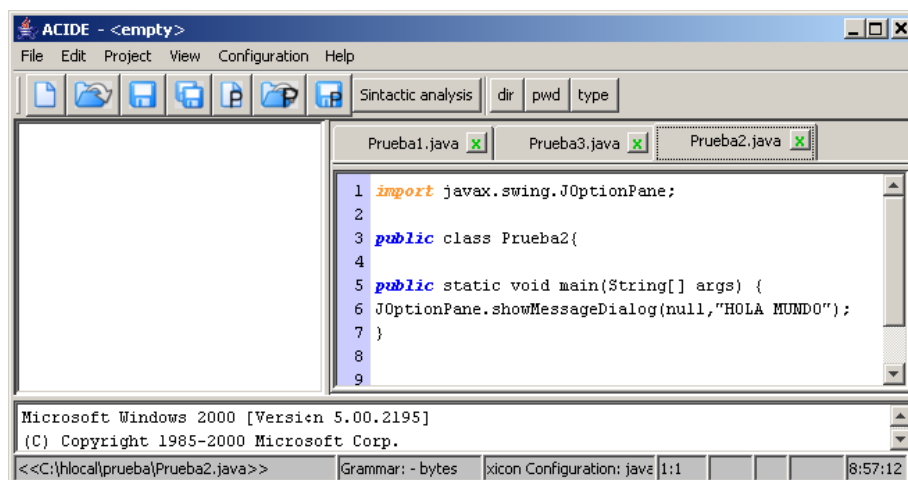
En este caso se ha elegido el compilador de Java y se ha puesto “ClassPath” como argumento.

Nota: No se activa la opción de “compilar por cada fichero marcado”, por lo tanto se pone “java” en el cuadro de texto como extensión.

Resultado de la compilación:



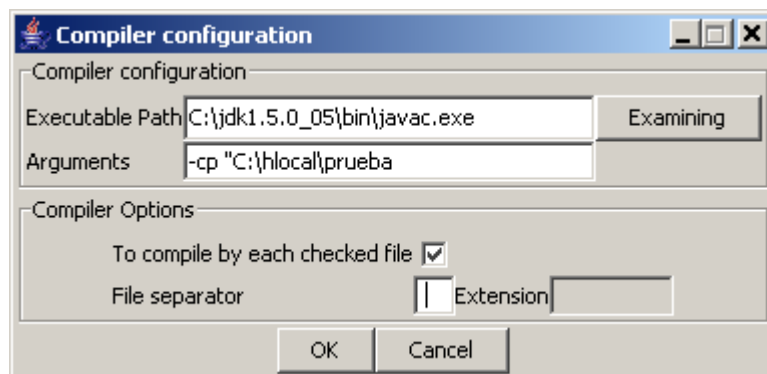
b) Compilar según “archivos marcados para compilar”.



En este caso se selecciona el fichero que queremos marcar como compilable.

El resultado se ve en la figura anterior, donde el path del fichero está señalado con “mayor” y “menor”.

En la configuración del compilador se marca la casilla de “compilar por cada fichero marcado” y se selecciona el separador de fichero.

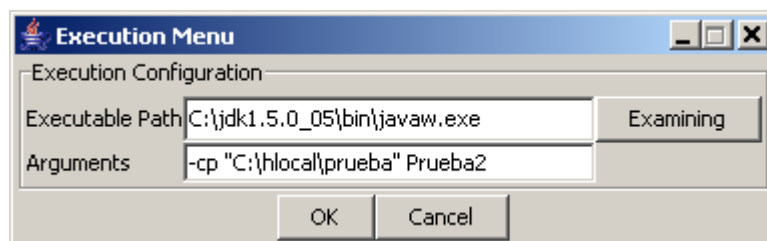


Ejemplo de ejecución

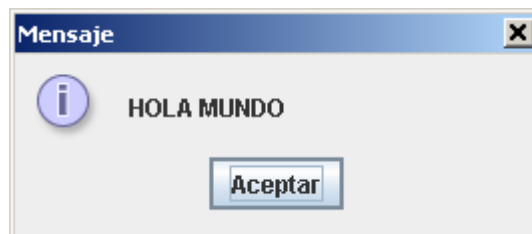
En la ruta del ejecutable (Executable Path) se pone el programa que carga el nuevo fichero compilado:

En argumentos se pone las diferentes opciones de la ejecución:

En este caso se ha establecido el classPath y la clase que contiene el método Main, en este caso es Prueba2.



El resultado de la ejecución se muestra a continuación:

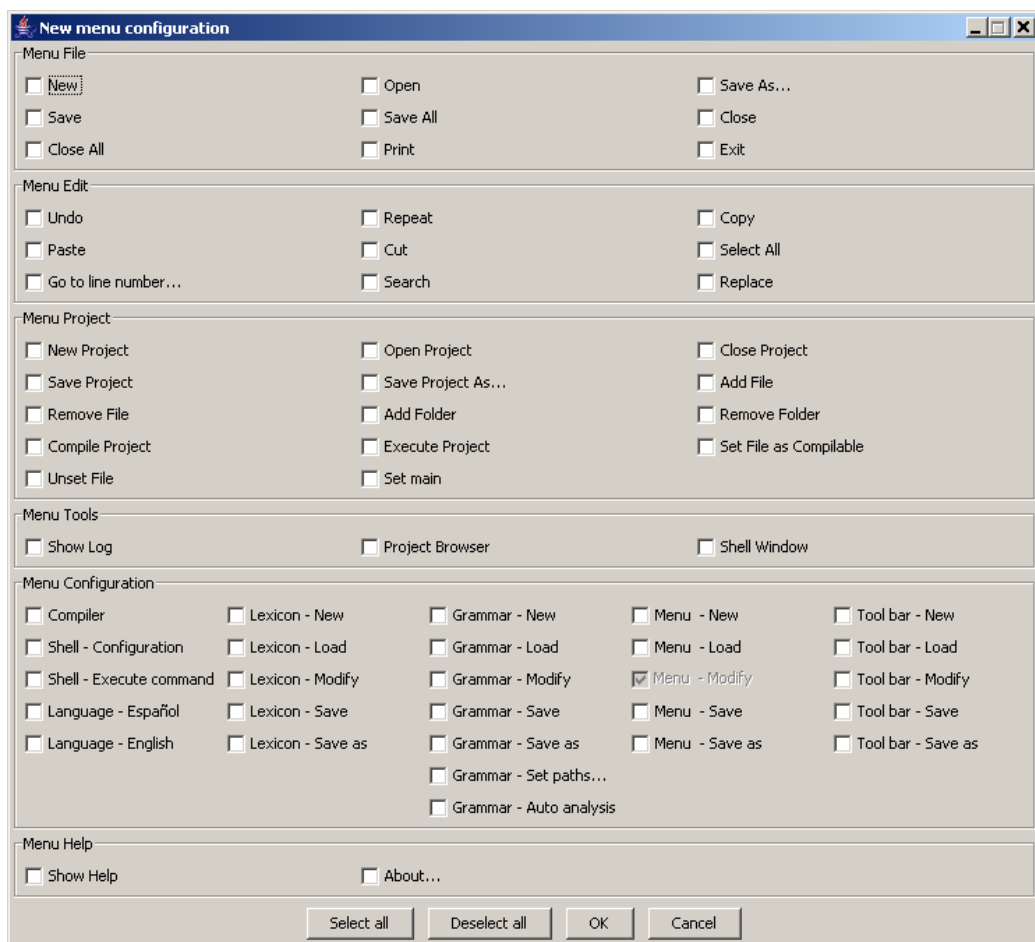


CONFIGURACIÓN DE MENÚ

El menú ofrece todas las opciones disponibles que el sistema presenta al usuario. Pero cada usuario puede no querer todas las opciones del menú visibles en cada momento, por eso ACIDE permite seleccionar cuáles de todas esas opciones el usuario desea tener activas.

Para poder definir una configuración de menú podremos hacerlo desde el submenú Configuración → Menú (Configuration → Menu) en el que encontraremos varias opciones:

- Nuevo (New): podemos definir una nueva configuración de menú sin partir de ninguna existente. Al seleccionar esta opción se mostrará la siguiente ventana:



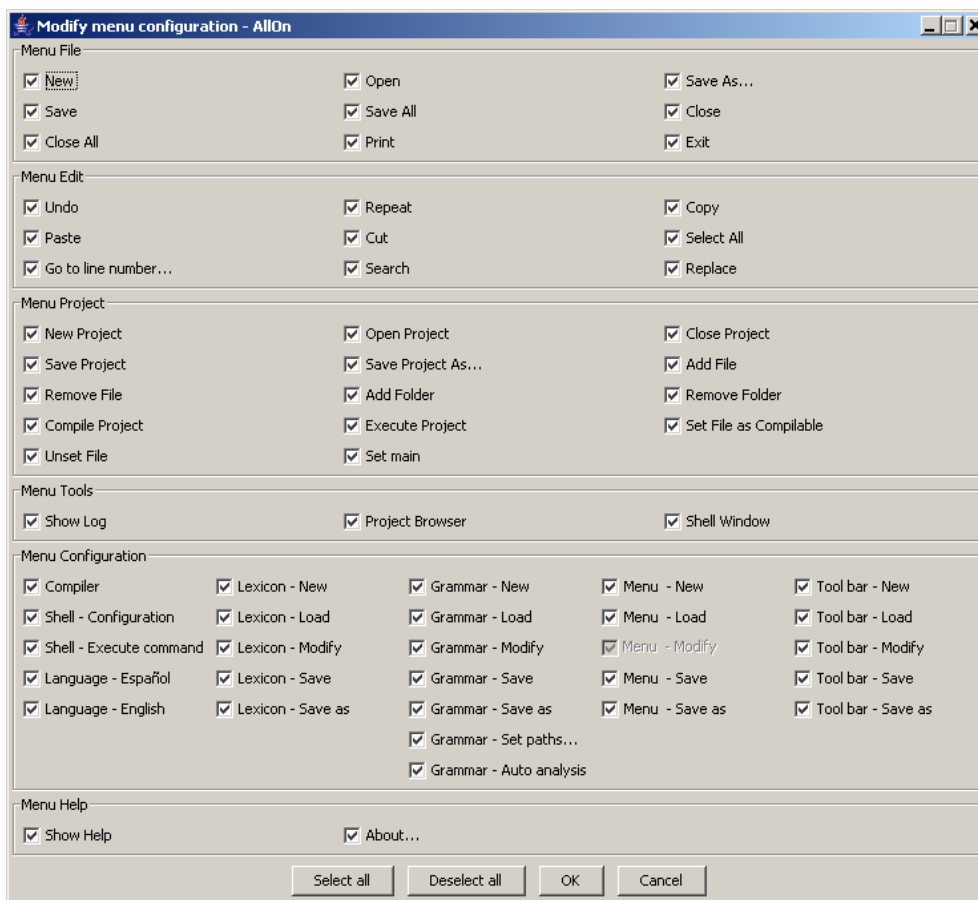
Aquí simplemente tendremos que marcar cada una de las opciones de menú que queramos tener visibles. Para ello también podemos usar los botones situados en

la parte inferior de la ventana que nos permiten seleccionar todas las opciones existentes o bien no seleccionar ninguna.

La nueva configuración de menú no quedará guardada hasta que el usuario lo haga explícitamente con la opción de configuración de menú correspondiente. En caso de cerrar la aplicación con la nueva configuración sin guardar, ésta se perderá y se recuperará la última configuración guardada.

Los archivos de configuración de menú tienen extensión menuCfg.

- Cargar (Load): permite cargar una configuración, a través del diálogo de sistema, de menú previamente guardada por el usuario.
- Modificar (Modify): es posible modificar una configuración de menú previamente definida. Para ello se muestra una pantalla similar a la de creación de nueva configuración de menú en la que aparecerían marcadas únicamente las opciones visibles en esa configuración de menú. También se muestra en la barra de título el nombre de la configuración de menú que estemos modificando



Al modificar una configuración original no se guardarán los cambios hasta que el usuario lo haga de forma explícita usando las opciones de menú correspondientes.

- Guardar (Save): guarda las modificaciones de un archivo original de configuración de menú sobrescribiéndolo.
- Guardar Como (Save As): guarda las modificaciones de un archivo original de configuración de menú en la ruta y archivo de destino que el usuario indique.

CONFIGURACIÓN DE BARRA DE HERRAMIENTAS

La barra de herramientas permite al usuario interactuar con el programa externo que se esté ejecutando en la consola. Para ello se podrán definir una serie de botones en los que se determinarán las funciones más prácticas y útiles que el usuario quiera tener siempre disponibles.

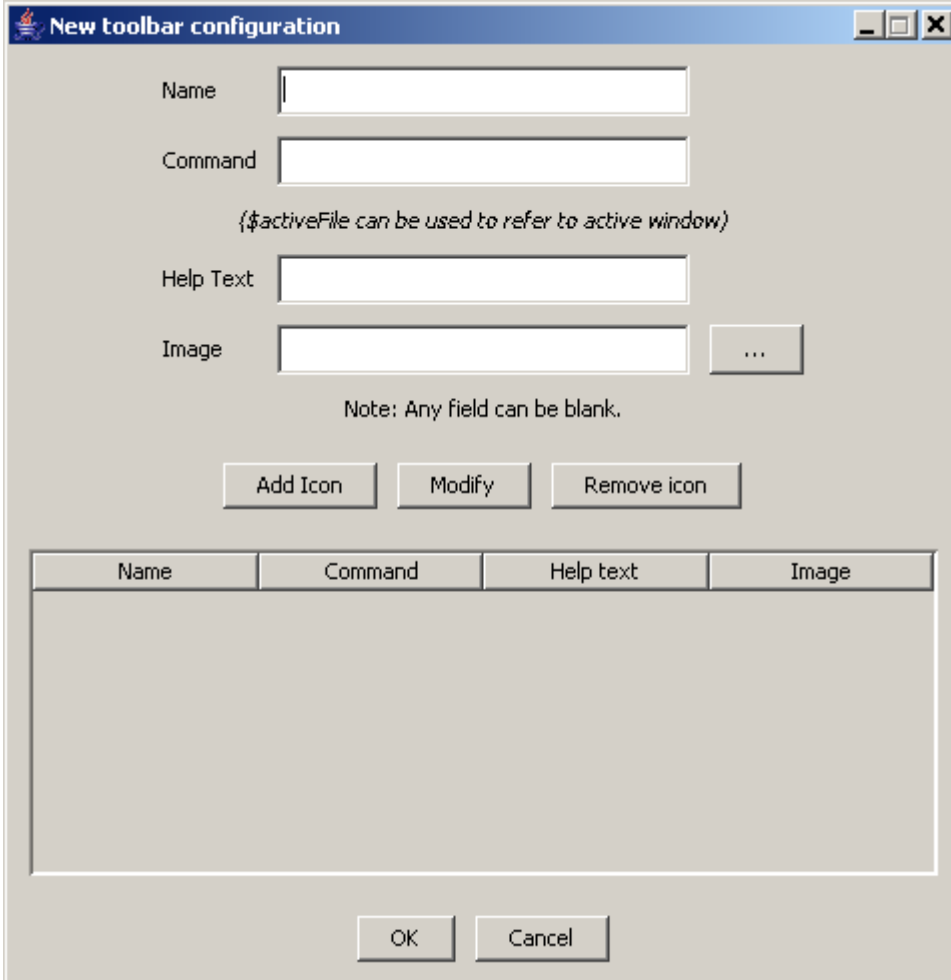
Inicialmente en la barra de herramientas podemos distinguir dos partes:

- Parte fija: corresponde al primer grupo de botones en el que tenemos una serie de opciones disponibles para poder manejar archivos y proyectos.
- Parte configurable: es el grupo de botones que se encuentran a la derecha del separador de la barra de herramientas y que son definidos completamente por el usuario.



Para el manejo de la barra de la barra de herramientas se proporcionan las siguientes opciones que se encuentran en el menú de Configuración → Barra de herramientas (Configuration → Tool Bar):

- Nuevo (New): al seleccionar la creación de una nueva barra de herramientas se lanzará una ventana para poder definir todos los aspectos de los botones que el usuario quiera tener en su barra de herramientas. El interfaz de configuración tiene un aspecto como el siguiente:



New toolbar configuration

Name

Command

(\$activeFile can be used to refer to active window)

Help Text

Image ...

Note: Any field can be blank.

Add Icon Modify Remove icon

Name	Command	Help text	Image
------	---------	-----------	-------

OK Cancel

En la parte superior de la ventana podemos configurar cada uno de los campos de que consta un botón de nuestra barra de tareas:

1. Nombre (Name): texto que mostrará el botón. En caso de existir un icono para este botón se mostrará el icono en lugar del nombre; y si dejamos este campo vacío el sistema le asignará automáticamente un número como nombre.
2. Comando (Command): es el comando que se ejecutará en el programa abierto en la consola en ese momento. Para este propósito podremos usar la variable de entorno \$activeFile que nos sirve para referirnos al archivo en foco en el momento de pulsar el botón.
3. Texto de ayuda (Help Text): texto que se muestra en un comentario emergente si dejamos el cursor posicionado sobre el botón.
4. Imagen (Image): si queremos que el botón muestre una imagen en lugar de el nombre que le hemos puesto podremos poner en este campo la ruta

de la imagen que deseemos. En este caso también podremos buscarla con el botón de apertura de archivos del diálogo del sistema que se encuentra situado a la derecha del campo de texto destinado a la ruta de la imagen.

En la parte inferior se muestra una tabla con todos los campos de cada uno de los botones definidos en la barra de herramientas hasta el momento. En esta tabla se puede seleccionar cualquiera de los botones ya configurados para que se muestren en la parte superior y poder modificarlos en caso de haber cometido algún error al definir alguna de sus características.

