

Índice de contenidos

| | | |
|--------|--|----|
| 1. | INTRODUCCIÓN | 2 |
| 2. | REQUISITOS..... | 3 |
| 3. | INSTALACIÓN Y CONFIGURACIÓN | 4 |
| 3.1 | Instalación de la Plataforma de Ejecución Java (JRE) | 4 |
| 3.2 | Instalación de la herramienta ArgoUML | 6 |
| 4. | PUESTA EN FUNCIONAMIENTO | 9 |
| 5. | GUIA DE USUARIO..... | 13 |
| 5.1 | PARTE DE DISEÑO..... | 13 |
| 5.1.1 | Crear un nuevo Proyecto. | 14 |
| 5.1.2 | Abrir proyecto..... | 15 |
| 5.1.3 | Importar Archivo XML | 18 |
| 5.1.4 | Refrescar contenidos | 24 |
| 5.1.5 | Seleccionar Clase Principal | 25 |
| 5.1.6 | Modificar Preferencias | 27 |
| 5.1.7 | Guardar Proyecto | 29 |
| 5.1.8 | Guardar Proyecto como | 30 |
| 5.1.9 | Fijar Diagrama..... | 33 |
| 5.1.10 | Ocultar Menú Principal | 34 |
| 5.1.11 | Borrar todos los elementos | 35 |
| 5.1.12 | Generar código | 36 |
| 5.1.13 | Gestionar Clase | 38 |
| 5.1.14 | Gestionar Atributos | 43 |
| 5.1.15 | Gestionar Métodos | 50 |
| 5.1.16 | Ayuda de Diseño | 57 |
| 5.2 | PARTE DE IDE DE JAVA | 58 |
| 5.2.1 | Nuevo Archivo Java..... | 59 |
| 5.2.2 | Abrir Archivo Java | 61 |
| 5.2.3 | Borrar Archivo Java..... | 65 |
| 5.2.4 | Guardar Todo..... | 66 |
| 5.2.5 | Guardar Pestaña | 67 |
| 5.2.6 | Abrir contenedor de archivos..... | 68 |
| 5.2.7 | Abrir Archivo en pestaña | 69 |
| 5.2.8 | Imprimir Archivo..... | 69 |
| 5.2.9 | Compilar Pestaña..... | 71 |
| 5.2.10 | Cerrar Pestaña | 73 |
| 5.2.11 | Copiar selección de código | 73 |
| 5.2.12 | Pegar selección de código..... | 74 |
| 5.2.13 | Cortar selección de código..... | 74 |
| 5.2.14 | Cambiar a Vista Diseño | 75 |
| 5.2.15 | Generar Documentación Javadoc | 76 |
| 5.2.16 | Compilar Proyecto | 78 |
| 5.2.17 | Ejecutar Proyecto..... | 79 |
| 5.2.18 | Ayuda de IDE de Java | 80 |

1. INTRODUCCIÓN

Este manual ofrece a los usuarios de la aplicación toda la información referente al sistema y trata de ser una guía para su correcto manejo, desde un punto de vista práctico e intuitivo para facilitar a cualquier tipo de usuario la utilización del sistema.

A continuación se detallarán detenidamente los pasos a seguir para llevar a cabo su instalación, configuración y puesta en funcionamiento.

Este manual se orienta hacia usuarios finales del sistema. Debido a que el Sistema se divide en dos partes Parte de Diseño y parte de IDE de Java, el manual de usuario se dividirá en dos secciones, en cada una de ellas se explicarán las opciones existentes.

2. REQUISITOS

Requisitos Hardware

El usuario que desee utilizar la herramienta precisará de un ordenador con las siguientes características mínimas:

- Pentium 1.73 GHz con memoria RAM de 1Gbyte. (para un mejor funcionamiento se recomienda un ordenador superior).
- Conexión a Internet (Recomendable).

Requisitos Software

El usuario necesitará tener instalado en su equipo la plataforma de ejecución de programas Java (JRE) para poder ejecutar la herramienta de Generación de código java.

Otros requisitos

El usuario deberá de tener instalado el navegador Firefox para la visualización de los ficheros de ayuda de la herramienta generada, creados con javadoc.

