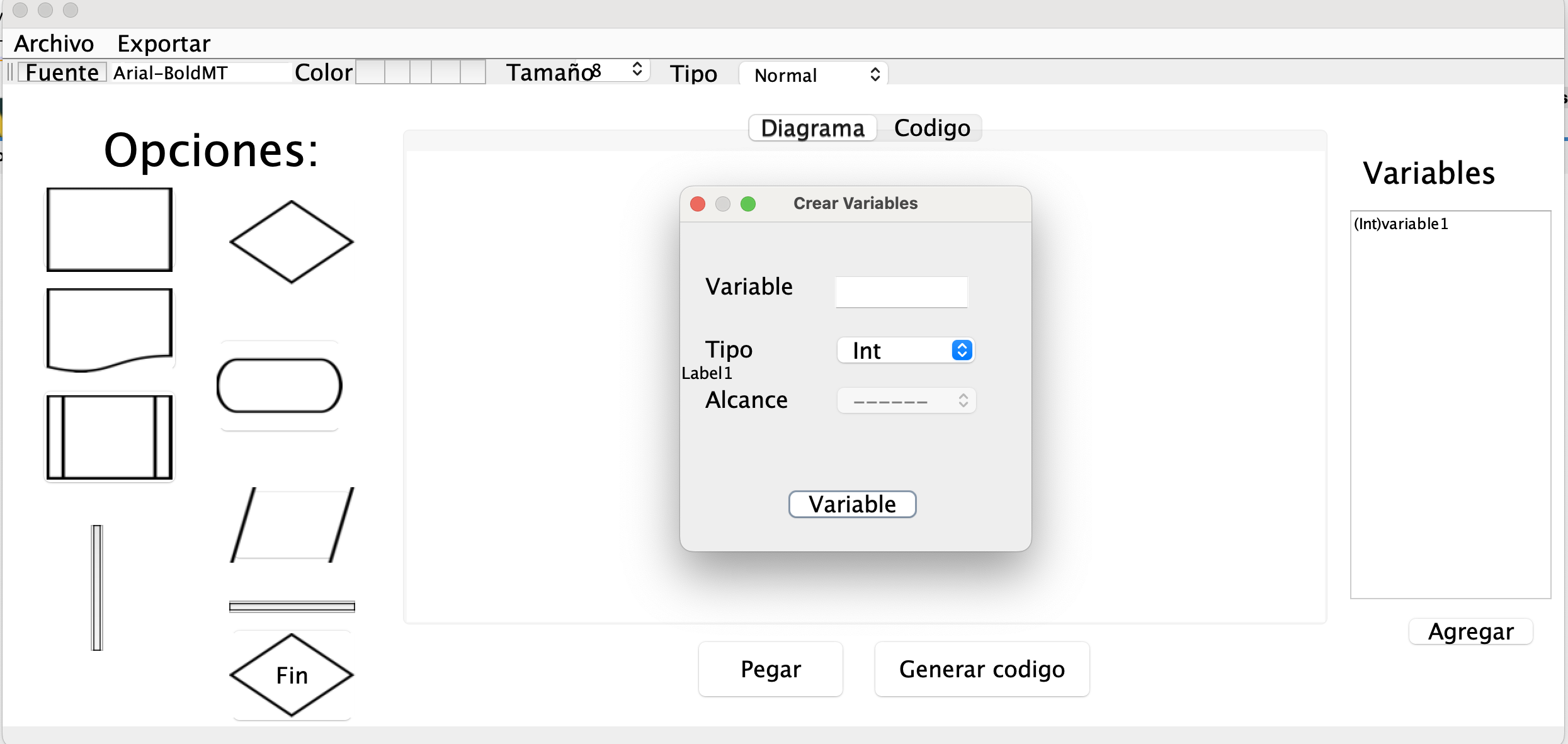
**Proyecto final Programación II.**

**Traductor de UML a Código Java**

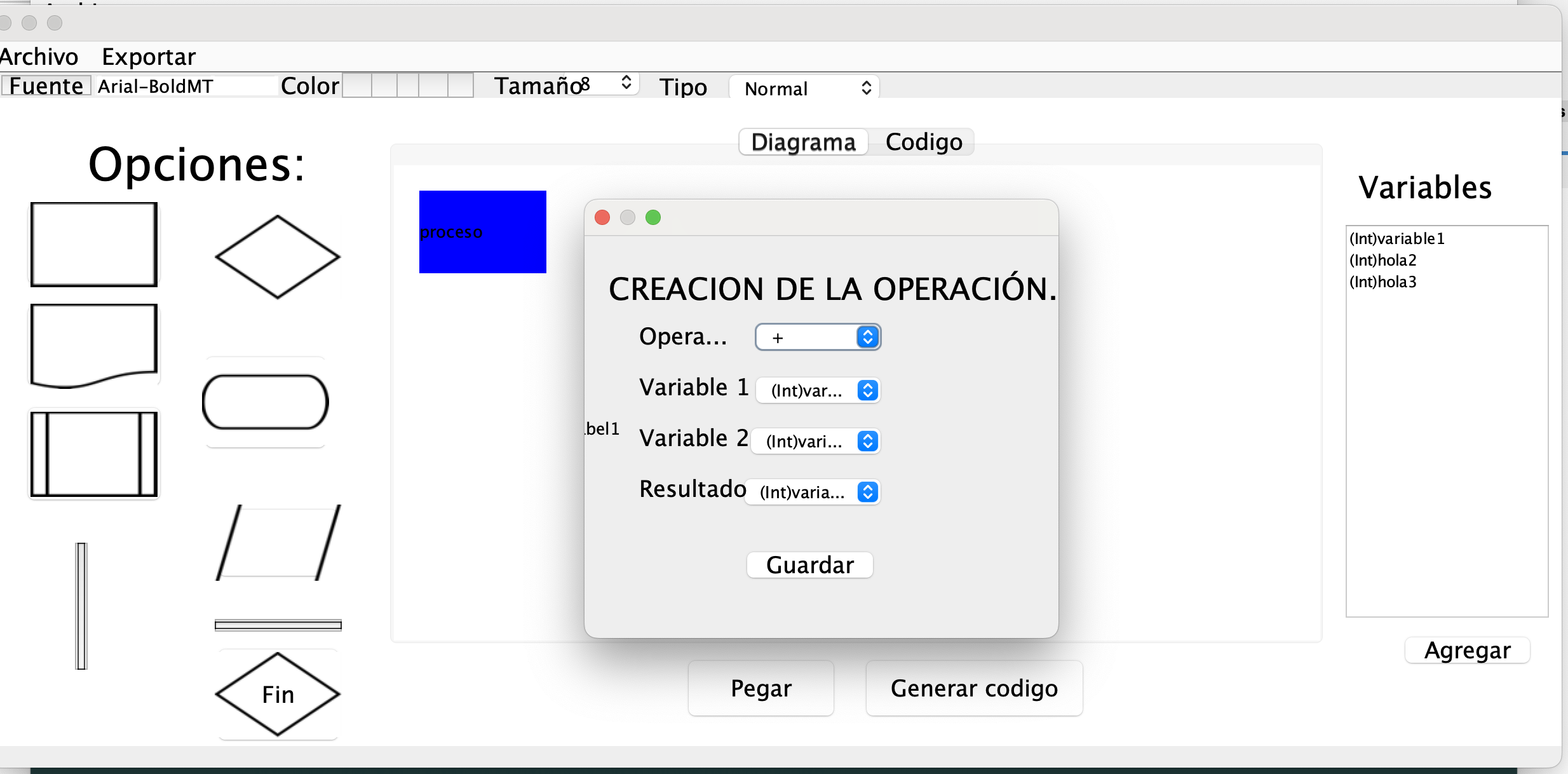
**Objetivo del Proyecto:** El objetivo de este proyecto es desarrollar una aplicación que permita a los estudiantes diseñar diagramas de UML (Lenguaje Unificado de Modelado) y traducirlos automáticamente a código Java. Este proyecto ayudará a los estudiantes a comprender cómo se mapean los elementos de diseño a código real, además de ofrecer una herramienta interactiva para reforzar conceptos de programación orientada a objetos.