En el caso particular de querer modificar la configuración de un botón simplemente tendremos que seleccionarlo en la tabla inferior y se mostrarán sus características en los campos de texto de la parte superior de la ventana; editaremos las opciones que queramos modificar y pulsaremos sobre el botón Modificar (Modify) para que se apliquen los cambios a la tabla que tenemos abajo con los botones definidos.

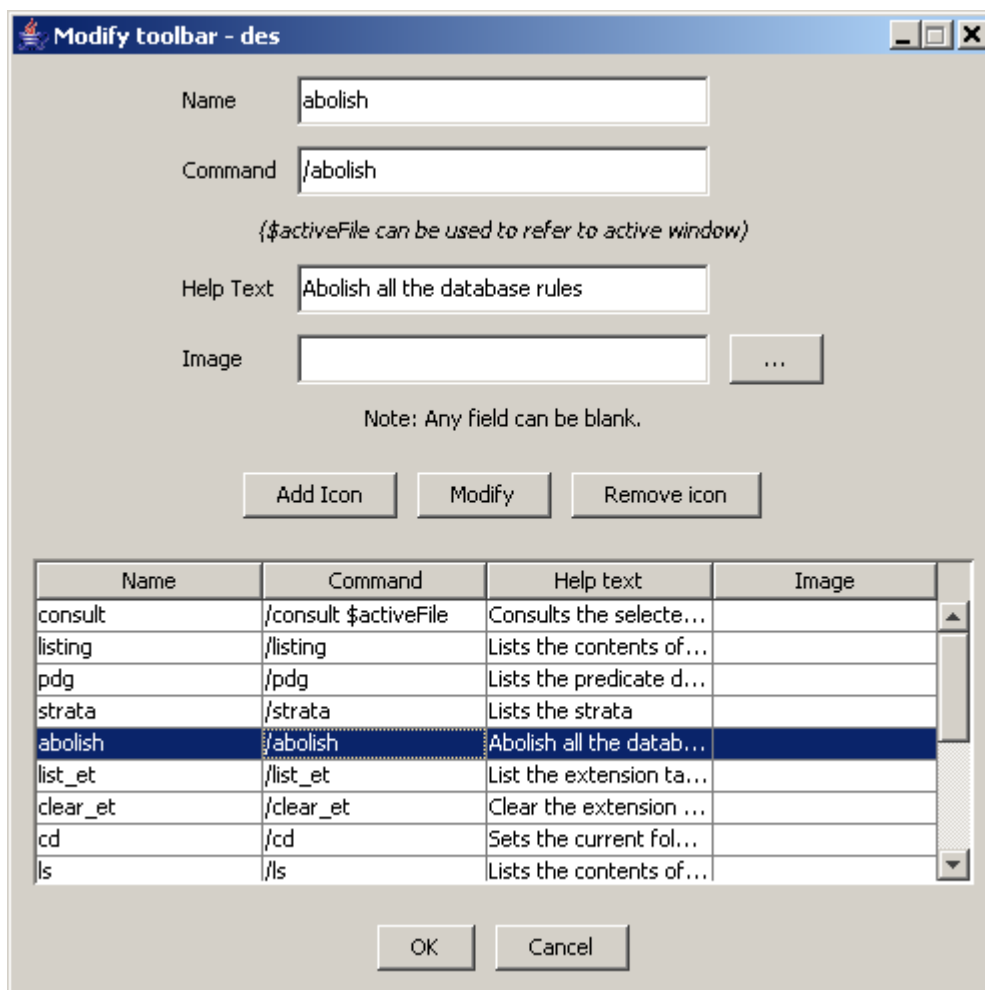
Además del botón Modificar tenemos otros dos botones para poder configurar nuestra barra de herramientas personalizada. El primero de ellos es Añadir Icono (Add Icon) que usaremos para añadir un botón nuevo con las características definidas en la parte superior a la lista de botones ya definida en la tabla inferior. Y el segundo es el botón Quitar Icono (Remove Icon) que elimina de la tabla el icono que esté seleccionado.

La configuración de la nueva barra de herramientas no quedará guardada hasta que el usuario no lo haga explícitamente con la opción de configuración destinada a tal efecto. Si en el momento de cerrar la aplicación la barra de herramientas no ha sido guardada, se perderá esa configuración y se recuperará la anterior configuración que estuviera guardada.

Los archivos de barra de herramientas tienen extensión BHcfg.

- Cargar (Load): esta opción permite cargar, a través del diálogo de sistema, una barra de herramientas previamente guardada por el usuario en un archivo de extensión BHcfg.
- Modificar (Modify): permite modificar la barra de herramientas actualmente en uso de forma que en el momento de abrirla se muestran los botones definidos para esa barra de herramientas así como el nombre de la misma en la barra de título de la ventana para saber qué configuración se está modificando en cada

momento. El aspecto, similar al de creación de nueva barra de herramientas, es el siguiente:



Cuando modificamos una barra de herramientas ocurre lo mismo que cuando creamos una barra de herramientas nueva: los cambios no se podrán recuperar una vez cerrada la aplicación a no ser que el usuario los haya guardado explícitamente en un archivo de configuración de barra de herramientas (BHcfg).

- Guardar (Save): permite guardar las modificaciones hechas sobre una configuración original. En el momento de guardar se sobrescribe el archivo original por uno nuevo con el mismo nombre y con los cambios definidos por el usuario.
- Guardar Como (Save As): permite guardar una configuración nueva o las modificaciones hechas sobre una configuración original. En este caso podremos

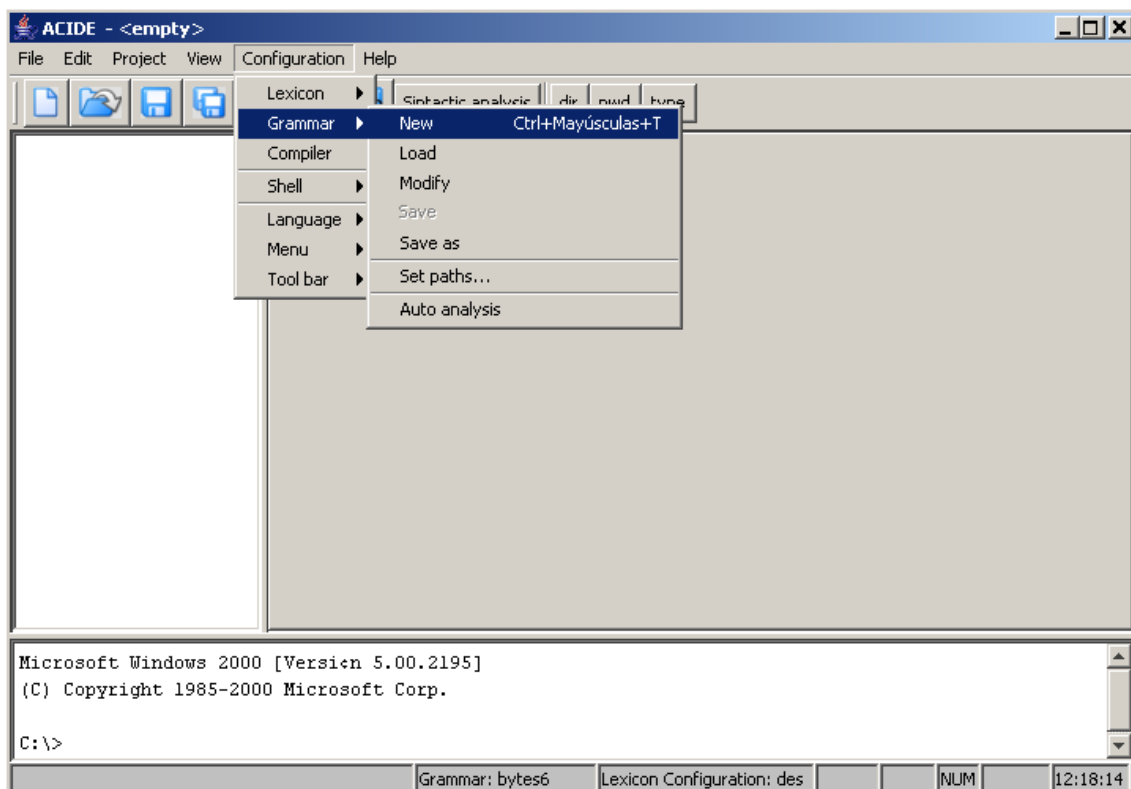
elegir una ruta y nombre nuevos para nuestra configuración de barra de herramientas.

CONFIGURACIÓN SINTÁCTICA

Una de las principales características de la aplicación es poder configurar la gramática que reconoce para poder detectar los errores sintácticos cometidos al escribir el código. Para ello es necesario insertar en las opciones de configuración una gramática escrita en sintaxis EBNF (Extended BNF) que ACIDE transformará automáticamente en otros ficheros para poder usarlos como analizadores léxicos y sintácticos.

Para poder generar nuevas gramáticas, modificar las ya existentes y cargar nuevas gramáticas previamente creadas es necesario que se cumplan una serie de requisitos descritos en el apartado de la memoria correspondiente.

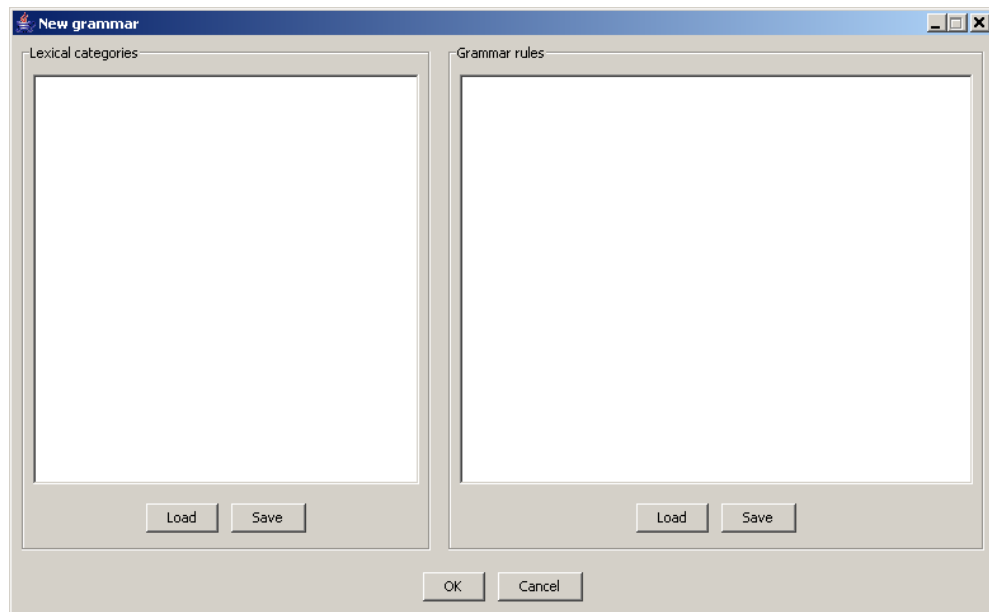
El menú de configuración de gramáticas tiene la siguiente apariencia:



Las opciones que muestra tienes las siguientes funcionalidades:

- Nuevo (New): permite crear nuevas gramáticas a partir de unas categorías léxicas y unas reglas sintácticas escritas en sintaxis EBNF e introducidas en

los cuadros correspondientes de la siguiente ventana, que es la que se lanza al pulsar sobre esta opción de menú:



Como se puede observar en cada uno de los títulos de los cuadros de texto, el de la izquierda está destinado a las categorías léxicas de nuestra nueva gramática y el de la derecha es el correspondiente a las reglas gramaticales.

En ambos casos se dispone de botones de cargado y guardado (Load, Save) de cada uno de los aspectos necesarios para la creación de la gramática. En este caso se trabaja sobre archivos de texto plano con extensión txt.

En cuanto el usuario pulse el botón Aceptar (Ok) ACIDE realizará una serie de operaciones que culminarán en la creación de un archivo comprimido y autoejecutable de Java (jar) que el entorno usará para hacer el análisis del código escrito. Estas operaciones requieren algo de tiempo hasta que se generan todos los archivos y se comprimen adecuadamente, así que por unos momentos no se podrá usar el entorno. Una vez terminadas tendremos nuestro nuevo archivo de gramática disponible y conectado con el programa.

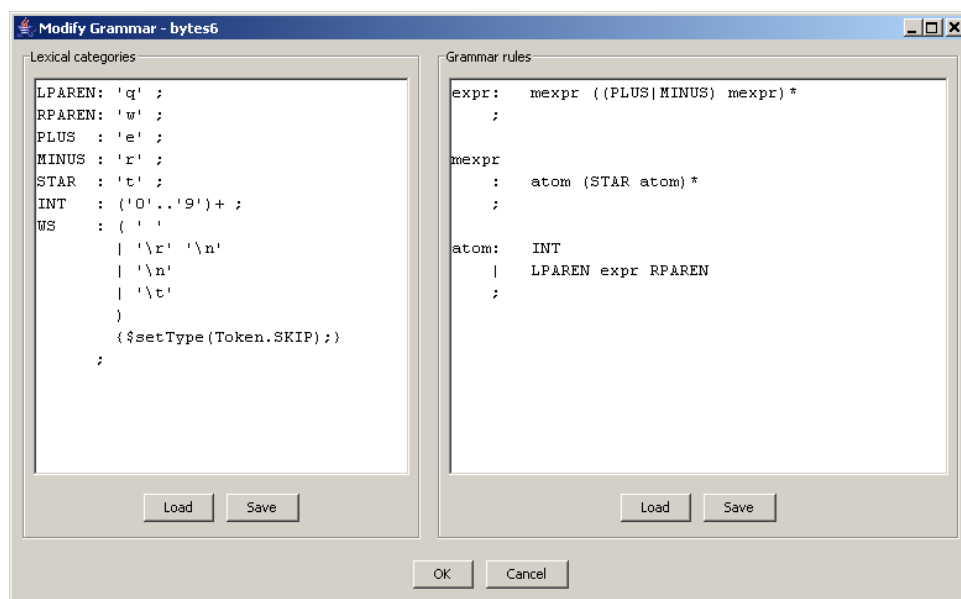
En el momento de crear una nueva gramática no se guarda hasta que el usuario no lo indique explícitamente con las opciones de menú destinadas a

ello. En caso de cerrar la aplicación sin guardar estos cambios se recuperará la última configuración sintáctica guardada.

Los archivos de configuración sintáctica tienen extensión jar.

- Cargar (Load): esta opción permite al usuario cargar, desde el diálogo de sistema, una configuración sintáctica previamente guardada.
- Modificar (Modify): para modificar una gramática los pasos a seguir son los mismos que para crear una nueva. Las dos diferencias con la creación de una nueva gramática son que en las ventanas de texto para introducir las categorías léxicas y las reglas gramaticales podremos ver las de la gramática actual para poder modificarlas directamente y que en la barra de título de la ventana aparece el nombre de la gramática que se está modificando.

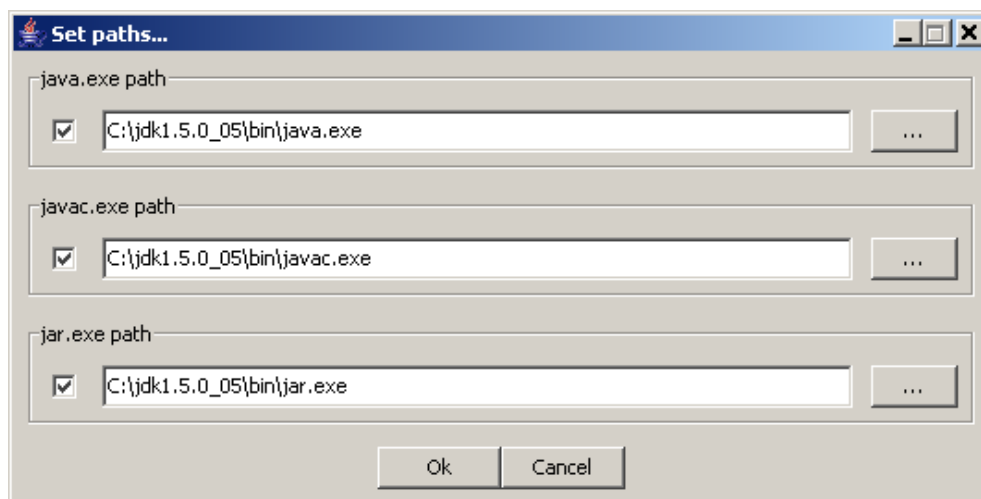
La ventana de modificación tiene la siguiente apariencia:



- Guardar (Save): permite guardar las modificaciones realizadas sobre una configuración sintáctica original sobrescribiendo el archivo en el que estaba guardada.
- Guardar Como (Save As): permite guardar las nuevas configuraciones sintácticas y las modificaciones hechas sobre una configuración original en la ruta y archivo de destino que el usuario desee.
- Configurar Rutas (Set Paths): para la creación, modificación y manejo de gramáticas es necesario tener instaladas una serie de herramientas de Java

explicadas en la sección de requisitos del sistema. Además de tenerlas instaladas tenemos que configurar sus rutas para que ACIDE pueda acceder y trabajar con ellas. En caso de no tener bien configuradas las rutas no será posible crear, modificar o manejar las gramáticas.