3. INSTALACIÓN Y CONFIGURACIÓN

3.1 Instalación de la Plataforma de Ejecución Java (JRE)

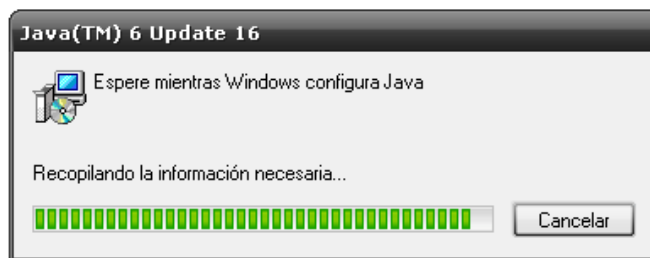
El software correspondiente a la plataforma de ejecución Java (JRE) se encuentra almacenado en el directorio JRE dentro de software. La versión que encontramos en ese directorio es la versión para ordenadores con procesadores de 32 Bits y Windows como SO.

Si desea descargar la última versión del JRE, o bien descargar la versión para procesadores de 64 bits, o bien desea descargarlo para otro SO que no sea Windows (Linux, Solaris, Mac...), por favor descárguelo de esta página y siga las instrucciones de instalación en castellano que en ella se indican para la versión descargada.

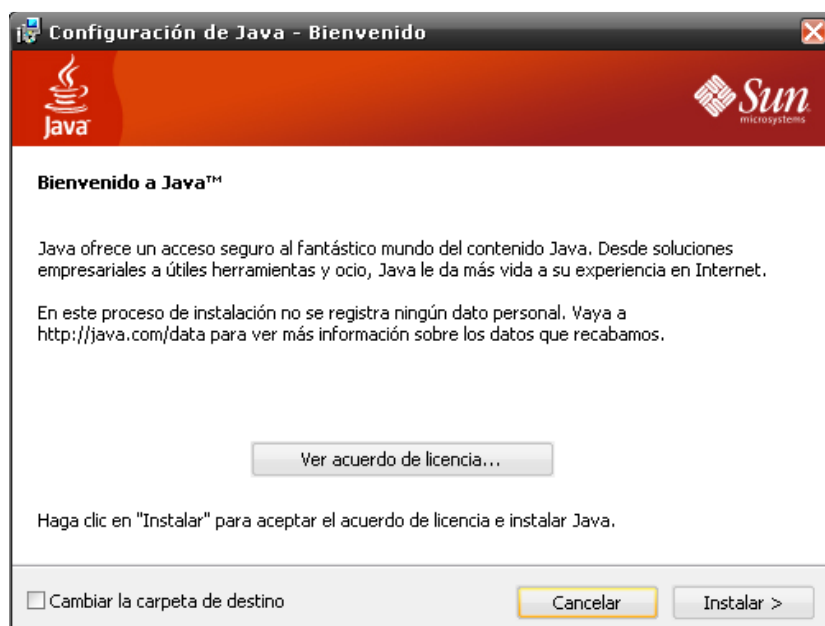
<http://www.java.com/es/download/index.jsp>

Para instalar JRE que se proporciona en el CD es necesario seguir los siguientes pasos:

1. Ejecutar el archivo **jre-6u16-windows-i586.exe** situado en el directorio anteriormente mencionado.



2. El instalador descomprime los archivos necesarios para proseguir. Después muestra una pantalla de bienvenida con el logotipo, y un botón desde el que se puede acceder al contrato de licencia. Cuando termine de leer el acuerdo, haga clic en el botón **Instalar** para aceptar los términos y proseguir con la instalación.



Si lo desea el usuario puede cambiar el directorio en el que se almacenarán los archivos del entorno de ejecución de java, por defecto se instalan en el directorio: C:\Archivos de programa\Java

3. Una vez pulsado el botón instalar comienza el proceso de instalación, y se abre una ventana desde la que se puede ver una barra de progreso de la instalación.



4. Terminada la instalación, si no se ha producido ningún error el sistema muestra la siguiente ventana informativa. Para terminar el usuario debe de pulsar el botón 'Cerrar'.



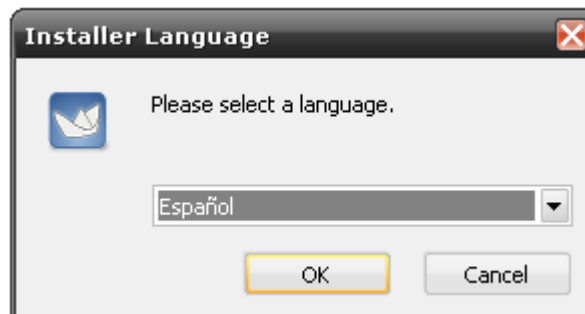
5. **Verificar la instalación.** Para comprobar si el JRE está instalado y activado, y si funciona correctamente en el equipo, ejecute este applet de prueba desde la Web de Sun: [Prueba del entorno de ejecución de Java](#).