**Componentes del Proyecto:**

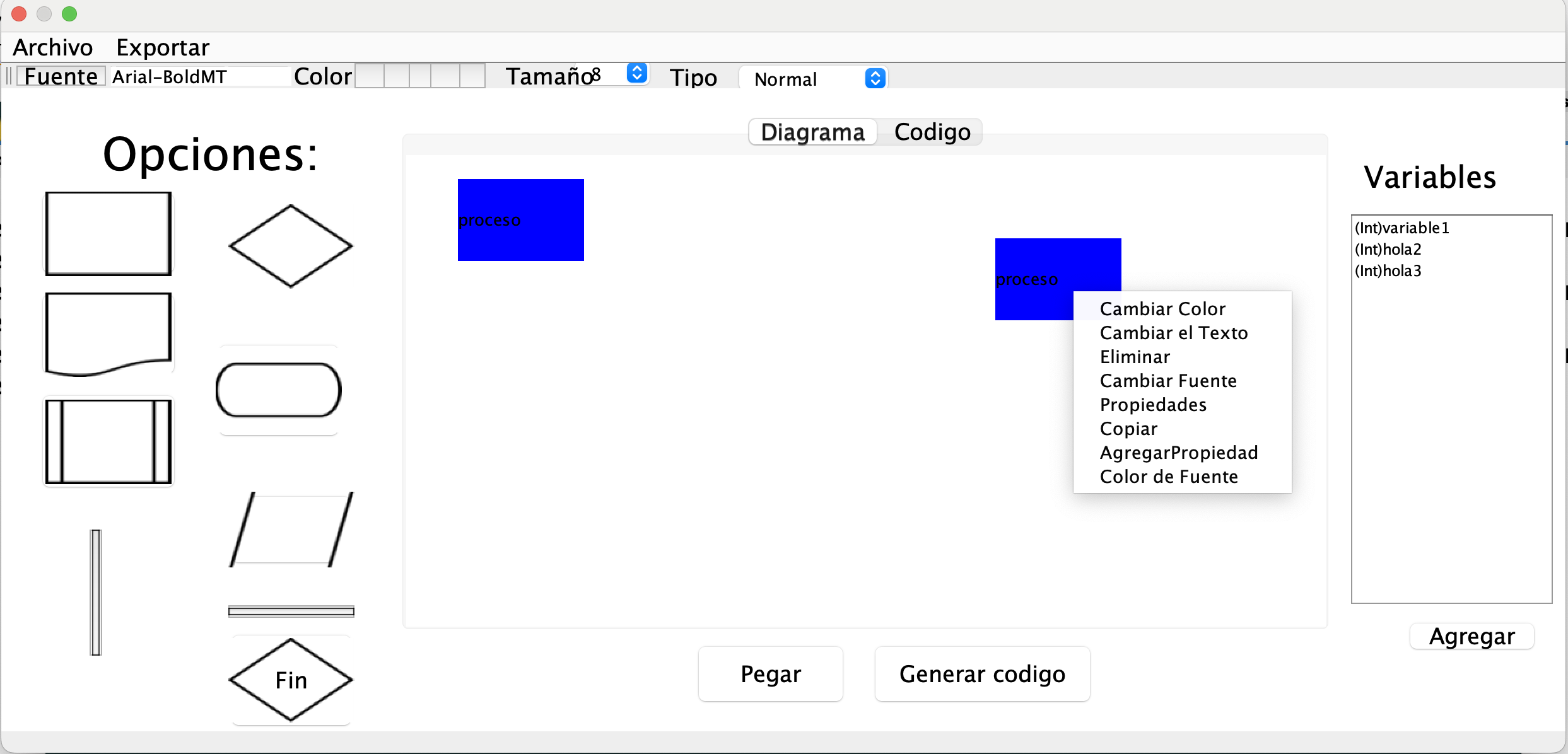
1. **Editor de Diagramas de Flujo UML:**
   * **Opciones Disponibles para diagrama UML:**
     + Inicio
     + Fin
     + Declarar operación
     + If
     + For
     + While
     + System.out.println (sout)
   * **Interactividad:**
     + Los usuarios podrán arrastrar y soltar estos elementos en un área de trabajo o hacer clic sobre ellos para que aparezcan en la pantalla.
     + Los elementos en la pantalla podrán ser copiados y pegados.
     + El diagrama podrá ser guardado(exportado) en formato PDF.
     + Todos los elementos en la pantalla tendrán elementos editables como ser: color, tipo de Fuente, tamaño de la letra y estilo (normal, negrita, itálica).
     + Se podrán agregar variable, para cada variable se listará su tipo de dato y su nombre, se presenta una imagen de ejemplo:



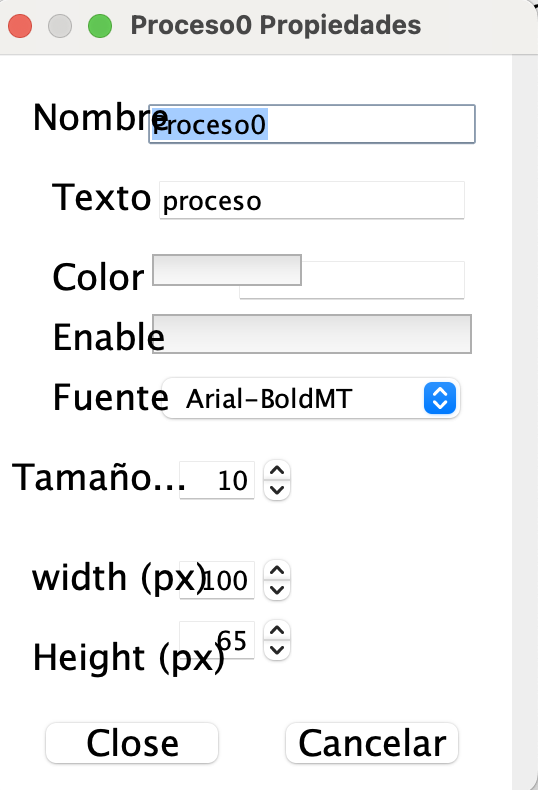
* Se pueden crear operaciones (suma, resta, multiplicación, división) seleccionando las variables que se utilizaran y en que variable se guardara el resultado:



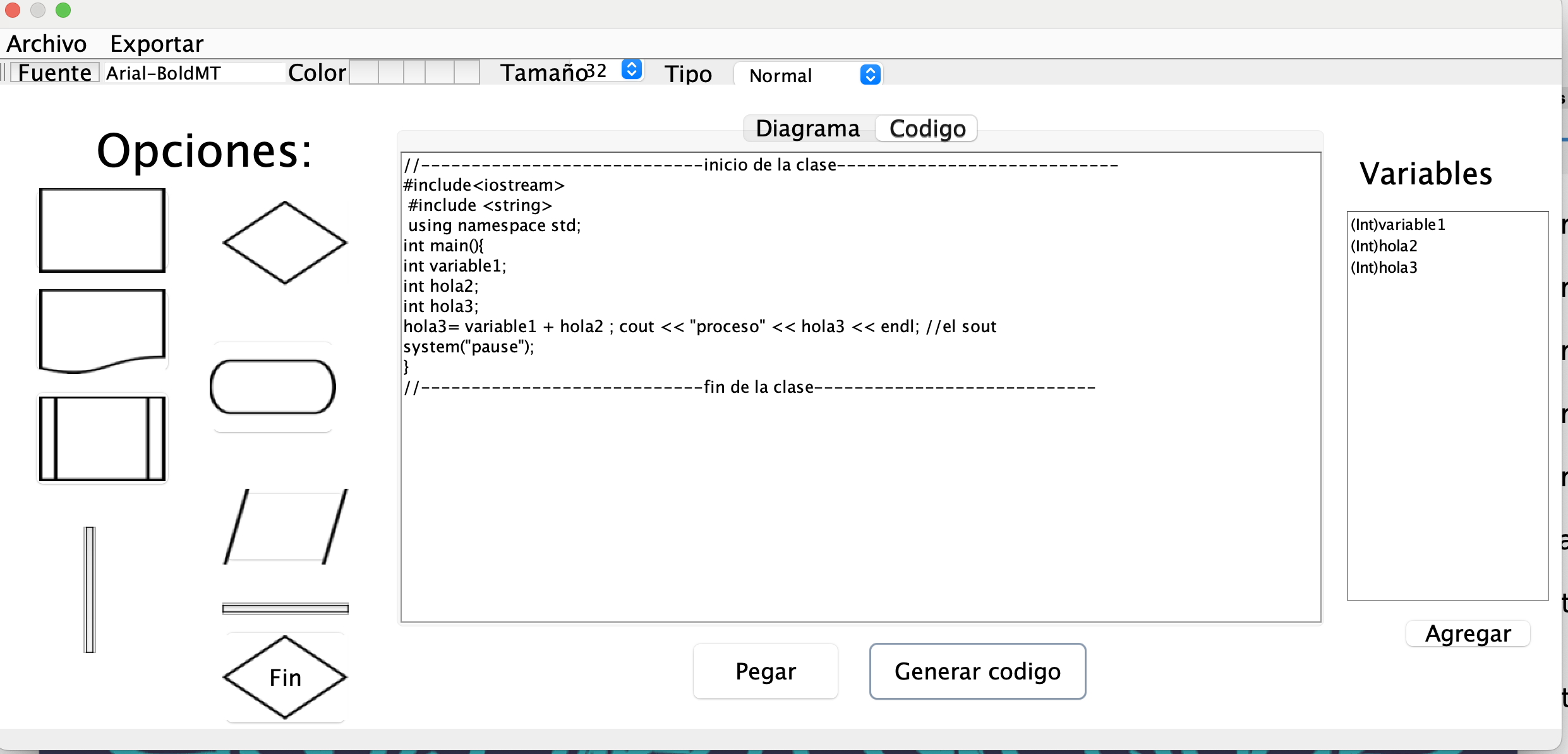
* Cada elemento tendrá un popup menú que mostrara la opciones que se pueden realizar para cada uno como se muestra a continuación:



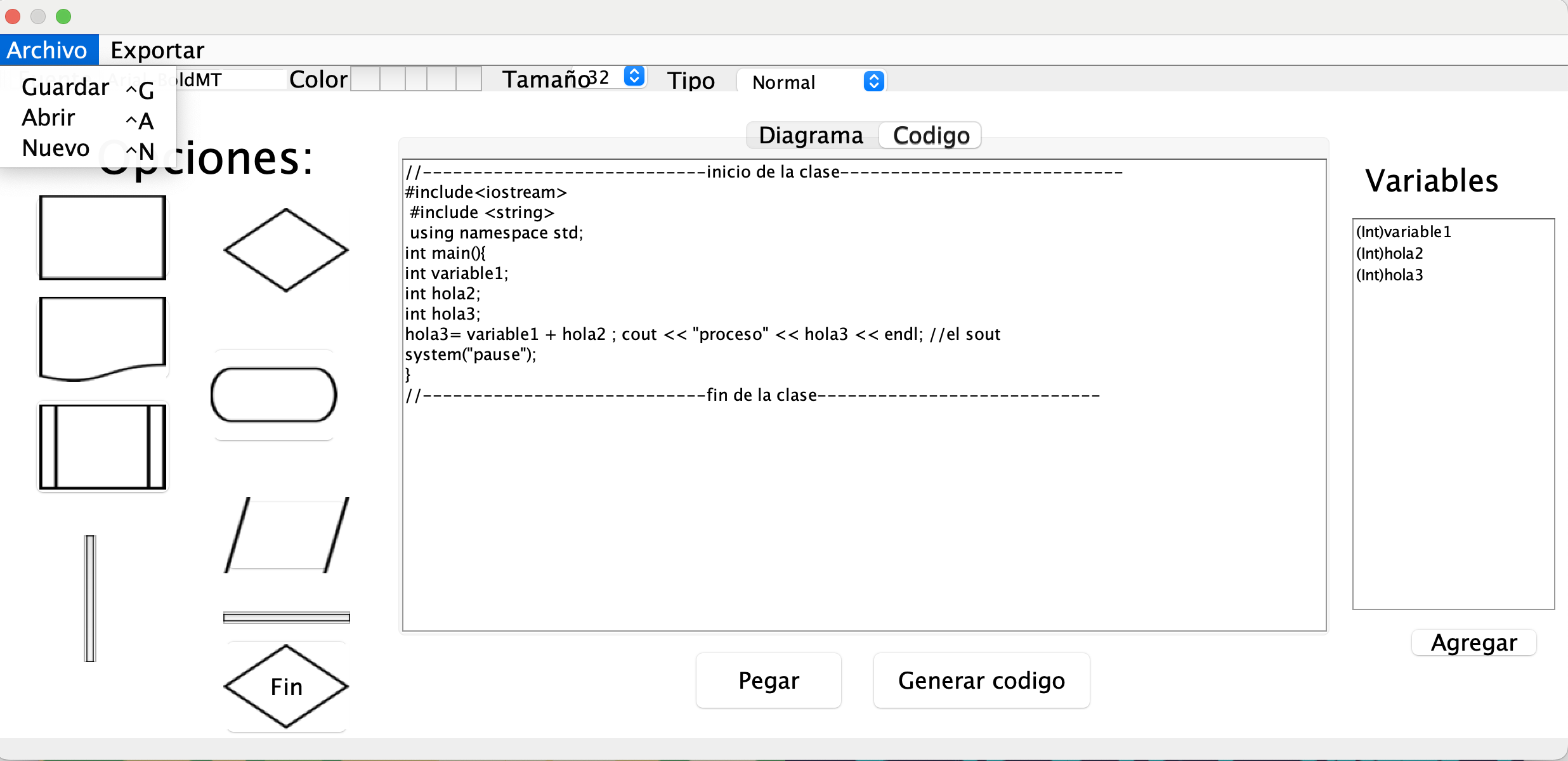
* La opción propiedades mostrará una pantalla emergente como se muestra a continuación:



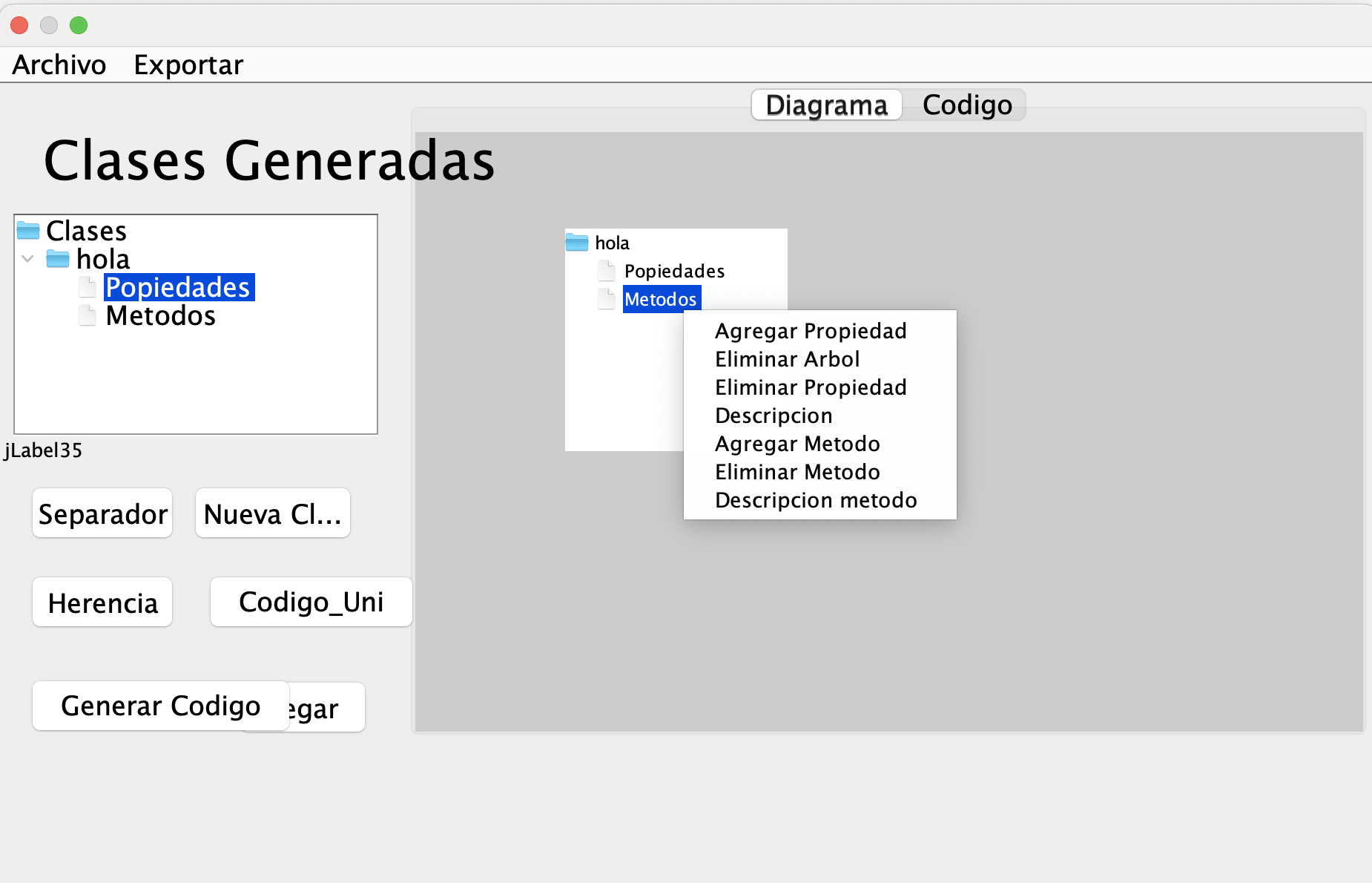
* Al darle a la opción generar código, se deberá generar el código relacionado al UML en la pantalla de edición y se mostrara en un texto área como se muestra a continuación:



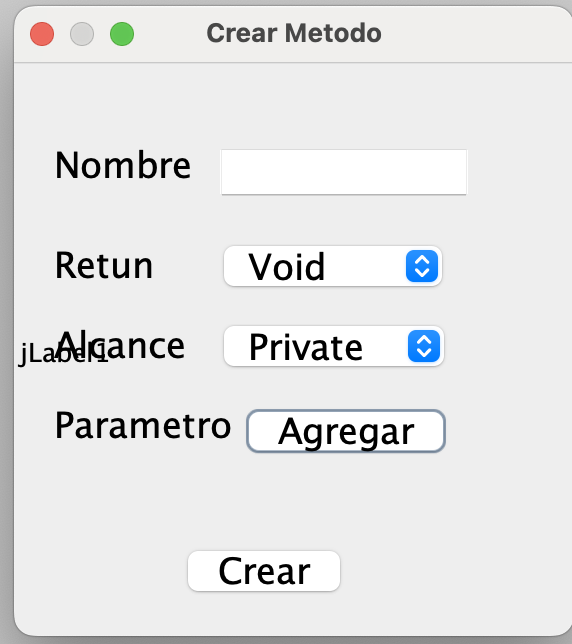
* Se deberá mostrar un menú superior con las opciones de archivo ( crear nuevo archivo, salvar archivo y abrir archivo).



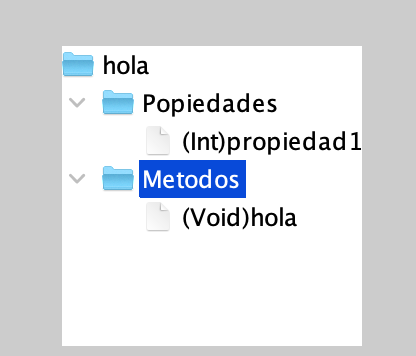
1. **Generador de Clases UML:**
   * **Funciones:**
     + Los usuarios podrán crear clases, agregar atributos y métodos a cada clase, y establecer o eliminar relaciones de herencia entre clases.
   * **Interactividad:**
     + Mover los elementos en un área de trabajo.
     + Los elementos en la pantalla podrán ser copiados y pegados.
     + El diagrama podrá ser guardado(exportado) en formato PDF.
     + El generador permitirá agregar Clases (y listarlas mostrando sus propiedades y métodos).
     + Mediante pop up menús se Deberá agregar la función de agregar propiedades y métodos a cada clase