La ventana que se muestra cuando se van a configurar las gramáticas es la siguiente:

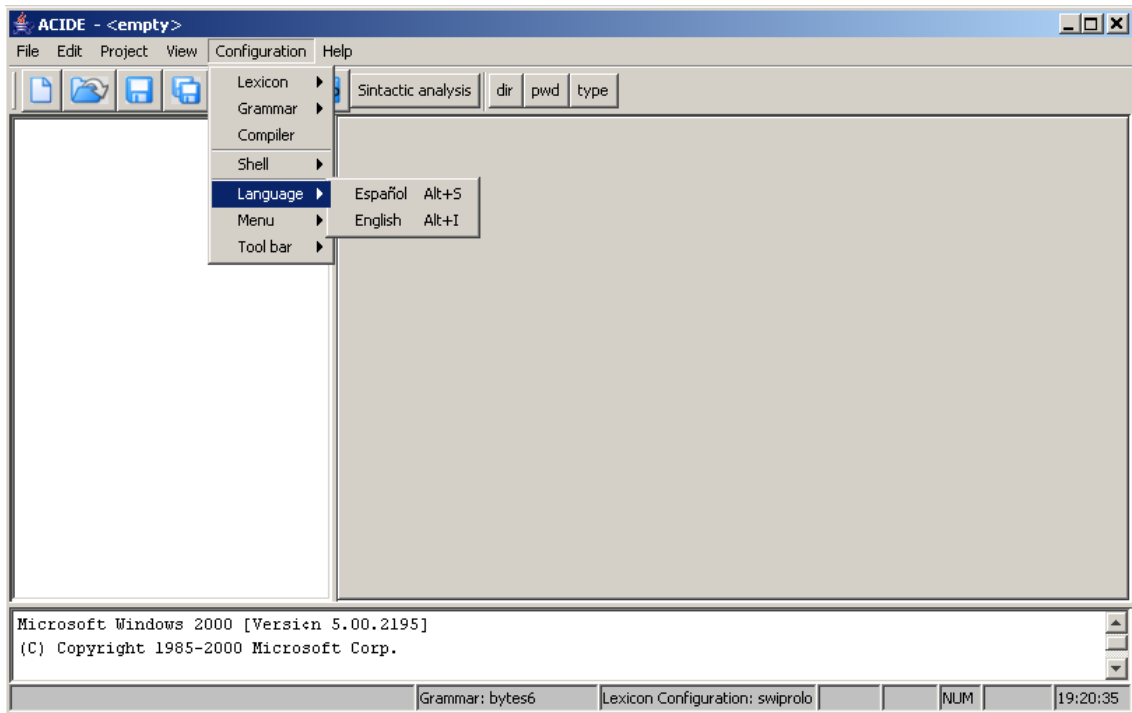


En cada cuadro de texto hay que introducir las rutas de las herramientas necesarias. Junto a cada cuadro de texto tenemos otros dos elementos:

1. Cuadro de validación: si lo tenemos seleccionado se usará la ruta insertada en el cuadro de texto; y en caso de no seleccionarlo se usará la ruta de esa herramienta que esté insertada en el CLASSPATH del sistema operativo.
 2. Botón de exploración: en caso de no conocer la ruta de la herramienta podemos buscarla en el sistema de carpetas por medio de este botón que lanza el diálogo de sistema.
- Auto análisis (Auto analysis): el análisis sintáctico puede ser automático o bien se puede realizar solamente cuando el usuario considere oportuno. En caso de tener el análisis automático activado el botón de la barra de herramientas que realiza el análisis sintáctico desaparecerá; y volverá a aparecer de nuevo si se desea que el análisis no se automatice.

CONFIGURACIÓN DE IDIOMA

Una de las características de nuestra aplicación es la funcionalidad de cambiar de idioma en tiempo de ejecución. La elección del idioma se hace desde el menú Configuration → Lenguaje.



Para poder hacer esto hemos tenido que almacenar todas las cadenas de texto que utilizamos en la aplicación en archivos de recursos de texto plano.

```
PideBundle_es.properties x
# este es el archivo PideBundle.properties por defecto
s1 = Archivo
s2 = Edición
s3 = Proyecto
s4 = Ver
s5 = Configuración
s6 = Idioma
s7 = Ayuda
s8 = Nuevo
s9 = Abrir
s10 = Guardar como...
s11 = Español
s12 = English
s13 = Salir
s14 = Nuevo proyecto
s15 = Abrir proyecto
s16 = Guardar proyecto
s17 = Añadir archivo
s18 = Compilar proyecto
s19 = Ejecutar proyecto
s20 = Depurar
s21 = Deshacer
s22 = Repetir
s23 = Copiar
s24 = Cortar
s25 = Pegar
s26 = Buscar
s27 = Reemplazar
s28 = Mostrar Log
```

Como podemos ver en la captura anterior todas las cadenas de texto están asociadas a variables que utilizará internamente ACIDE.

La gran ventaja de utilizar estos archivos de texto plano es que el usuario podrá modificar cualquiera de estas cadenas o incluso podrá pasar el archivo completo por un programa de traducción generando automáticamente un nuevo archivo de idioma.

CONFIGURACIÓN LÉXICA

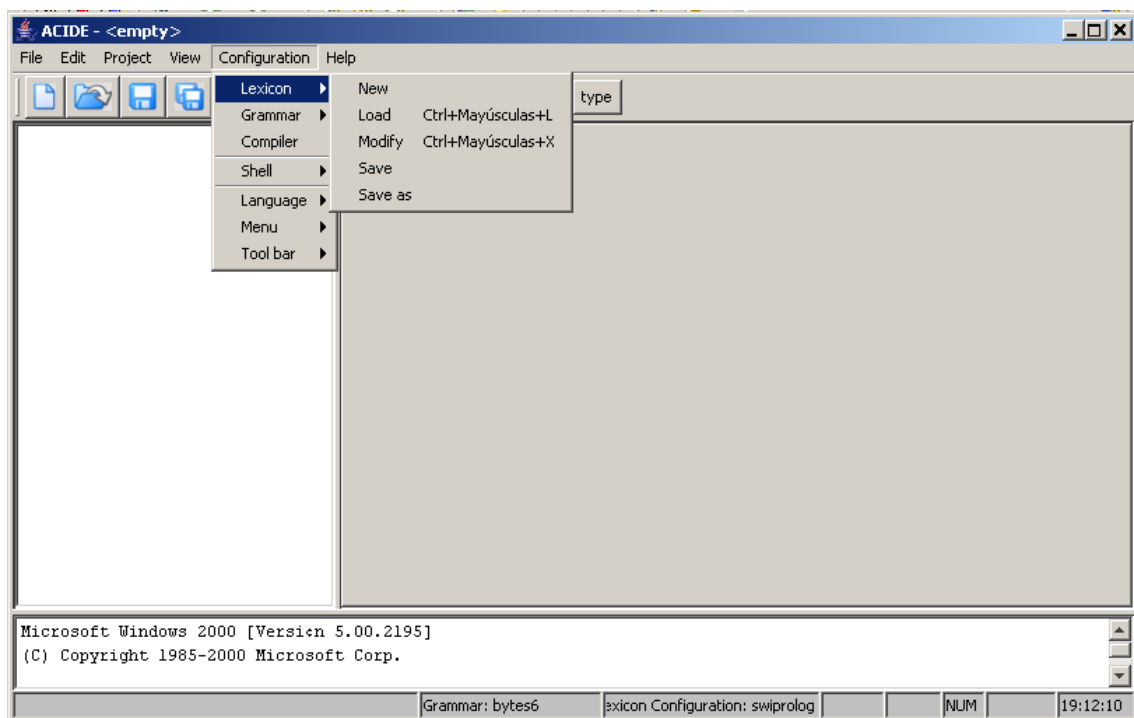
Las configuraciones léxicas son necesarias para poder realizar el análisis léxico en el editor de texto.

Toda configuración léxica tendrá una lista de palabras reservadas, una lista de delimitadores y un símbolo de comentario de línea.

A su vez cada palabra reservada y cada delimitador tendrá asociados un color, un estilo como pueden ser cursiva o negrita y una opción que permitirá hacer que el reconocimiento léxico sea sensible a mayúsculas y minúsculas.

Respecto al símbolo de comentario podremos asociar un color que se aplicará sobre las líneas de texto comentadas.

Desde el menú Configuration → Lexicon, podremos realizar las operaciones básicas sobre configuraciones de lenguaje que son : Nueva (New) , Cargar (Load), Modificar (Modify), Guardar (Save), Guardar Como (Save As).

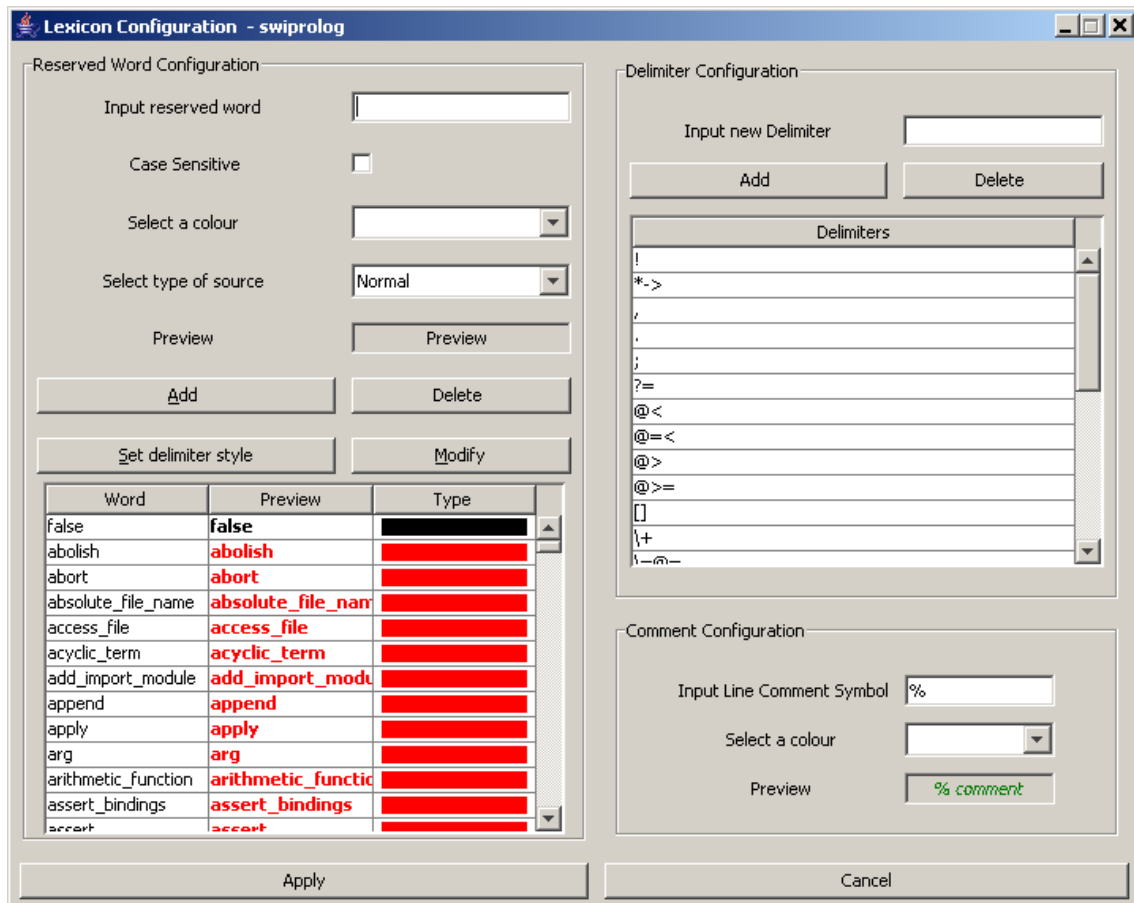


La opción de menú “Nueva” creará una nueva configuración de lenguaje con un nombre dado.

La opción “Cargar” nos permitirá carga archivos de lenguaje que hayamos guardado previamente. Estos archivos tendrán la extensión XML.

Las opciones de “Guardar” y “Guardar Como” nos permitirán almacenar los cambios realizados sobre un lenguaje con el nombre guardado o con un nuevo nombre respectivamente.

La opción “Modificar” abrirá una nueva pantalla en la que podremos definir todas las partes configurables del lenguaje.



En esta pantalla encontramos tres grandes zonas.

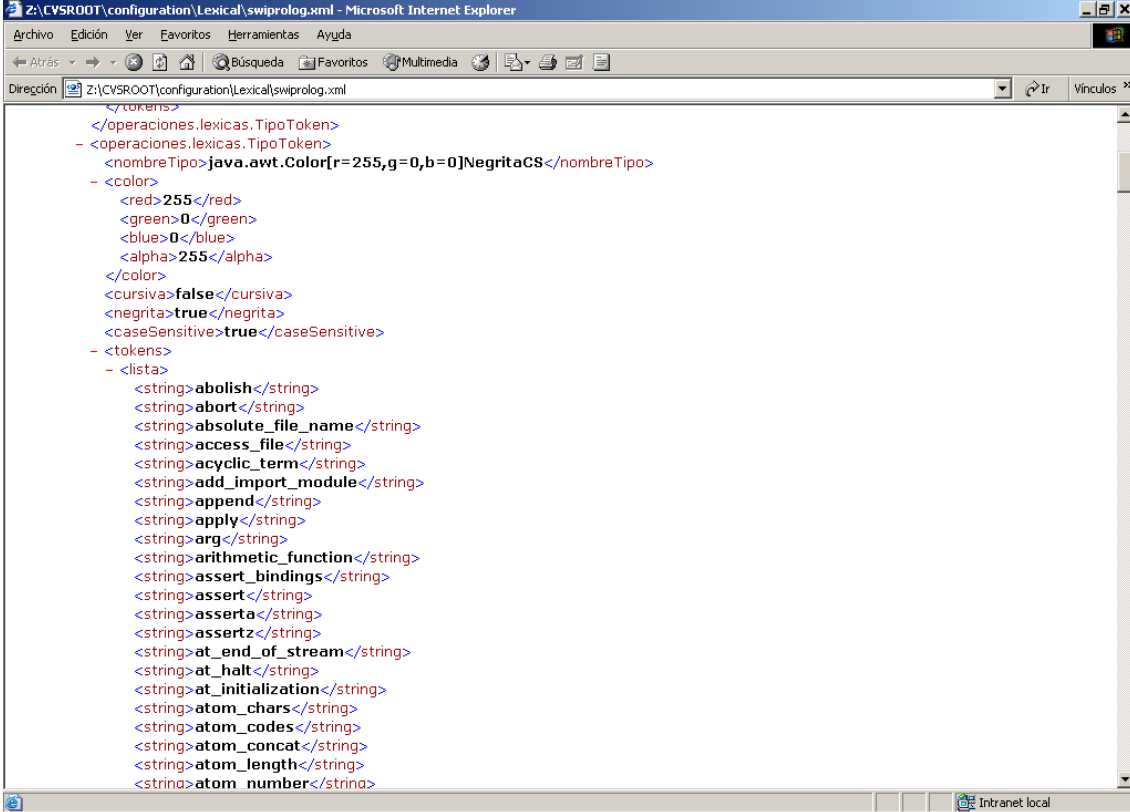
La primera estará asociada a las palabras reservadas. Aquí podremos añadir, eliminar y modificar palabras reservadas de nuestra configuración léxica. Mostraremos la información en tablas que se podrán ordenar alfabéticamente, por tipos y por colores.

Otra zona estará asociada a los delimitadores. En este caso mostraremos una lista ordenada con los delimitares almacenados y tendremos la posibilidad de eliminar o añadir nuevos delimitadores.

La tercera división de esta pantalla esta reservada para la configuración de los comentarios de línea. En este caso podremos definir un símbolo de comentario de línea y asociarle un color que aplicaremos sobre las líneas comentadas en los editores. También se mostrará una vista previa de una línea comentada.

Otra funcionalidad importante de esta pantalla es aplicar un estilo a todos los delimitadores, de modo que la lista de delimitadores también pasará formar parte de la lista de palabras reservadas.

Al crear una nueva configuración de lenguaje nuestra aplicación genera automáticamente archivos en formato XML que contendrán toda la información asociada al lenguaje. A continuación mostramos una captura de pantalla de uno de estos archivos.



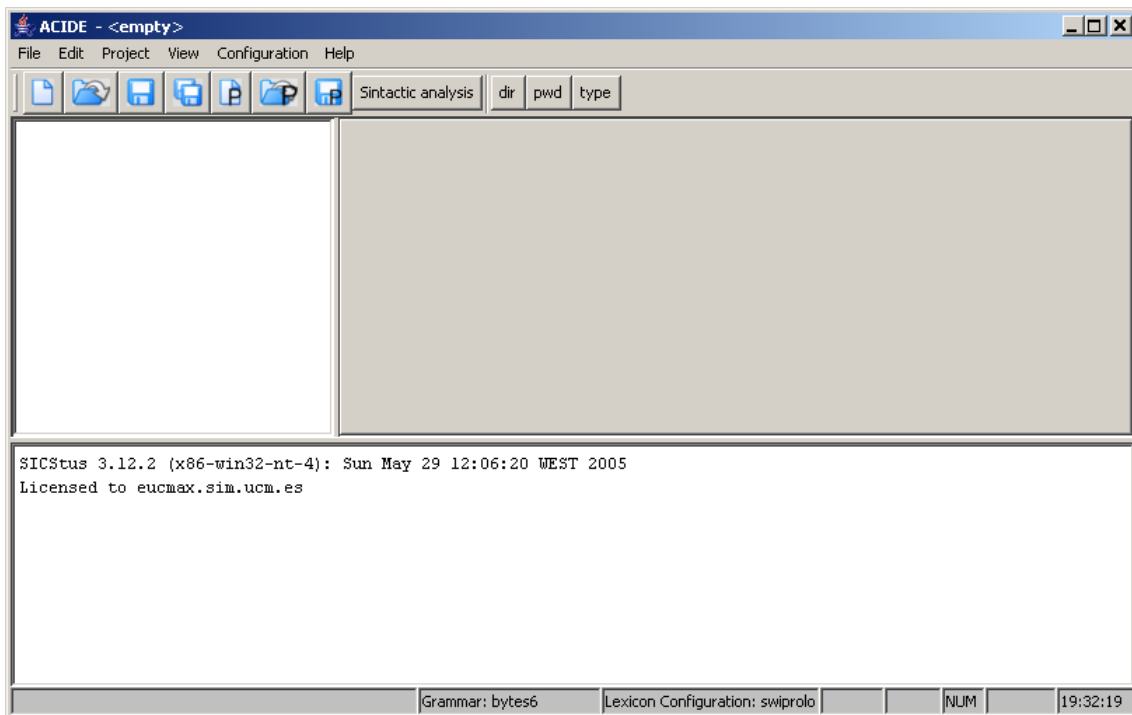
```
</tokens>
</operaciones.lexicas.TipoToken>
- <operaciones.lexicas.TipoToken>
  <nombreTipo>java.awt.Color[r=255,g=0,b=0]NegritaCS</nombreTipo>
  - <color>
    <red>255</red>
    <green>0</green>
    <blue>0</blue>
    <alpha>255</alpha>
  </color>
  <cursiva>false</cursiva>
  <negrita>true</negrita>
  <caseSensitive>true</caseSensitive>
- <tokens>
  - <lista>
    <string>abolish</string>
    <string>abort</string>
    <string>absolute_file_name</string>
    <string>access_file</string>
    <string>acyclic_term</string>
    <string>add_import_module</string>
    <string>append</string>
    <string>apply</string>
    <string>arg</string>
    <string>arithmetic_function</string>
    <string>assert_bindings</string>
    <string>assert</string>
    <string>asserta</string>
    <string>assertz</string>
    <string>at_end_of_stream</string>
    <string>at_halt</string>
    <string>at_initialization</string>
    <string>atom_chars</string>
    <string>atom_codes</string>
    <string>atom_concat</string>
    <string>atom_length</string>
    <string>atom_number</string>
```

Estos archivos se pueden modificar por el usuario sin necesidad de tener abierta la aplicación.