3.2 Instalación de la herramienta ArgoUML

El software correspondiente a la aplicación ArgoUML se encuentra almacenado en el directorio ArgoUML dentro de software. El software se ha descargado de la página oficial de ArgoUML: <http://argouml.tigris.org/>.

Para instalar el programa debemos de seguir los siguientes pasos:

1. Ejecutar el archivo **ArgoUML-0.28.1-setup.exe** situado en el directorio anteriormente mencionado. Se abre una ventana en la que debemos escoger el idioma.



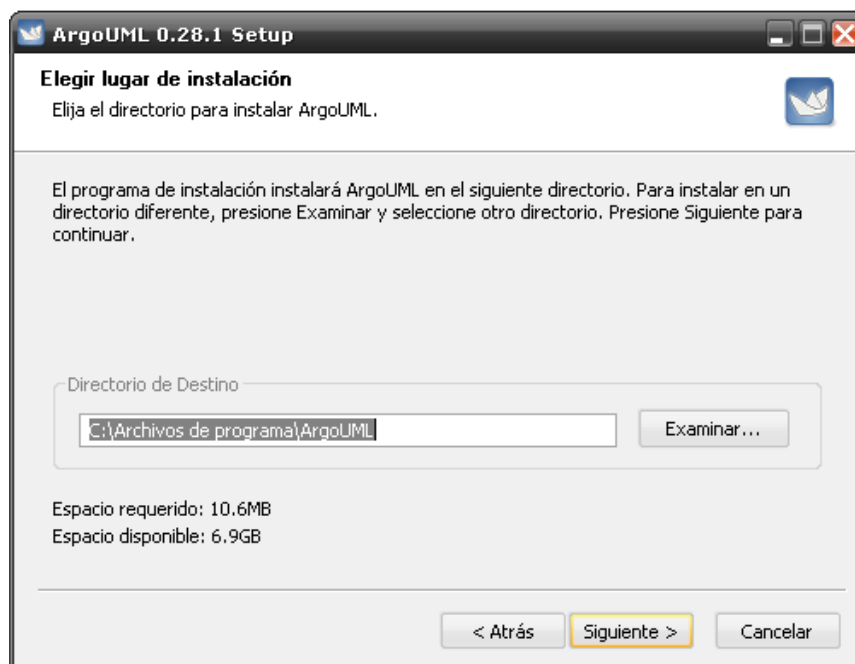
2. A continuación se muestra una pantalla de bienvenida, para continuar con la instalación debemos pulsar el botón Siguiente.



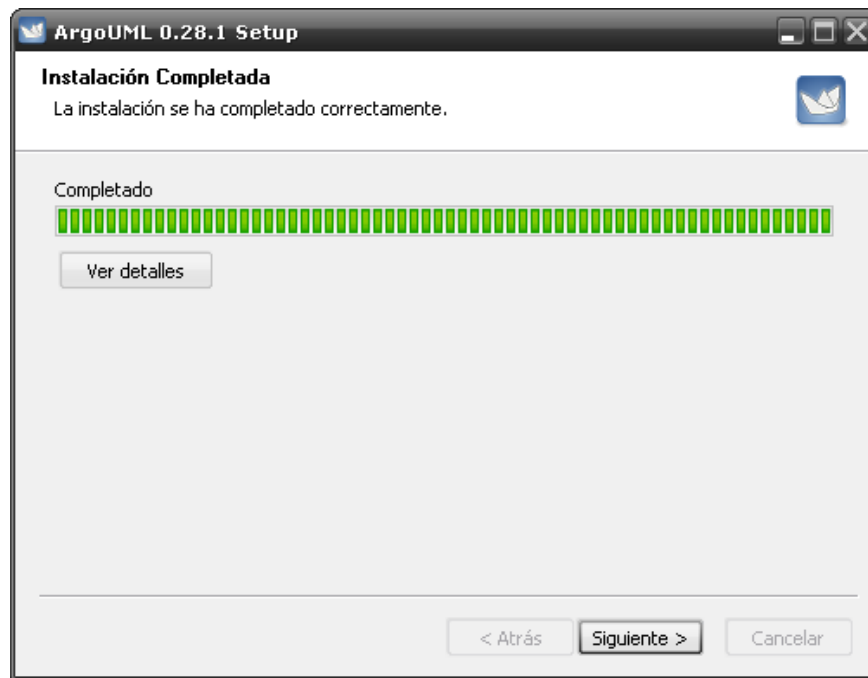
3. Se abre una nueva ventana en la que debemos escoger los componentes que queremos instalar, en este caso hacemos clic en el check de ArgoUML y pulsamos el botón siguiente.