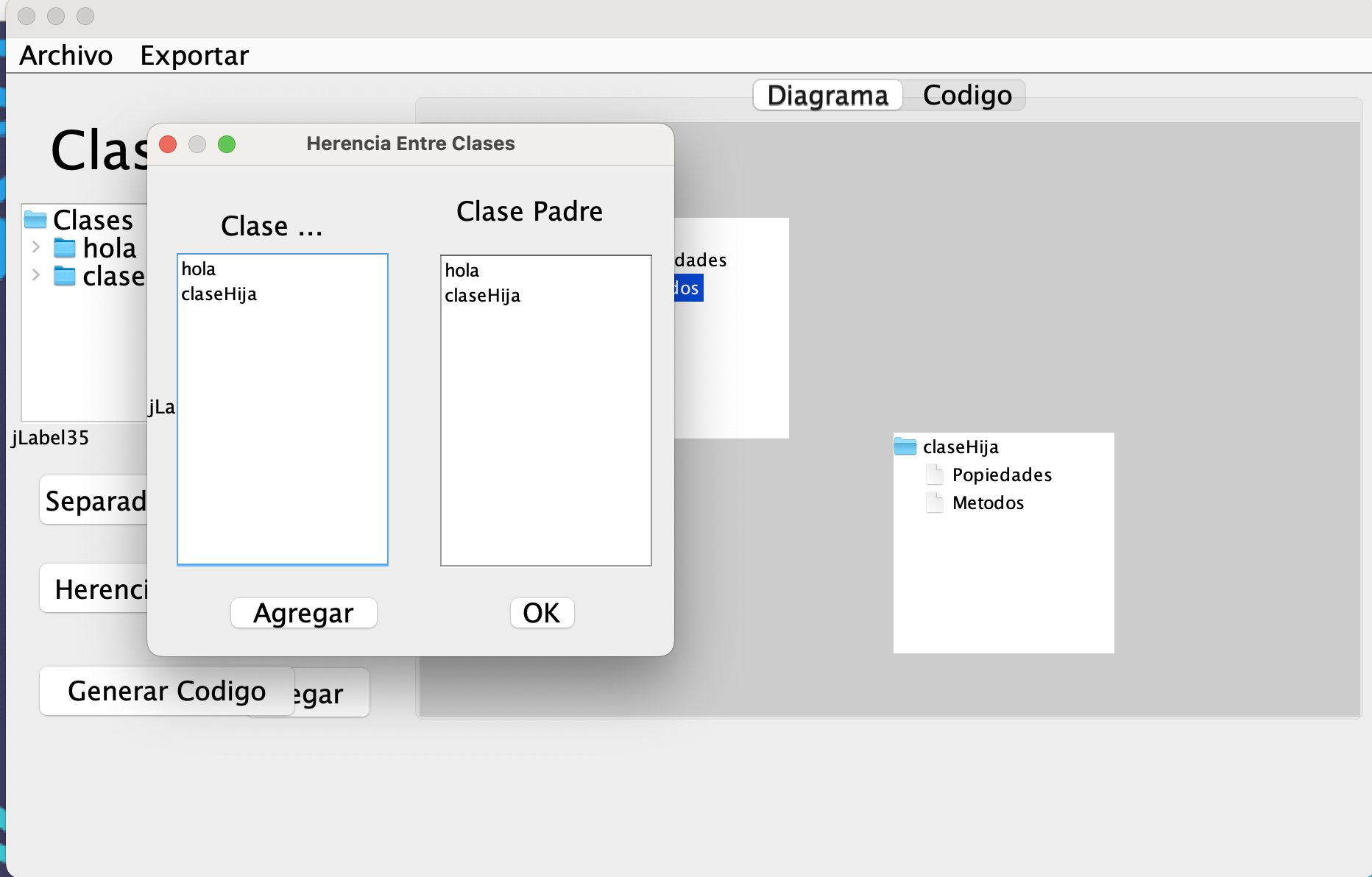
* Las propiedades se agregarán en una pantalla similar a la de agregar variables, para agregar métodos se deberá mostrar una pantalla como la siguiente:



* Cada clase deberá verse así:



* Se podrá definir herencia entre las clases:



1. **Generación Automática de Código:**
   * **Código para Diagramas de Flujo:**
     + A partir del diagrama de flujo creado, el programa generará automáticamente el código Java correspondiente.
   * **Código para Clases UML:**
     + El programa también permitirá generar código Java para las clases y relaciones definidas en el generador de clases.
2. **Salvado y Exportación:**
   * **Guardar Diagramas:**
     + Los usuarios podrán guardar el diagrama actual como un archivo PDF para su posterior revisión o impresión.
   * **Guardar Proyecto:**
     + El proyecto podrá ser guardado y cargado en futuras sesiones, permitiendo la edición y revisión continuas.

**Criterios de Evaluación:**

* Funcionamiento correcto de la interfaz de usuario y de las funcionalidades solicitadas.
* Precisión en la traducción del diagrama UML a código Java.
* Calidad y organización del código fuente.
* Documentación adecuada del proyecto.