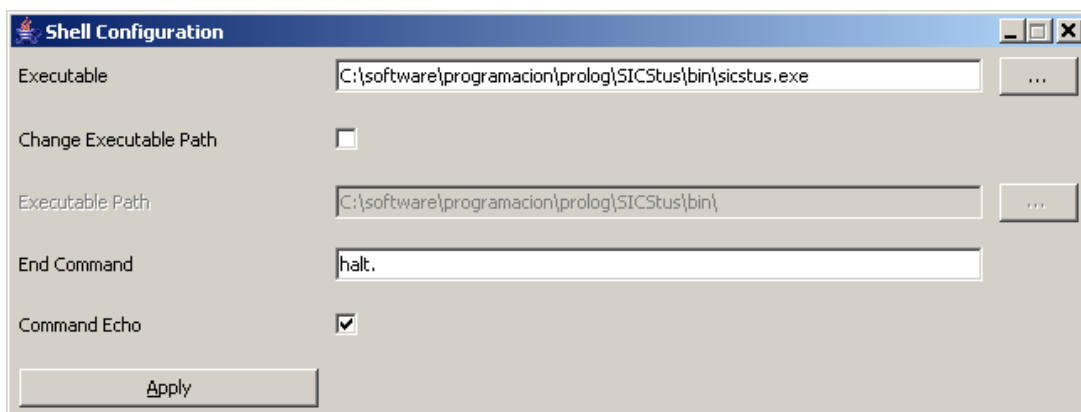
CONFIGURACIÓN DE LA CONSOLA

Una parte de la aplicación será la consola que nos permitirá interactuar con otros programas como compiladores o interpretes.

La ventana llamada consola se visualizará en la parte inferior de la aplicación y será una ventana de entrada/salida, para poder interactuar con el usuario.



La configuración de la consola se realizará desde el menú Configuration → Shell, y nos mostrará una ventana como la que presentamos a continuación.



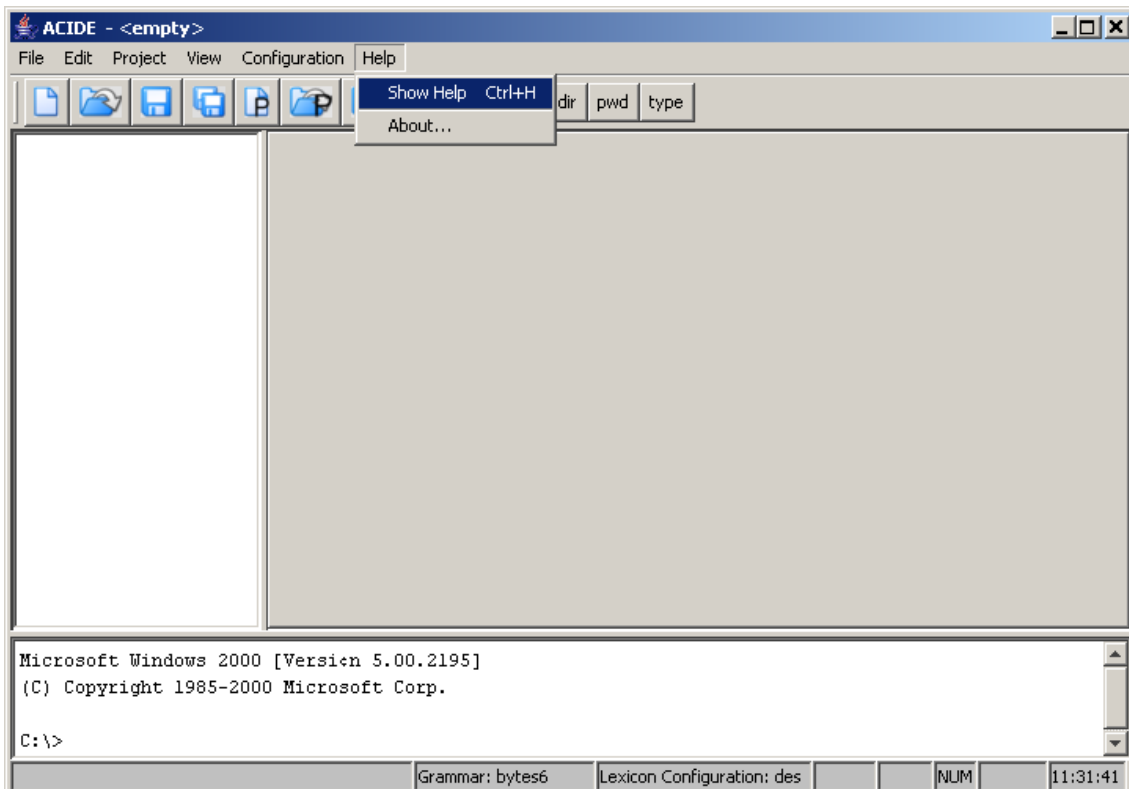
En esta pantalla tendremos que configurar una ruta donde se encontrará el ejecutable, un directorio de ejecución en caso de que sea distinto al directorio del ejecutable, así como un comando de salida para poder cerrar el flujo de datos.

La consola nos permitirá interactuar directamente con otros programas a través de la ventana asociada, pero también podremos realizar operaciones mediante las opciones de menú, como por ejemplo la opción de Ejecutar del menú Proyecto. (El resultado de la ejecución se mostrará en la ventana consola).

Otra posibilidad que tiene el usuario para interactuar con la consola es automatizar comandos definiendo nuevos botones en la barra de herramientas. Recomendamos ver la sección Configuración de barra de herramientas para más información.

MENÚ AYUDA

La última opción del menú es la ayuda (Help) del entorno, que consta básicamente de dos apartados como se muestra a continuación:



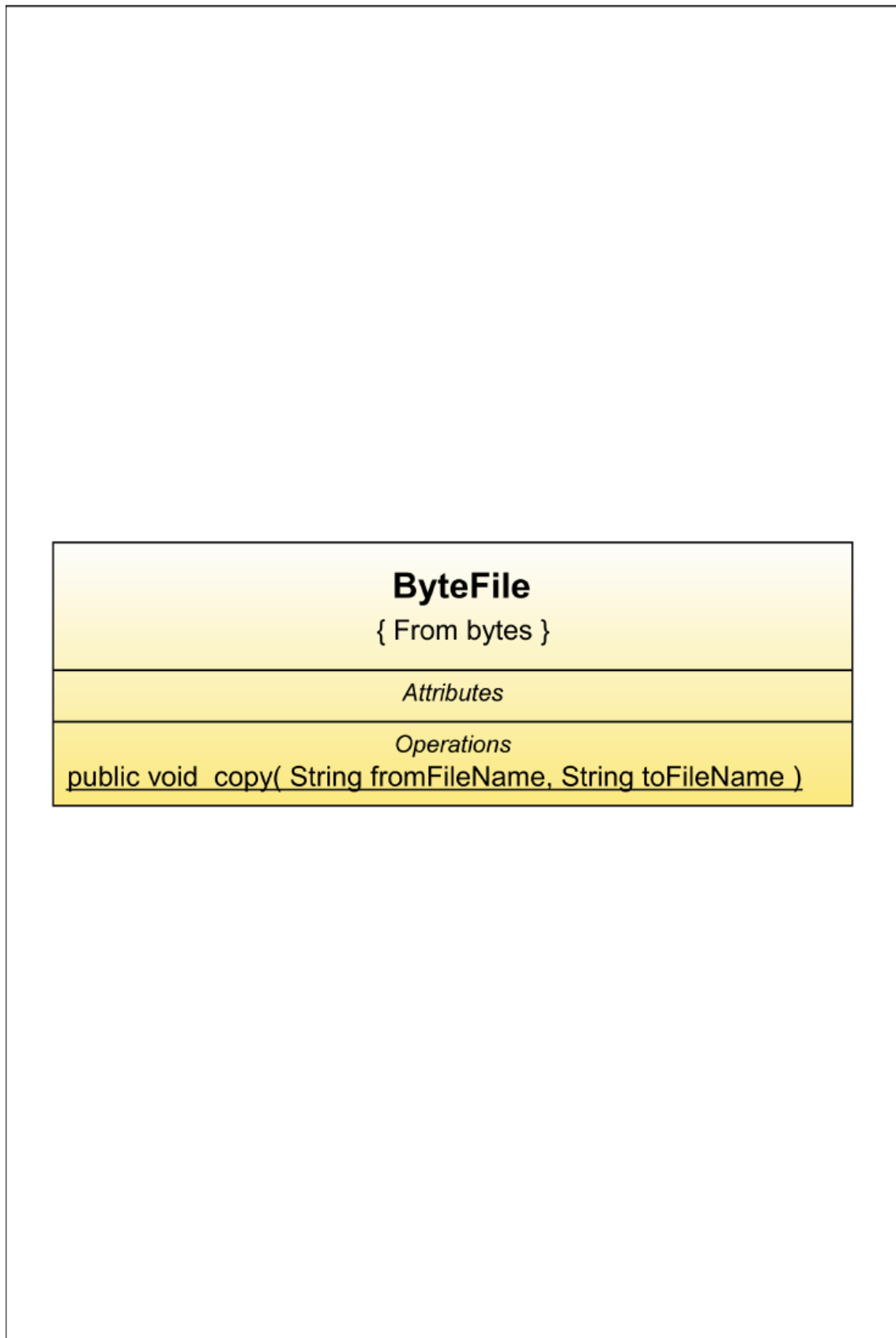
La primera opción de la lista es Mostrar ayuda (Show Help) que directamente enlaza con este archivo de manual de usuario para cualquier consulta que sea necesaria.

La segunda y última opción disponible es Acerca de... (About...) que simplemente muestra una ventana con las características técnicas del programa, su versión, la página de internet a la que poder acudir para remitir cualquier pregunta o sugerencia y el nombre de las personas que lo han llevado a cabo.