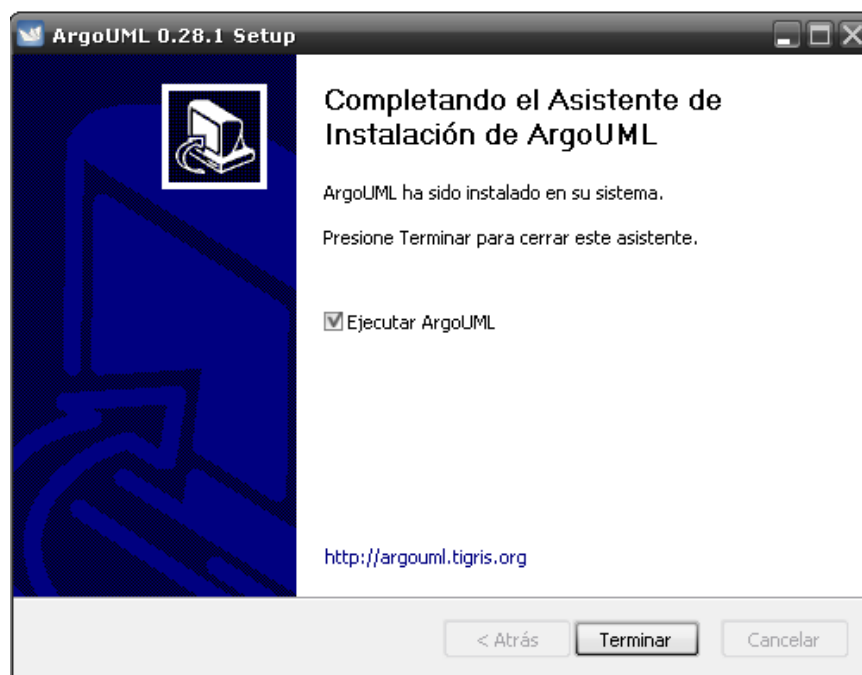
4. Escogemos un directorio en el que deseemos almacenar los archivos del programa y pulsamos el botón siguiente.



5. Una vez pulsado el botón instalar comienza el proceso de instalación, y se abre una ventana desde la que se puede ver una barra de progreso de la instalación.



6. Si la instalación ha finalizado con éxito, se muestra una última ventana como la siguiente y se termina la instalación.



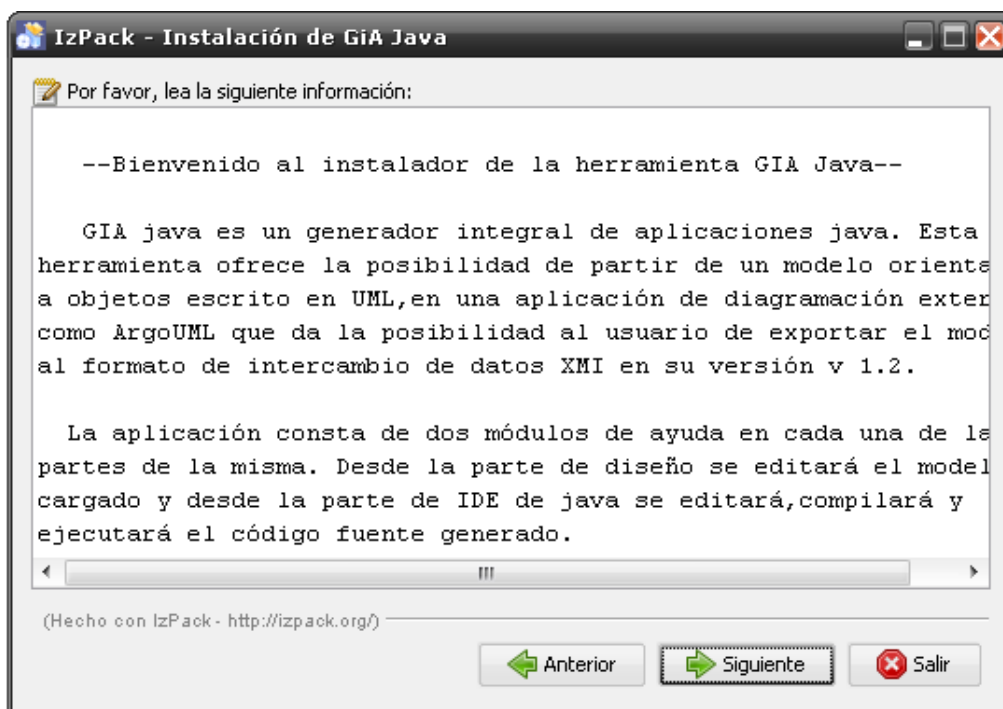
4. PUESTA EN FUNCIONAMIENTO

Todo el software que necesario para el correcto funcionamiento de la herramienta se encuentra en la carpeta Aplicación situada en el CD adjunto. Para su puesta en funcionamiento se deben realizar los siguientes pasos:

1. Accedemos a la carpeta GiA Java, en esta carpeta se encuentra un archivo denominado **install.jar**, lo ejecutamos y se abre el instalador del programa.



2. A continuación se abre una ventana desde la que se pueden consultar la funcionalidad de la herramienta y pulsamos el botón siguiente.



3. A continuación se muestra una ventana desde la que el usuario debe elegir la ruta de instalación de la aplicación y se pulsa el botón siguiente.



4. Se abre una nueva ventana desde la que se seleccionarán los paquetes que se deseen incluir en la instalación. El paquete base es obligatorio.



5. Una vez pulsado el botón instalar comienza el proceso de instalación, y se abre una ventana desde la que se puede ver una barra de progreso de la instalación.

Completada la instalación se pulsa el botón siguiente.



6. A continuación se configurarán los accesos directos de la aplicación.



7. Una vez terminado el instalador muestra la ventana de finalización.



5. GUIA DE USUARIO

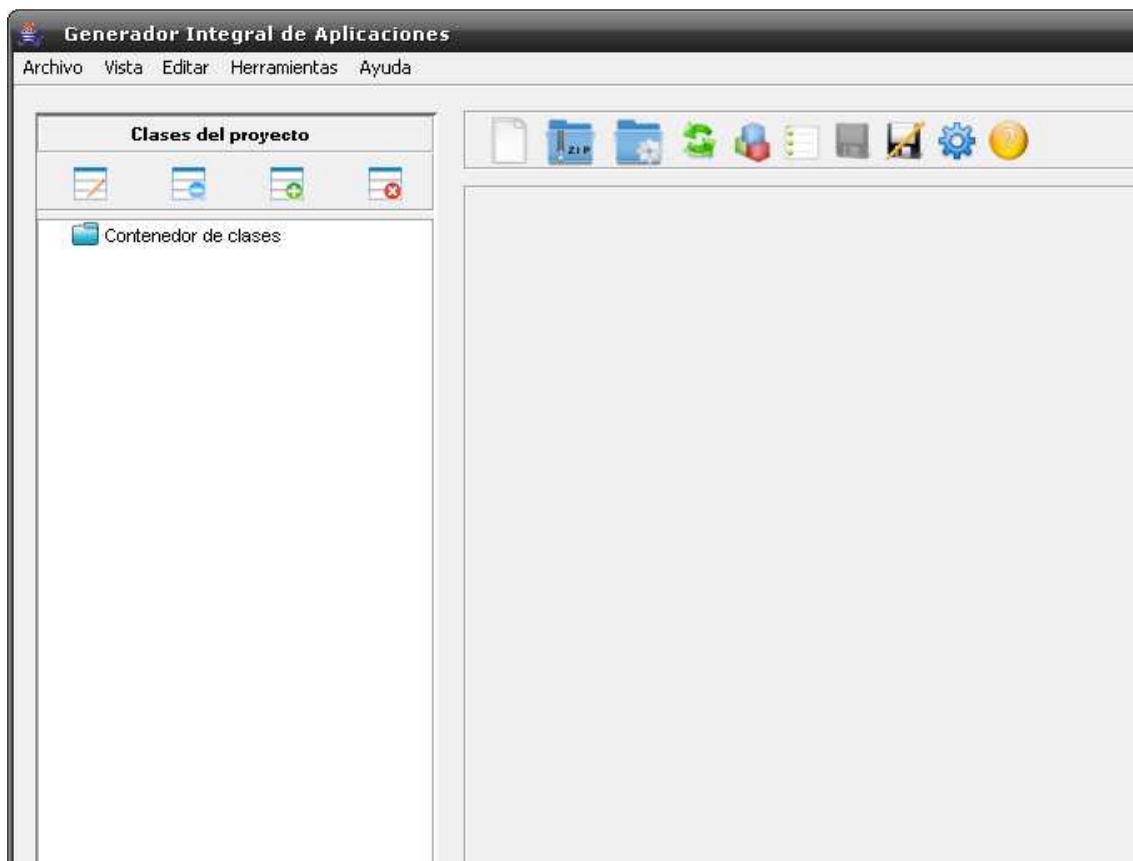
A continuación explicaremos las diferentes acciones que se pueden llevar a cabo tanto en la vista de diseño como en la de entorno de desarrollo java.

5.1 PARTE DE DISEÑO

Desde esta vista se realizan las modificaciones en el modelo de datos creado, este modelo puede haberse cargado desde un archivo XMI versión 1.2, creado previamente con una herramienta de diagramación como ArgoUML, también puede crearse directamente en la aplicación desde cero o abriendo un proyecto ya creado en la herramienta GiA Java que tiene una estructura especial que la herramienta entiende.

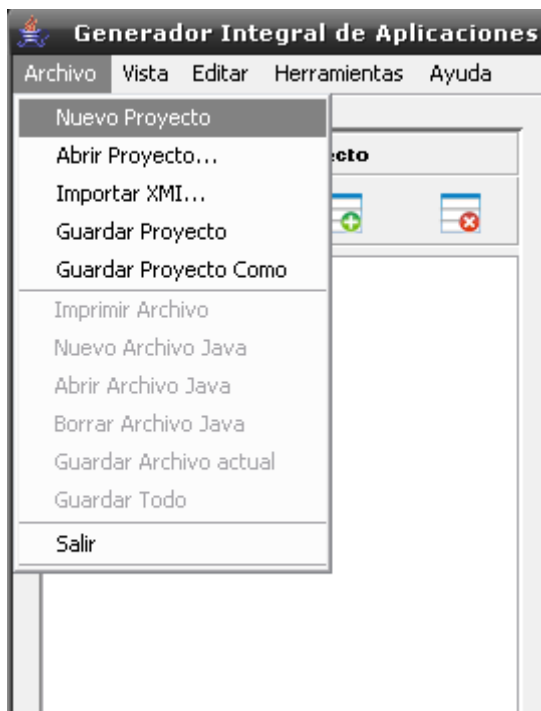
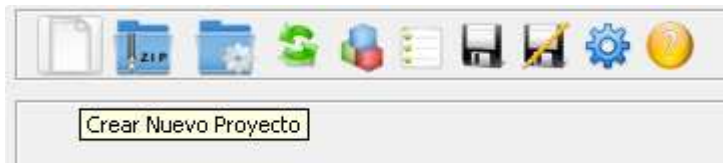
Una vez modificado el modelo, creando un diagrama de clases completo, con clases, atributos y métodos se proseguirá a la generación de código.

A continuación se muestra una captura de pantalla de la vista de Diseño.

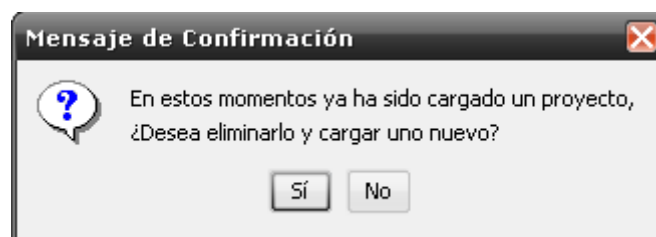


5.1.1 Crear un nuevo Proyecto.

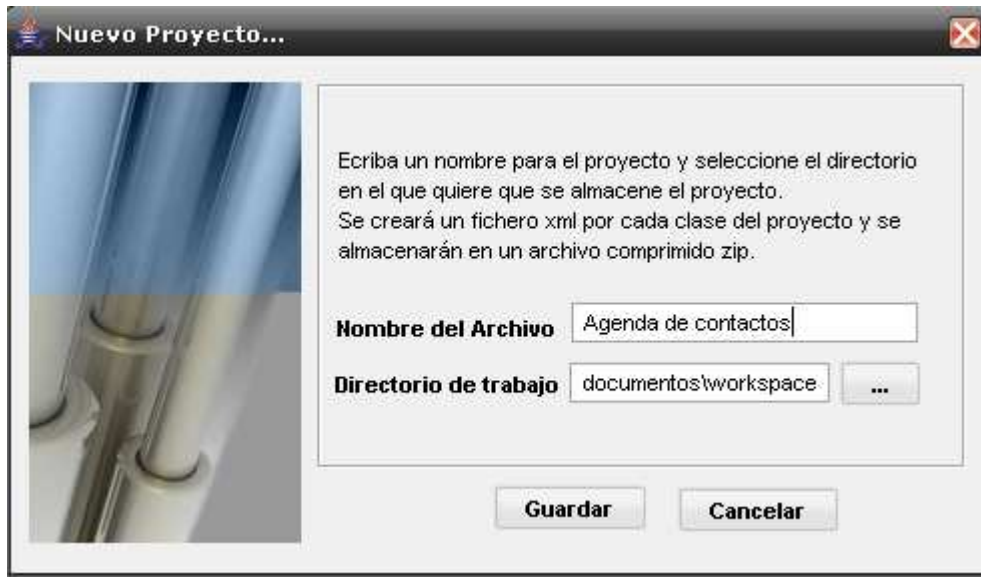
Para crear un proyecto en blanco sobre el que trabajar accederemos al menú Archivo - Nuevo Proyecto, o pulsaremos el botón 'Crear Nuevo Proyecto'.