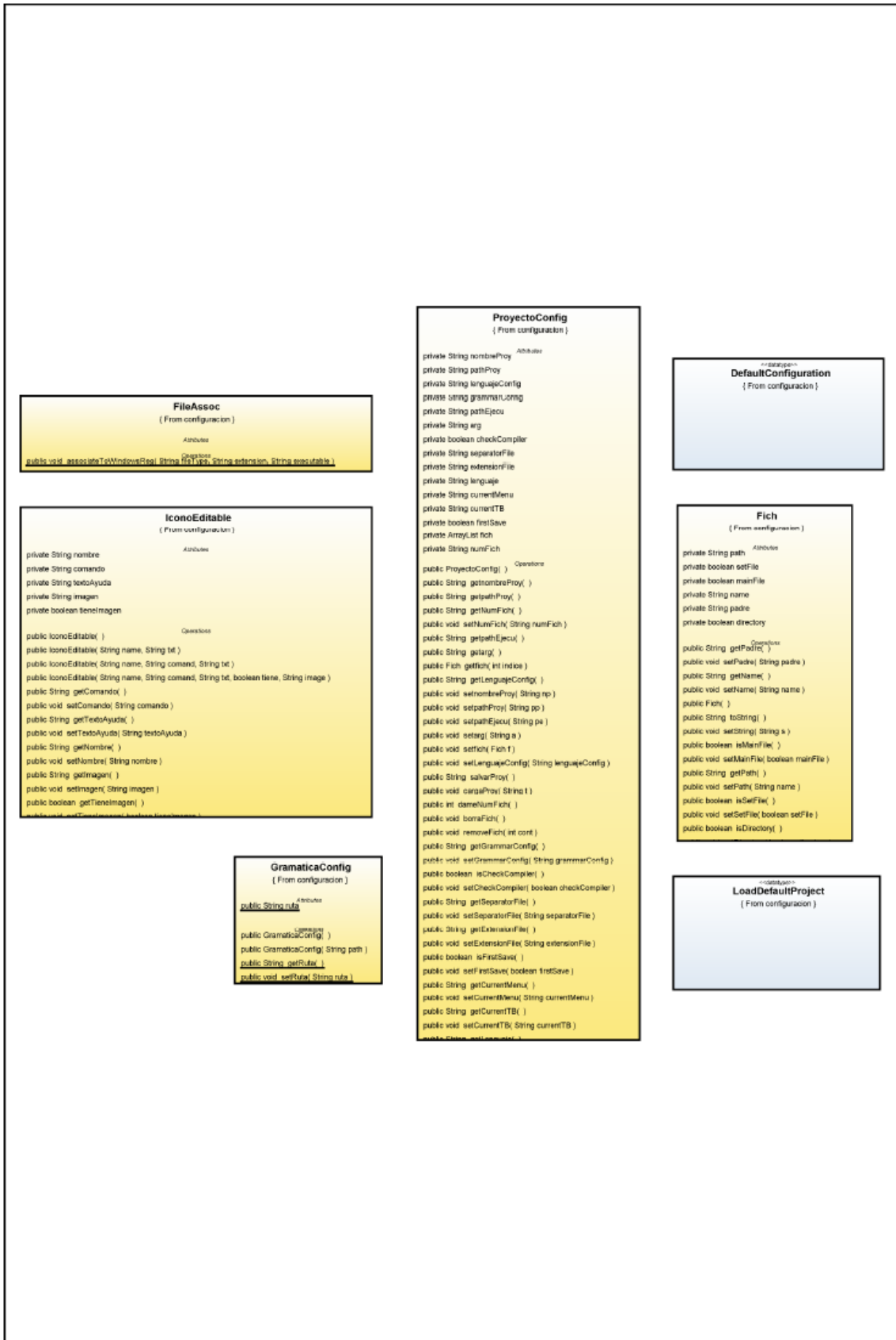
ACIDE

DIAGRAMAS UML

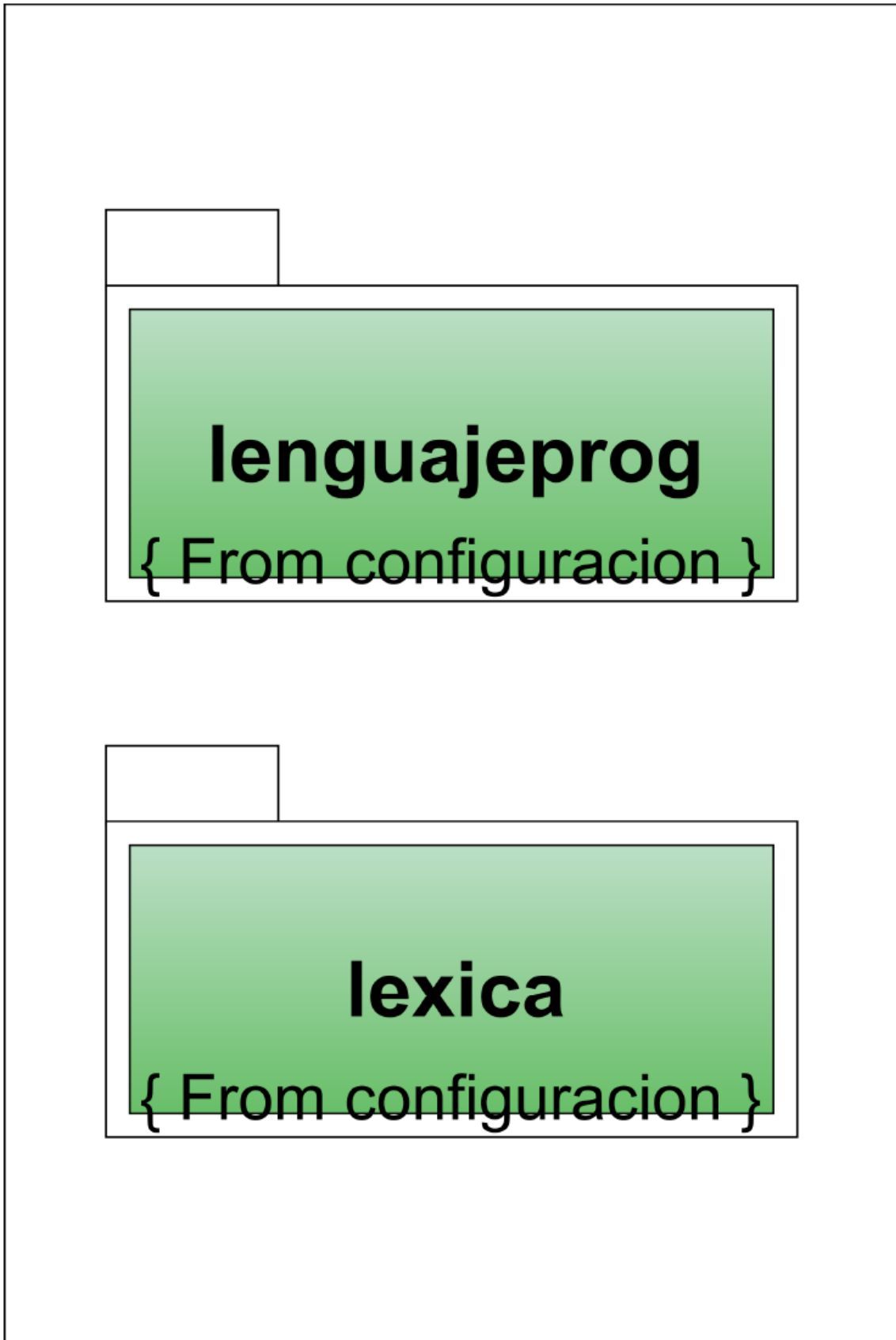
Versión: 5.0
Fecha: 05/07/2007



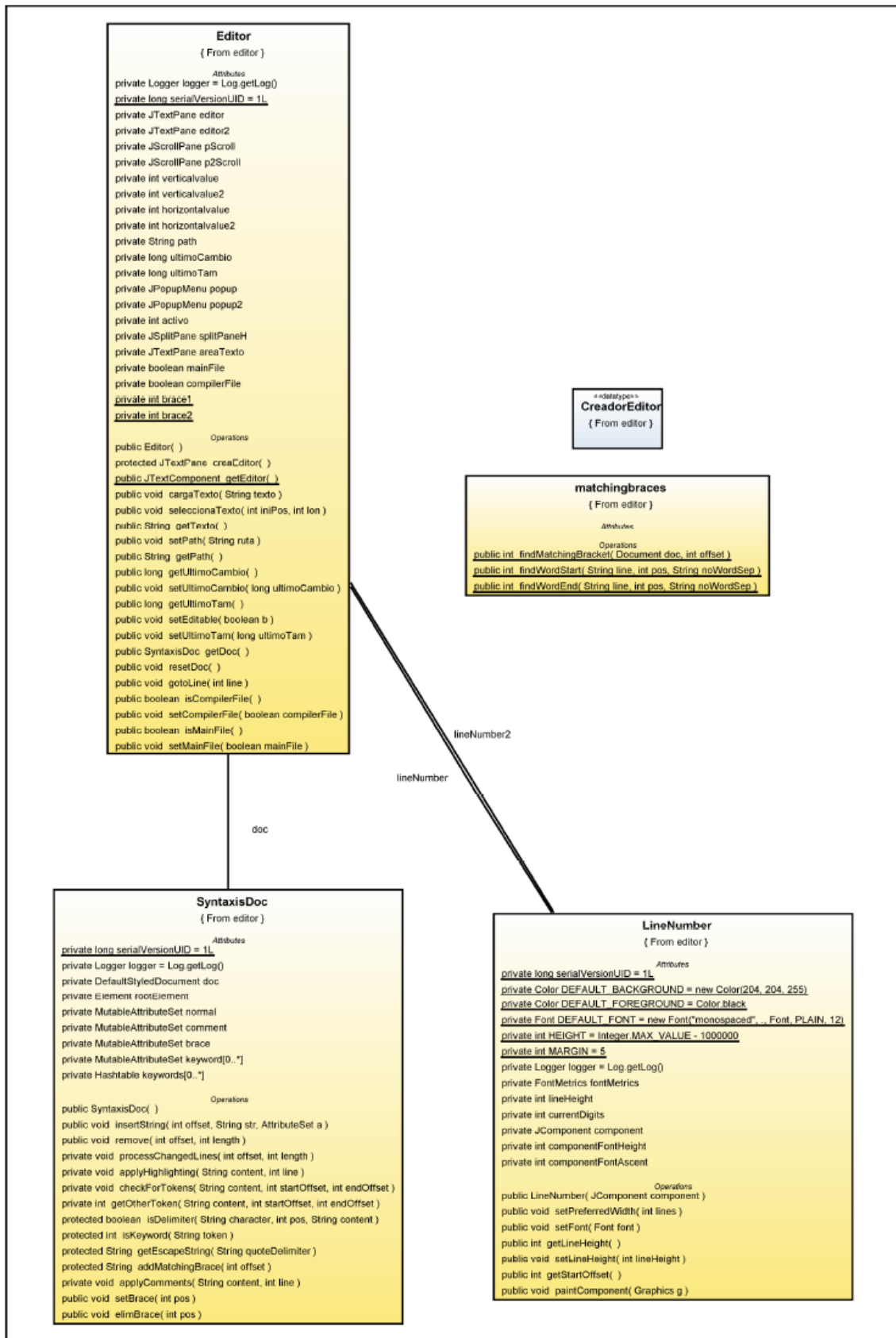
bytes Diagram



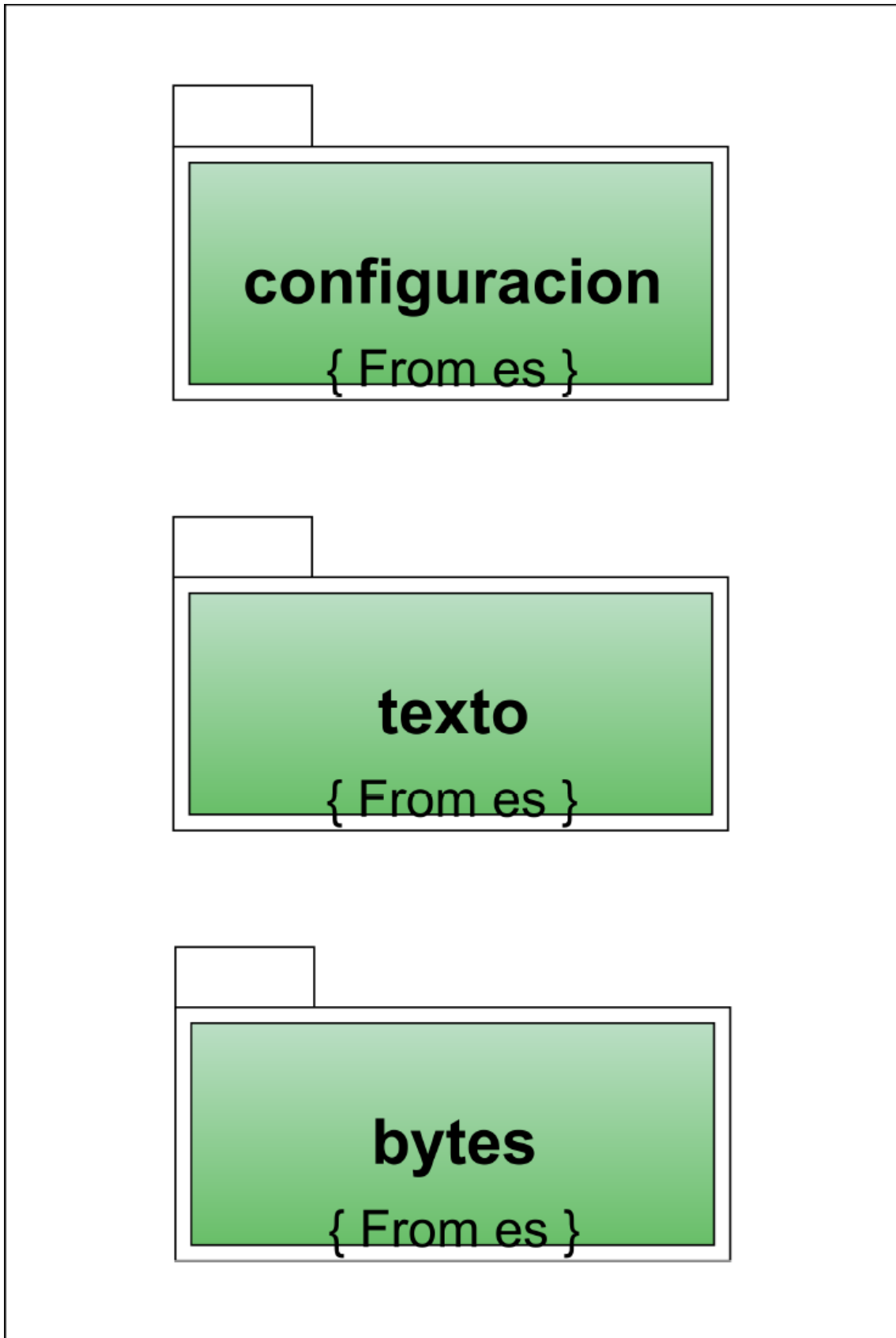
configuracion_Op Diagram



configuration Diagram



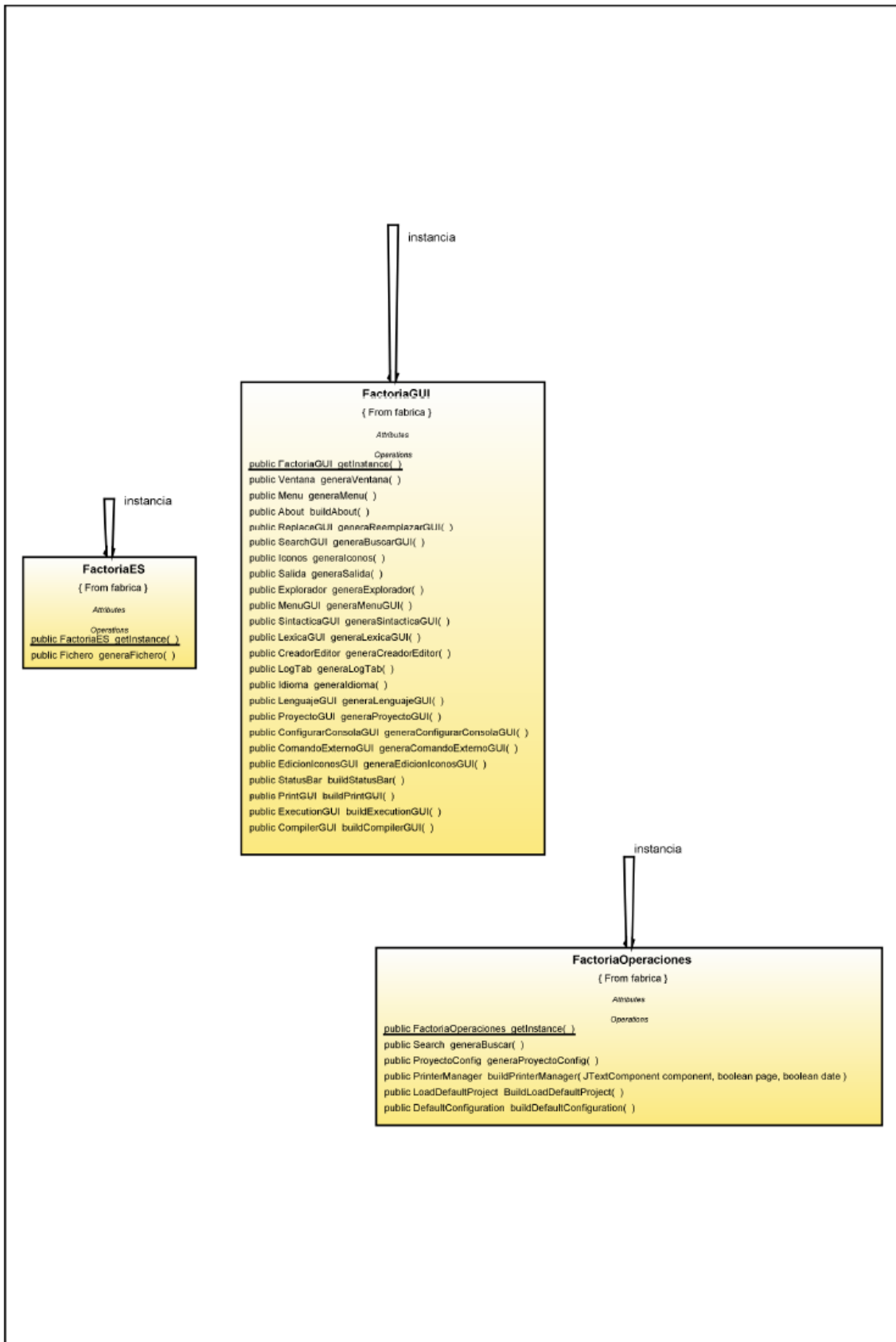
editor Diagram



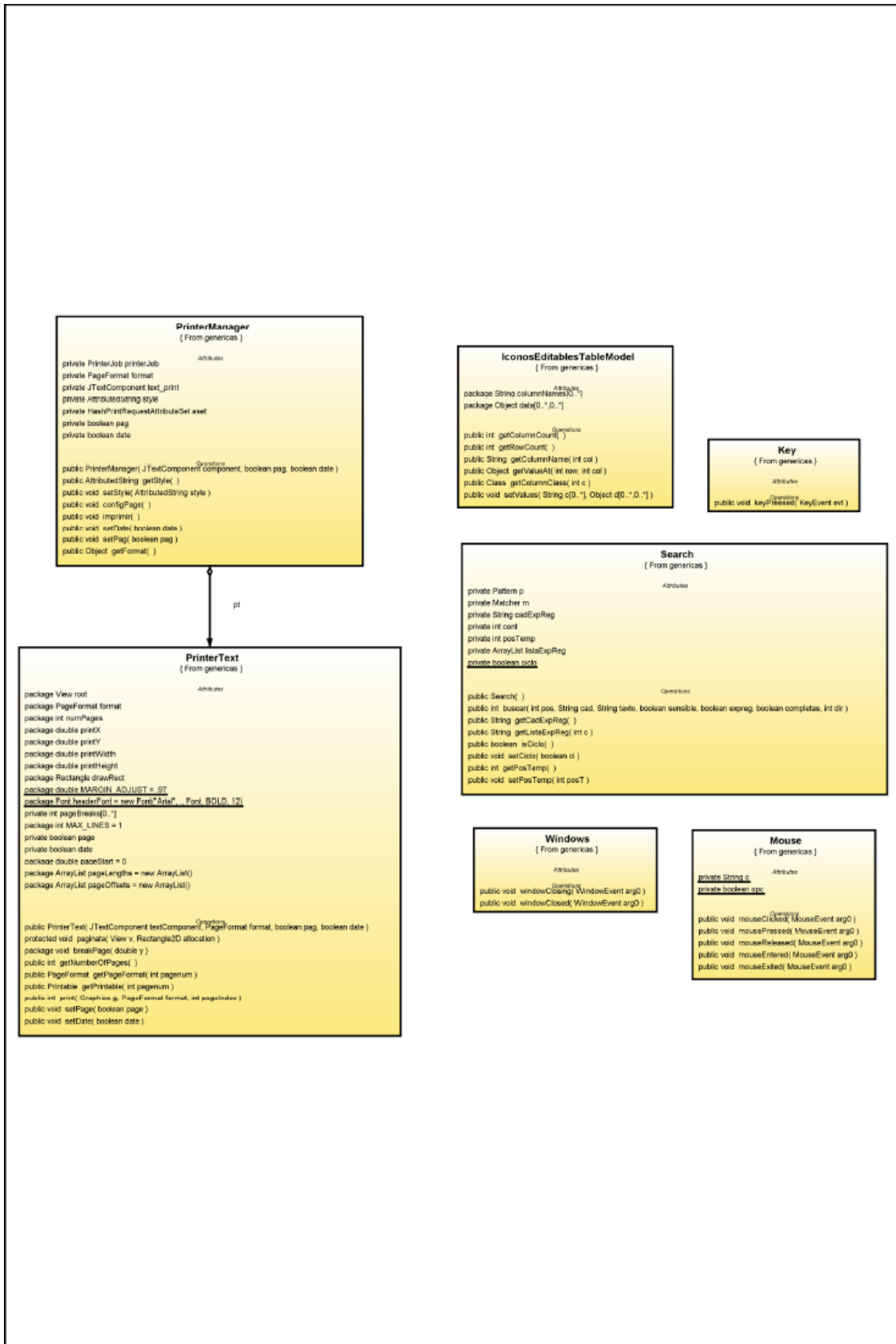
es Diagram



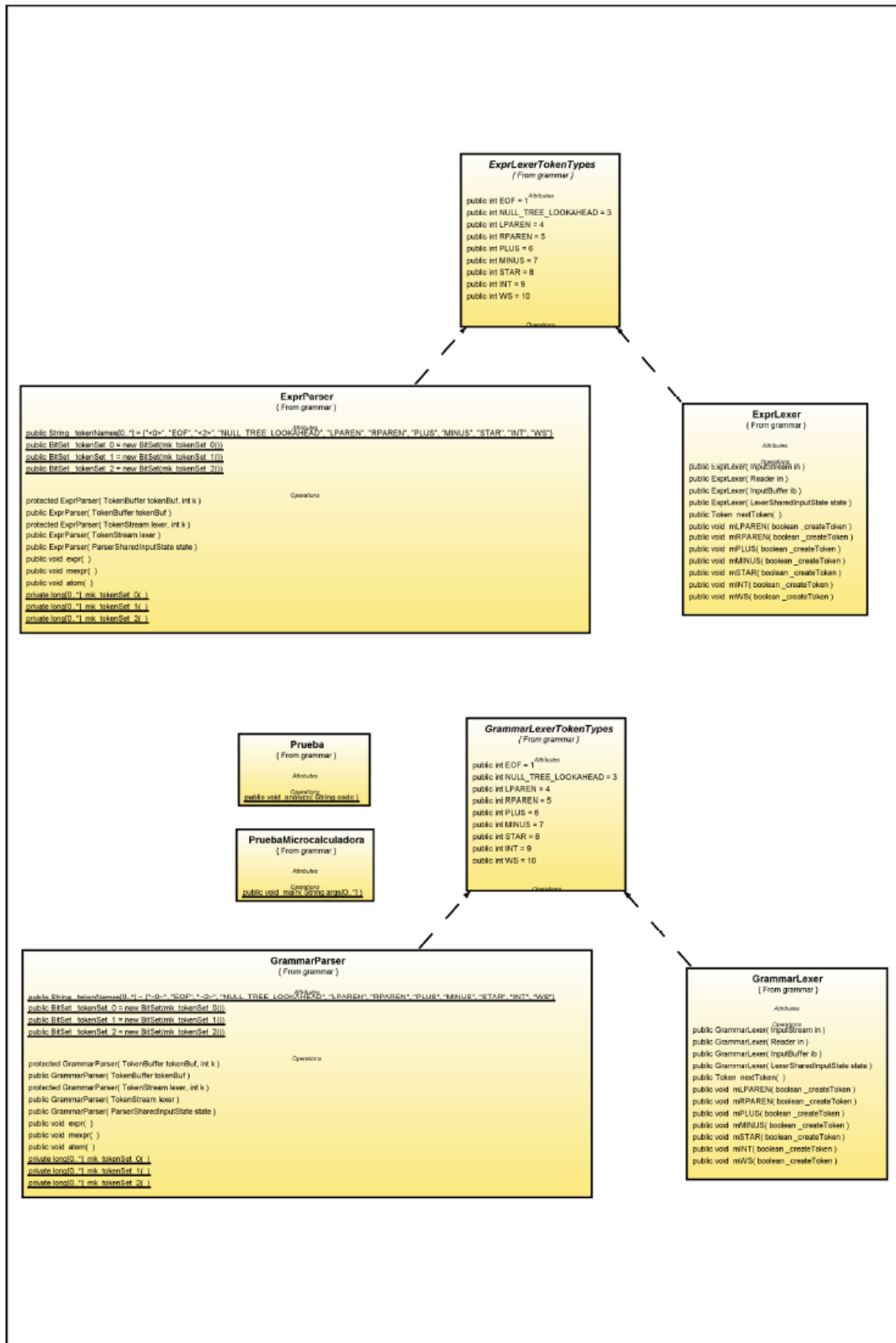
explorador Diagram



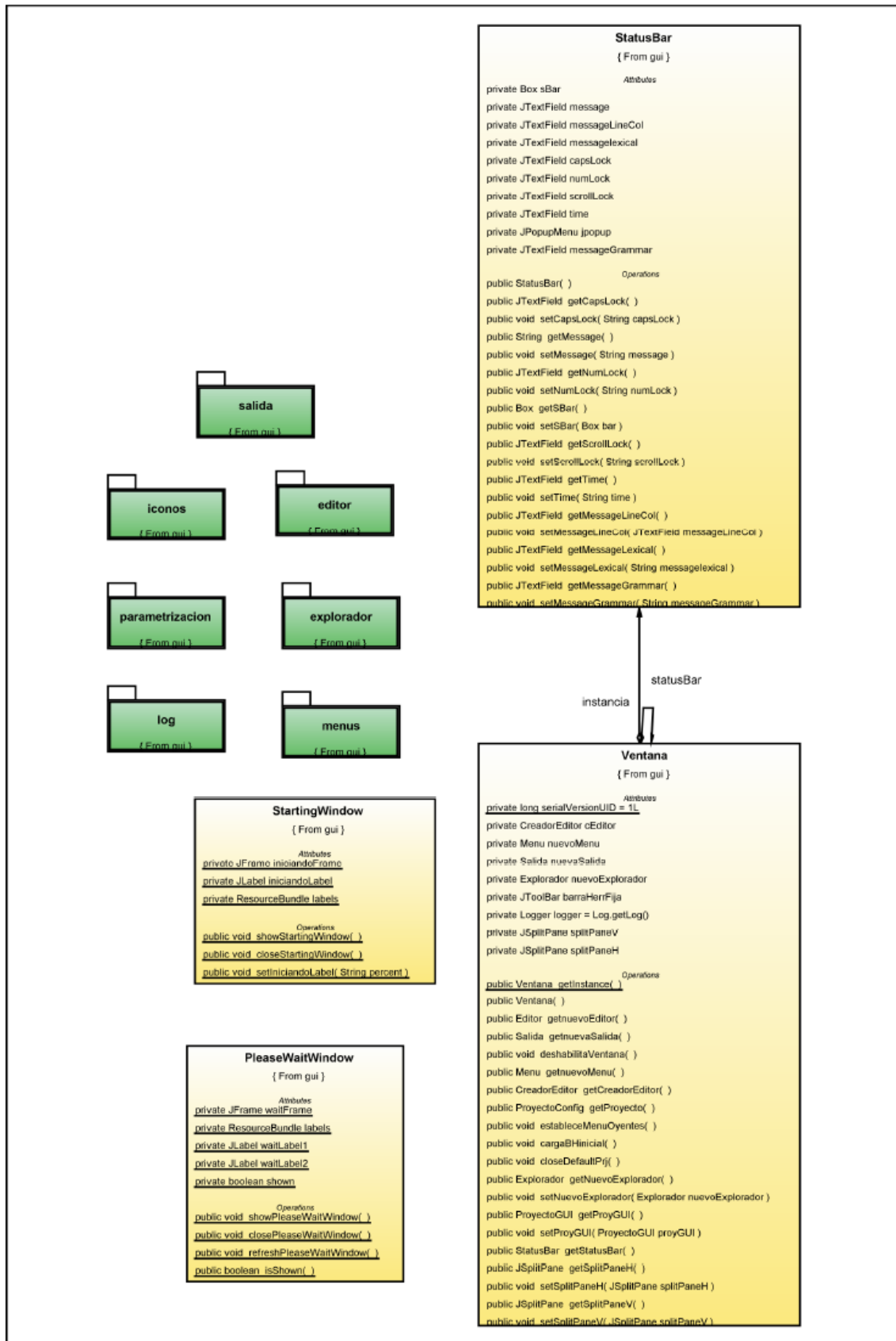
fabrica Diagram



genericas Diagram



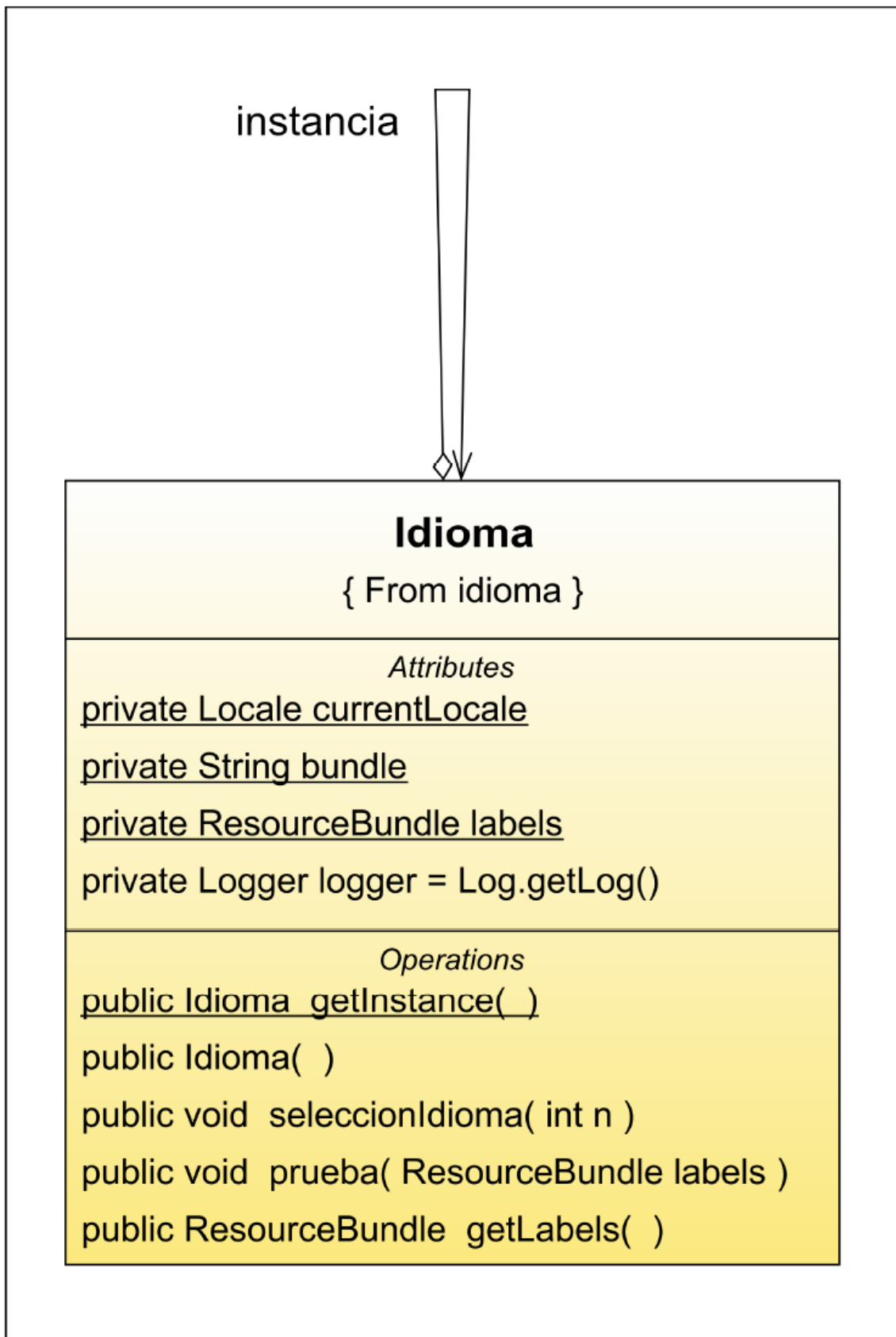
grammar Diagram



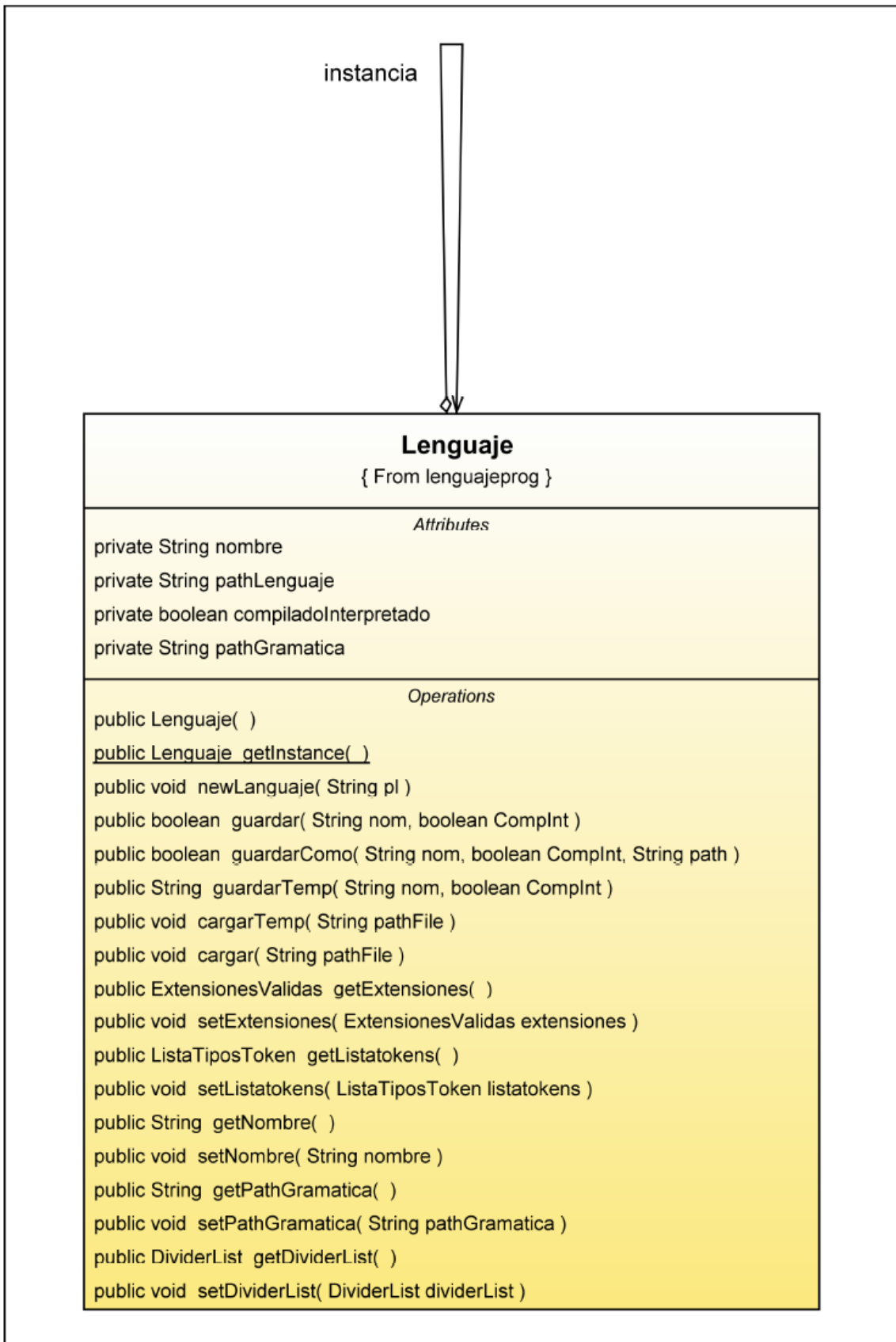
gui Diagram



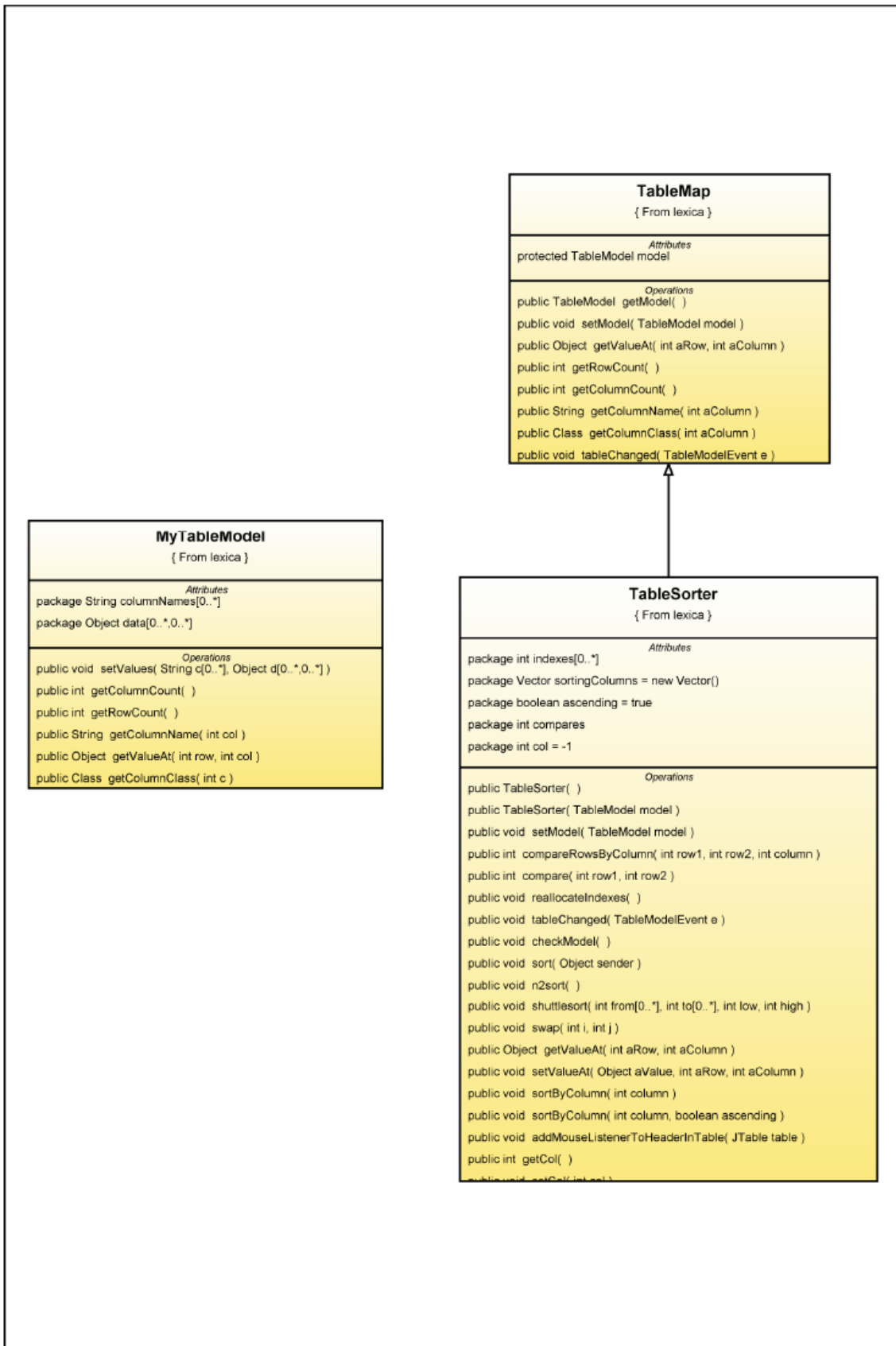
iconos Diagram



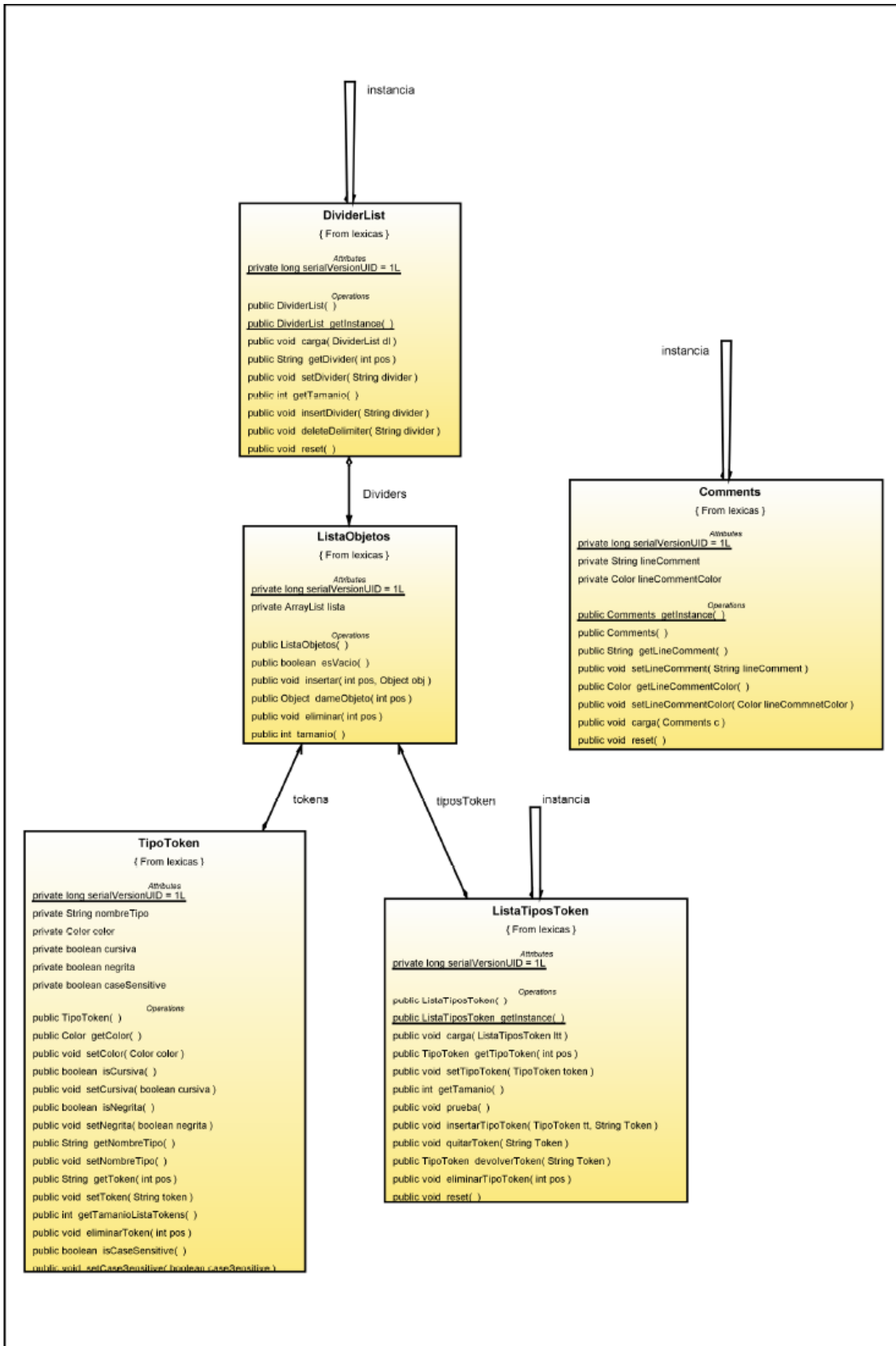
idioma Diagram



lenguajeprog Diagram



lexica Diagram



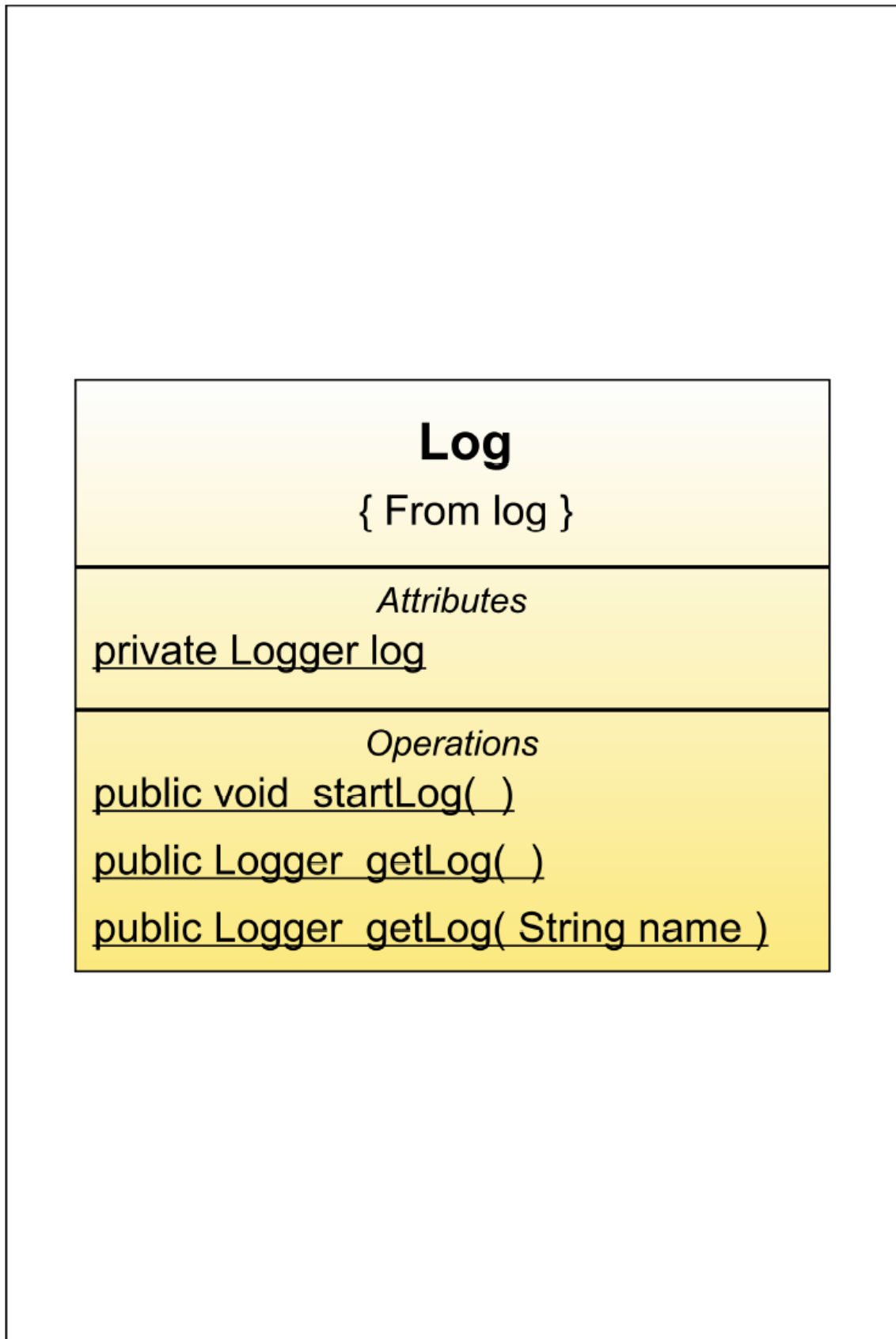
lexicas Diagram

<<datatype>>

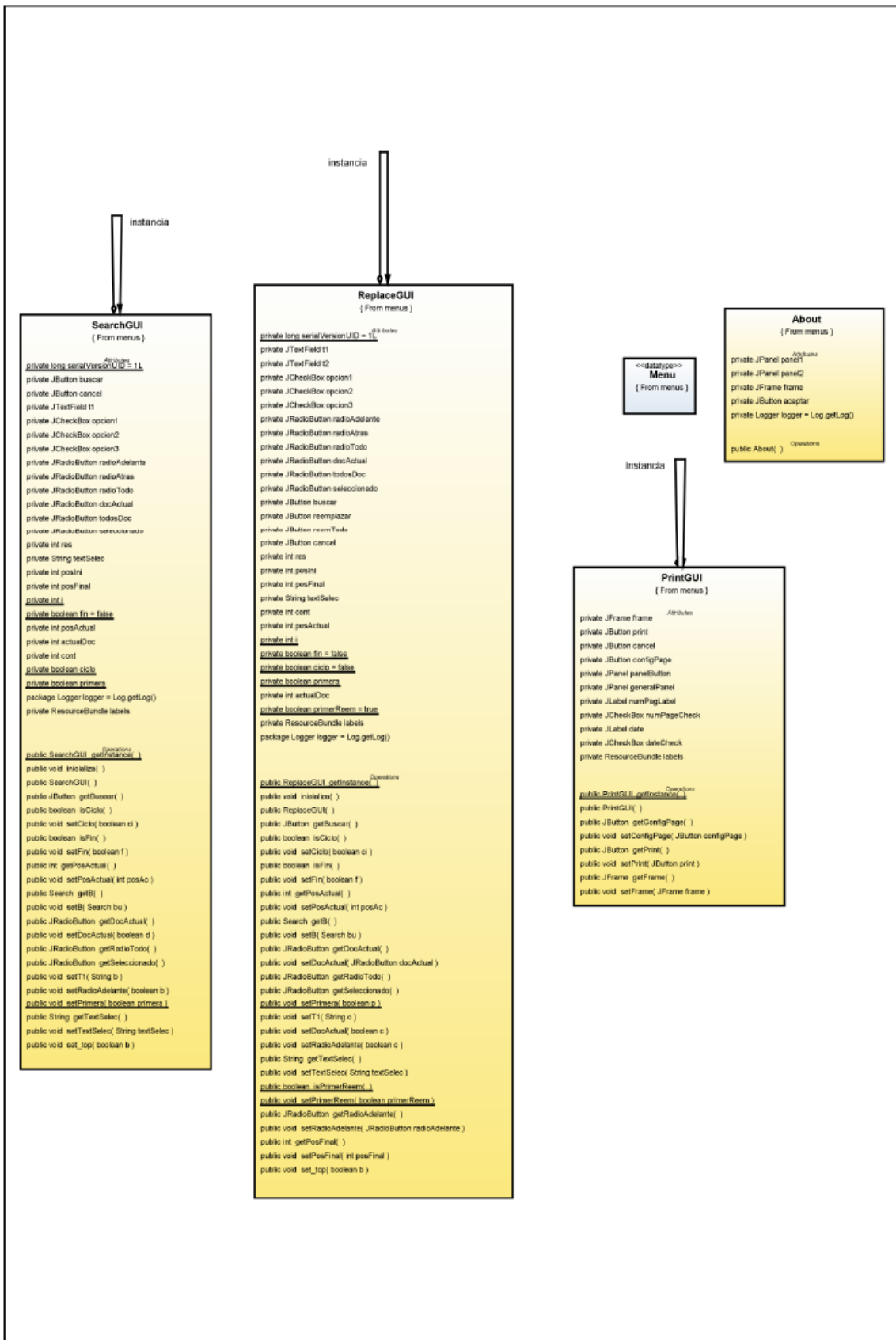
LogTab

{ From log }

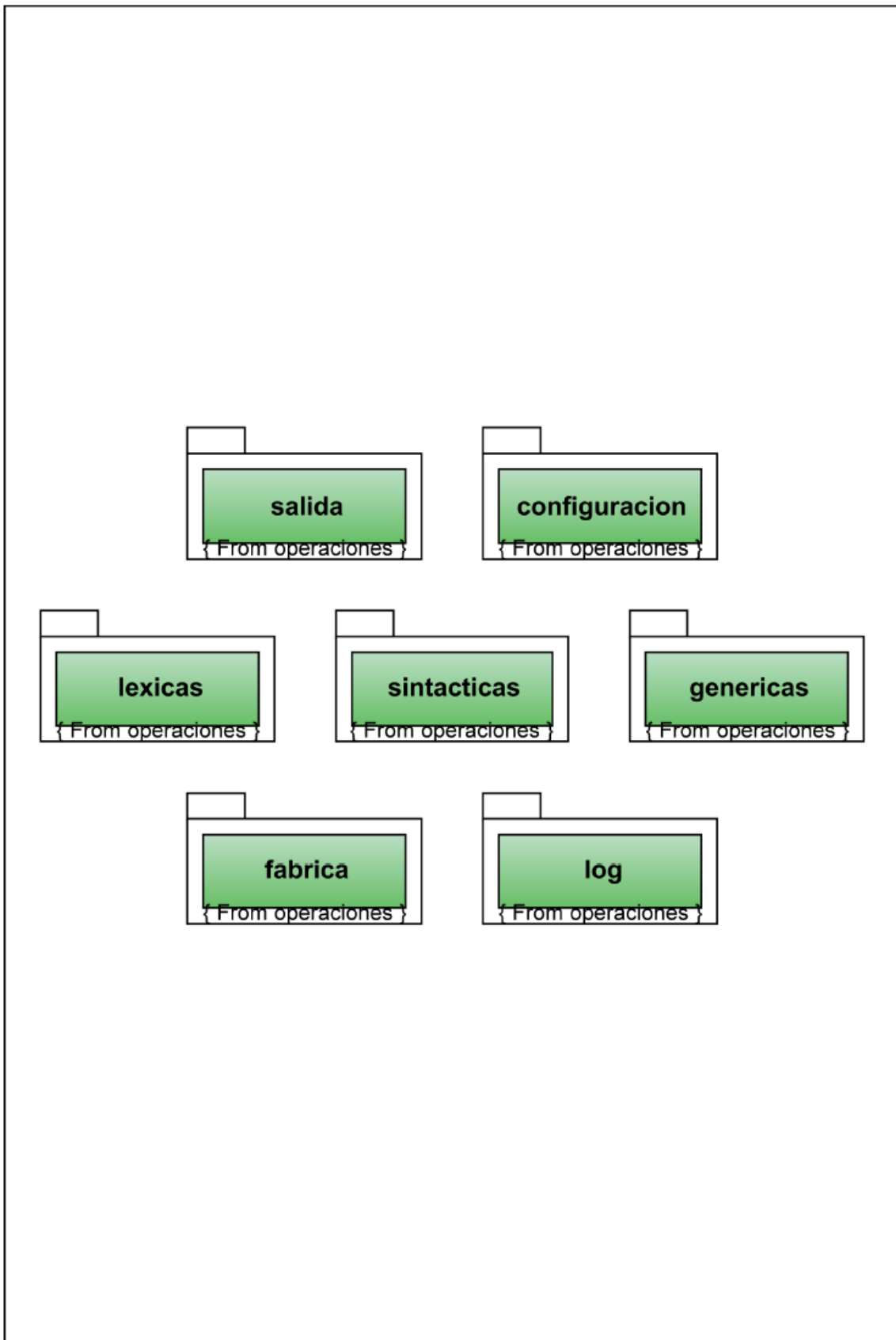
log Diagram



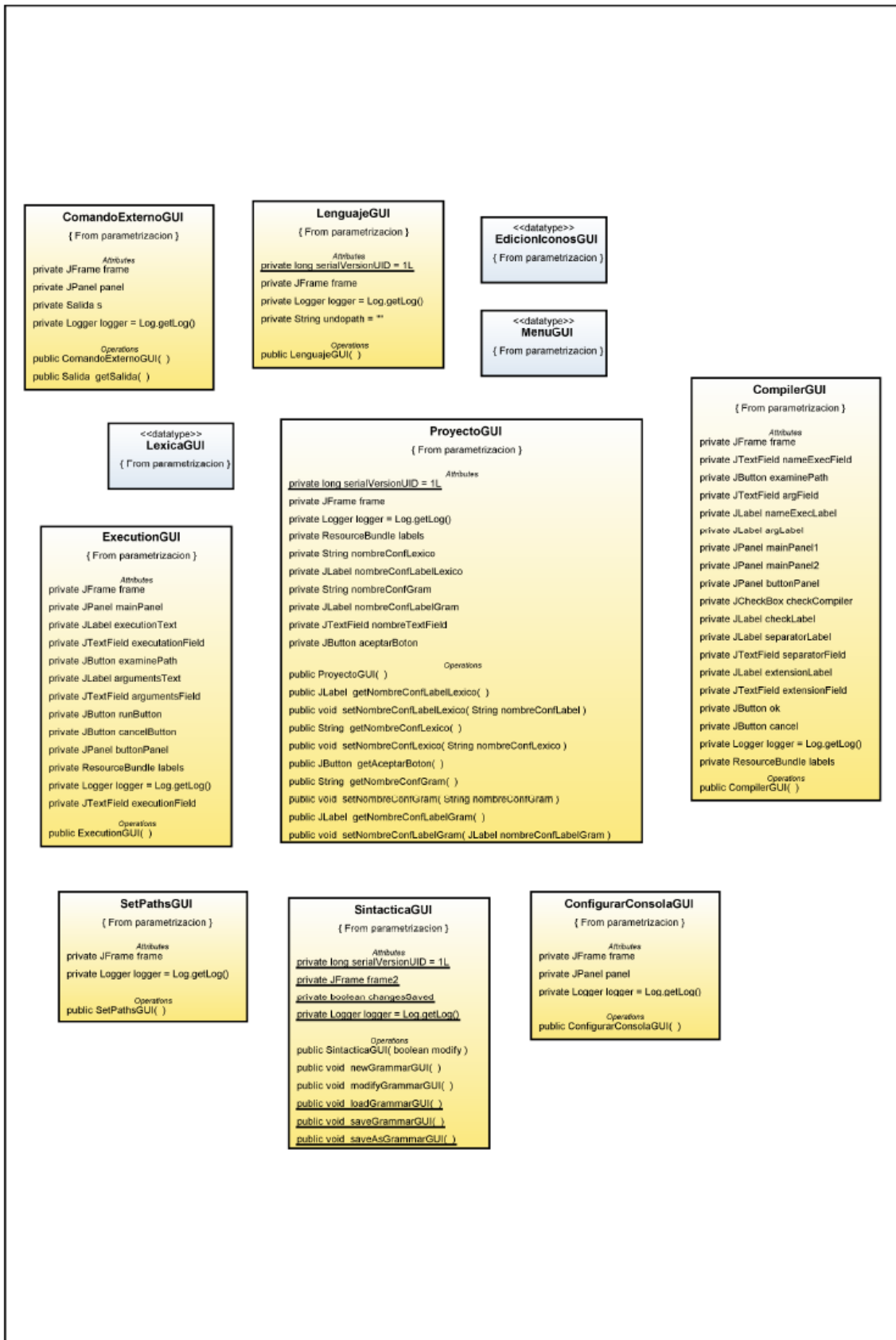
log Diagram



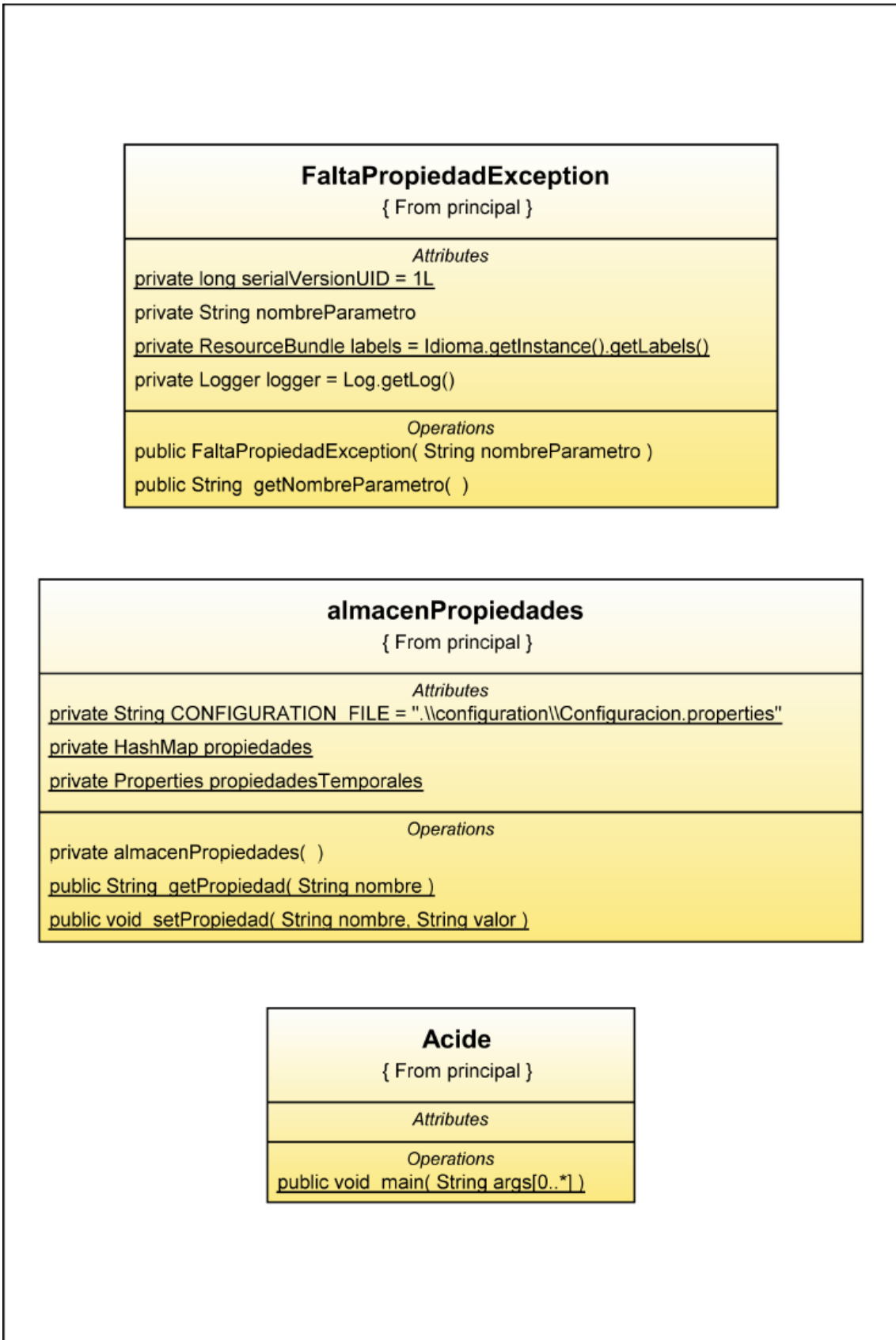
menu Diagram



operaciones Diagram



parametrizacion Diagram



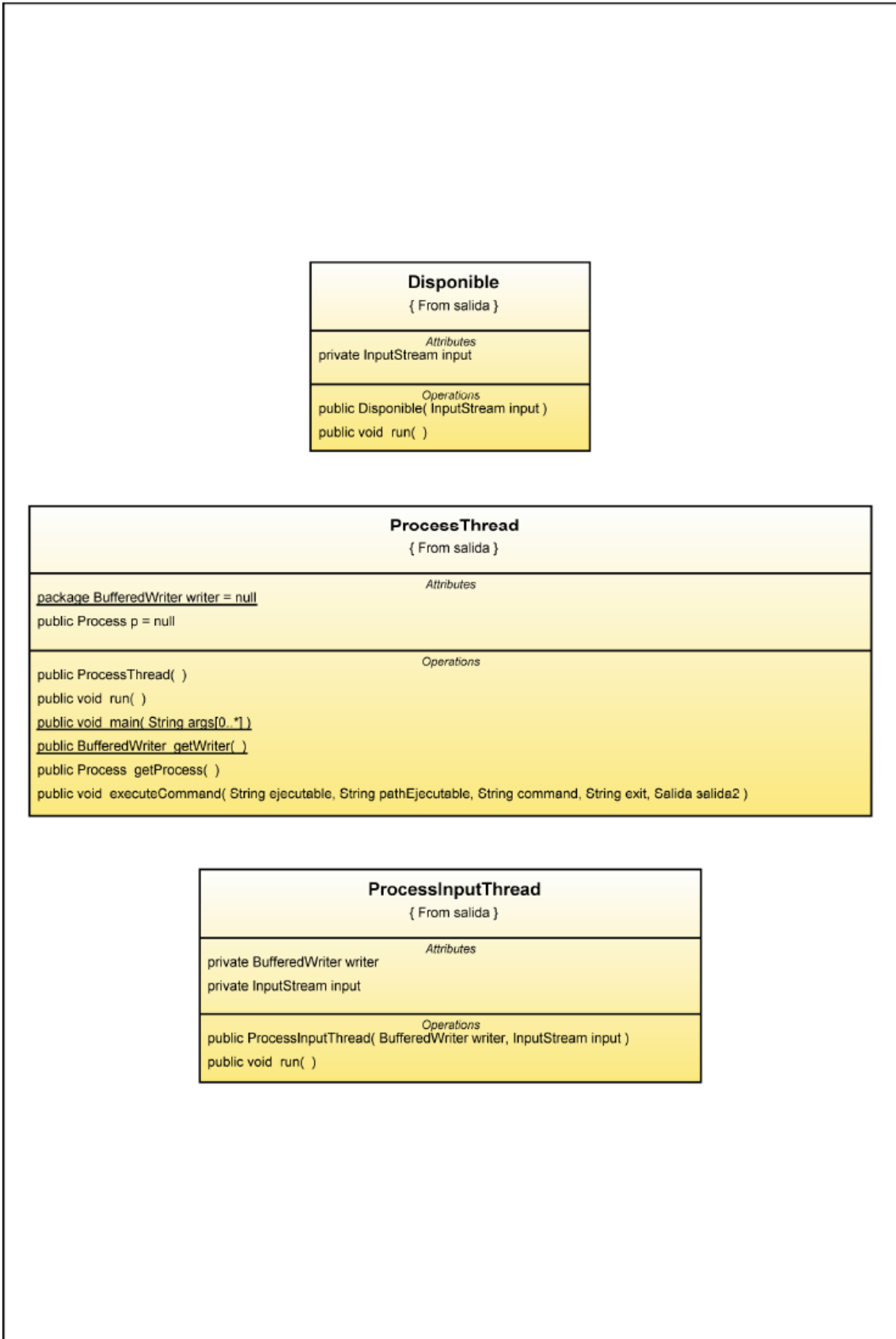
principal Diagram



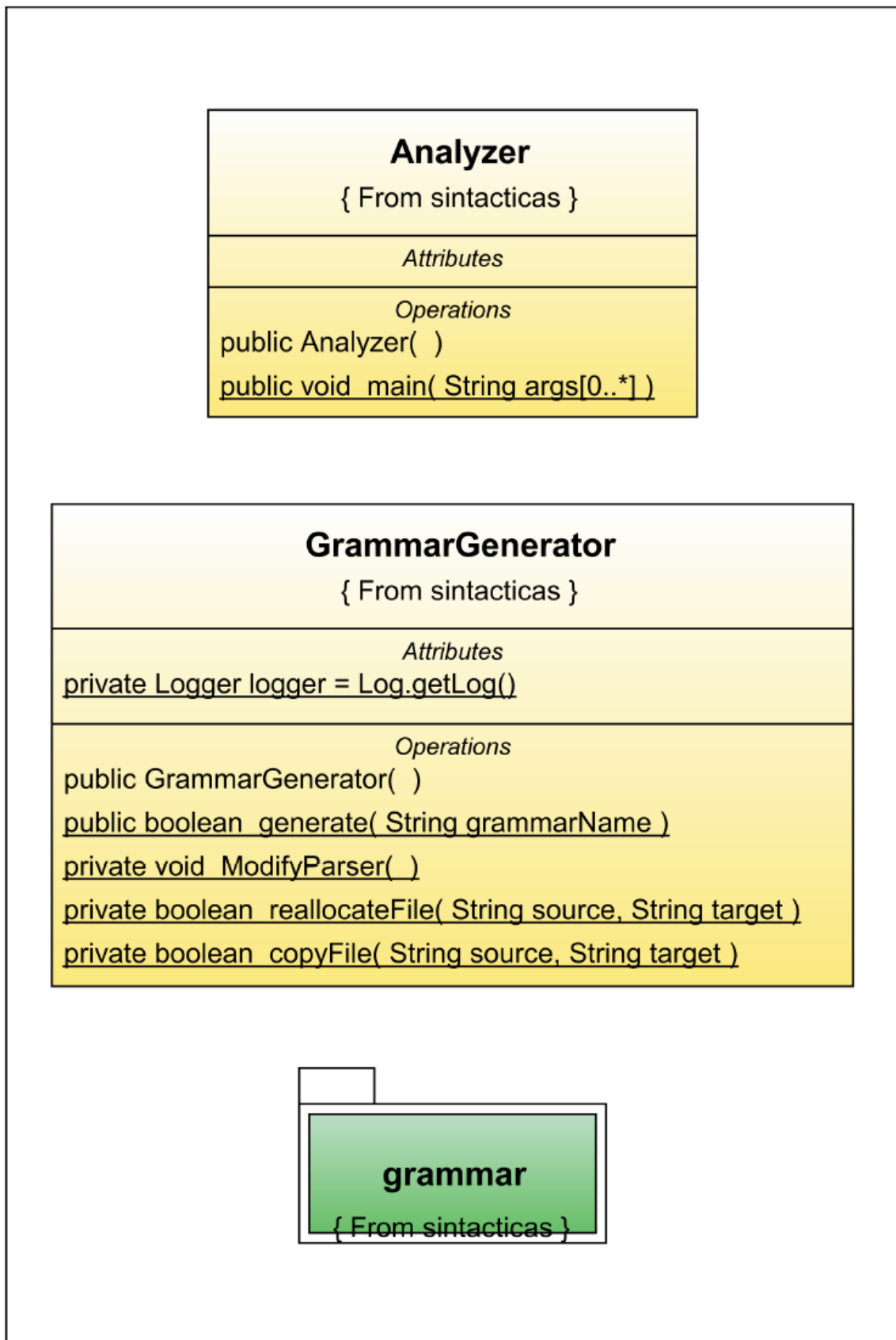
The diagram shows a class named **Salida** with a stereotype `<<datatype>>` and a generalization relationship indicated by a solid line with an open arrowhead pointing to the class. The relationship is labeled `{ From salida }`.