En caso de que ya se haya creado un proyecto previamente en el espacio de trabajo se preguntará si se desea eliminar para crear uno nuevo. Si se pulsa el botón si, se eliminará el proyecto actual. Para no perder los cambios se debe de guardar el proyecto previamente.



En este momento se abre una ventana con un formulario en el que el usuario podrá introducir el nombre que quiere darle al proyecto, que será el nombre que va a tomar el archivo Zip creado. También se introduce la ruta en la que se almacenarán el archivo Zip a crear.



Cuando pulsamos el botón Aceptar el sistema comprueba que se han rellenado los datos y que son válidos. En caso de que la ruta no exista la crea, y almacena en ella un archivo Zip que contiene en estos momentos:

- Un archivo MANIFEST.txt en el que se ha escrito en su primera línea el nombre del proyecto que le hemos dado al proyecto en este caso 'Agenda de contactos'.
- Un archivo XML principal con el nombre del proyecto 'Agenda de contactos.xml' en el que se enlazarán los ficheros XML creados por cada clase del sistema.

```
<?xml version="1.0" encoding="UTF-8" ?>
<project>
</project>
```

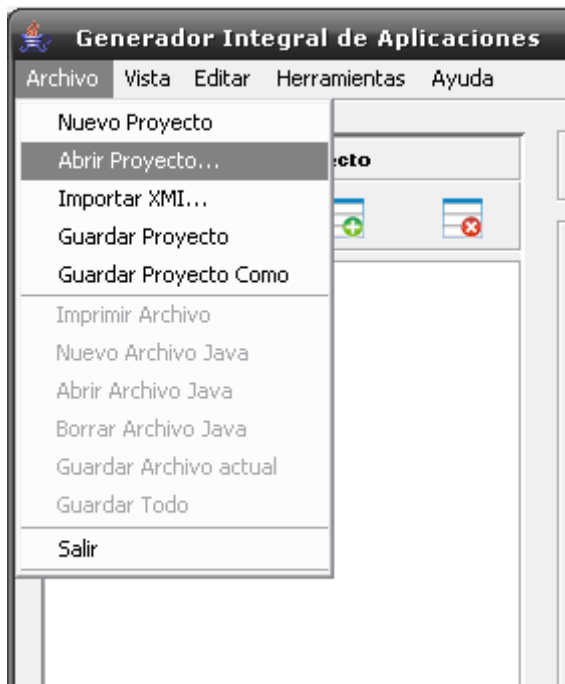
Actualmente se encuentra vacío porque todavía no se han creado clases en el sistema.

Una vez que se añadan clases al sistema y se guarde, automáticamente se creará un fichero XML por cada clase y será añadido al Zip y enlazado desde el XML principal.

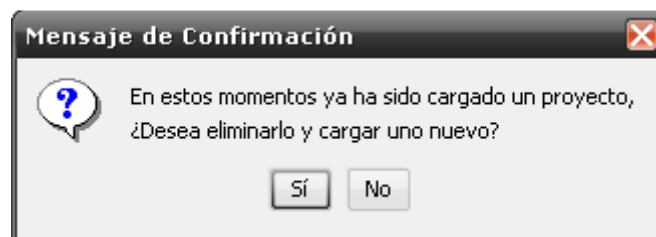
5.1.2 Abrir proyecto

Si ya hemos creado con anterioridad en la herramienta un proyecto, podremos abrirlo nuevamente para su posterior modificación para ello accederemos al menú Archivo – Abrir proyecto o pulsaremos el botón de acceso directo situado en la barra de herramientas.