```
classDiagram
    class Salida {
        <<datatype>>
    }
    class BaseClass {
    }
    BaseClass <|-- Salida
```

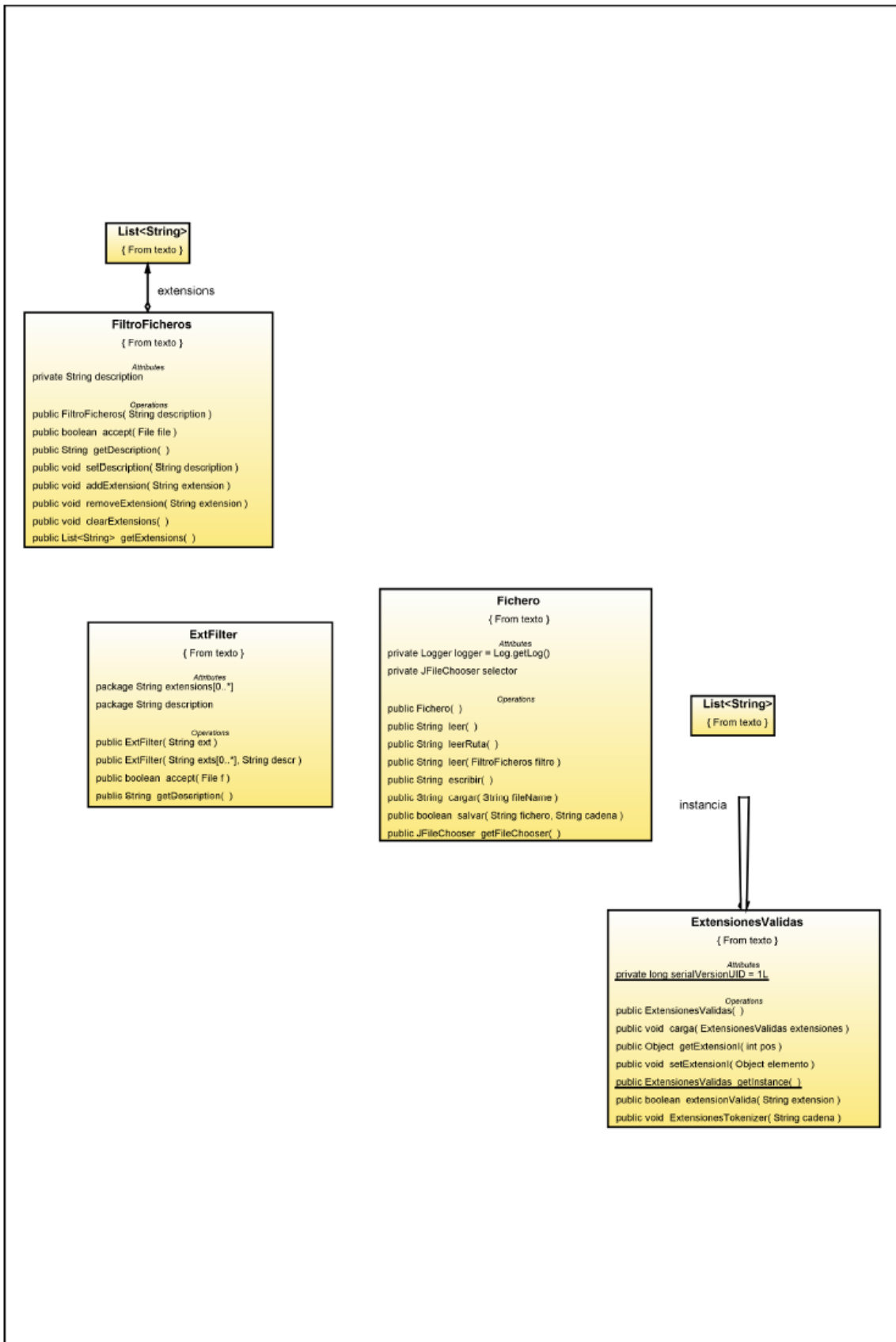
salida Diagram



salida Diagram



sintacticas Diagram



texto Diagram

ACIDE

PALABRAS CLAVE

Versión: 5.0
Fecha: 05/07/2007

LISTA DE PALABRAS CLAVE

1. Análisis léxico
2. Análisis sintáctico
3. Entorno de desarrollo integrado (IDE)
4. Consola
5. Gestión de proyectos
6. Editor de código
7. Lenguajes de programación compilados e interpretados
8. Edición multi-archivo
9. Coloreado de palabras
10. Configurable

INFORMACIÓN DE CONTACTO

Dado que es código libre tenemos todo el código fuente y el ejecutable en las siguientes direcciones de internet:

- Ejecutable: <http://acide.sourceforge.net>
- Código fuente: <http://pide.berlios.de>

Para ponerse en contacto con los desarrolladores del proyecto se pueden usar las siguientes direcciones de correo electrónico:

- Diego Cardiel Freire: diegoenlared@gmail.com
- Juan José Ortiz Sánchez: juanjortizrm@hotmail.com
- Delfín Rupérez Cañas: de300379@hotmail.com

AUTORIZACIÓN

Se autoriza a la Universidad Complutense a difundir y utilizar con fines académicos, no comerciales y mencionando expresamente a sus autores, tanto la propia memoria, como el código, la documentación y/o el prototipo desarrollado.

Diego Cardiel Freire

Juan José Ortiz Sánchez

Delfín Rupérez Cañas