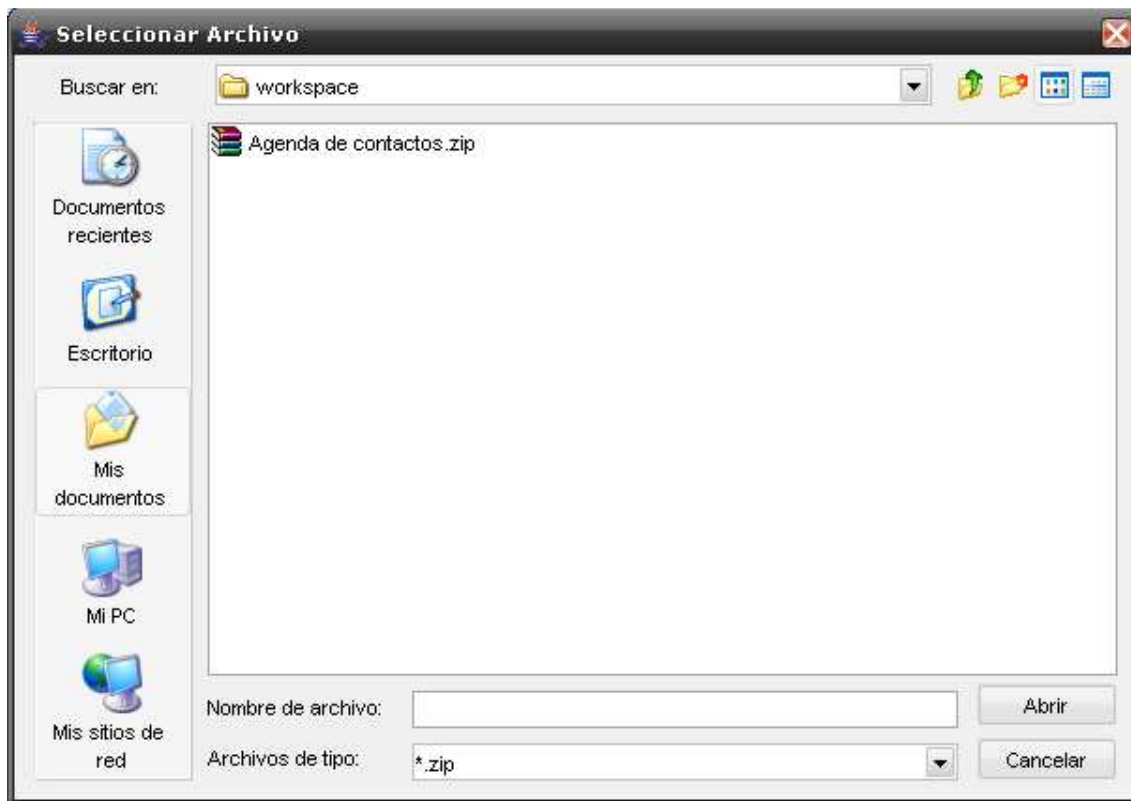
Al igual que cuando creamos un proyecto de cero, si ya existe un proyecto abierto o creado en la herramienta el sistema nos preguntará si deseamos eliminarlo y cargar uno nuevo. Para no perder los cambios deberemos guardarlo previamente.



A continuación se abre una nueva ventana en la que tendremos que añadir la ruta del archivo Zip que queramos abrir.



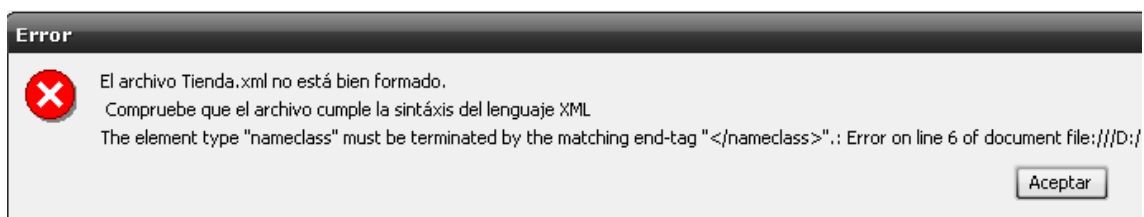
La ruta se puede introducir manualmente o pulsando en el botón de búsqueda de archivos Zip. Este botón filtra por la extensión Zip, lo que permite que solo se visualicen al usuario los archivos que puede elegir como entrada del sistema.



Una vez introducida la ruta y pulsado el botón validar el sistema comprobará que se han rellenado los campos y que la ruta introducida es correcta. En caso de que no lo sea se mostrará el siguiente error:



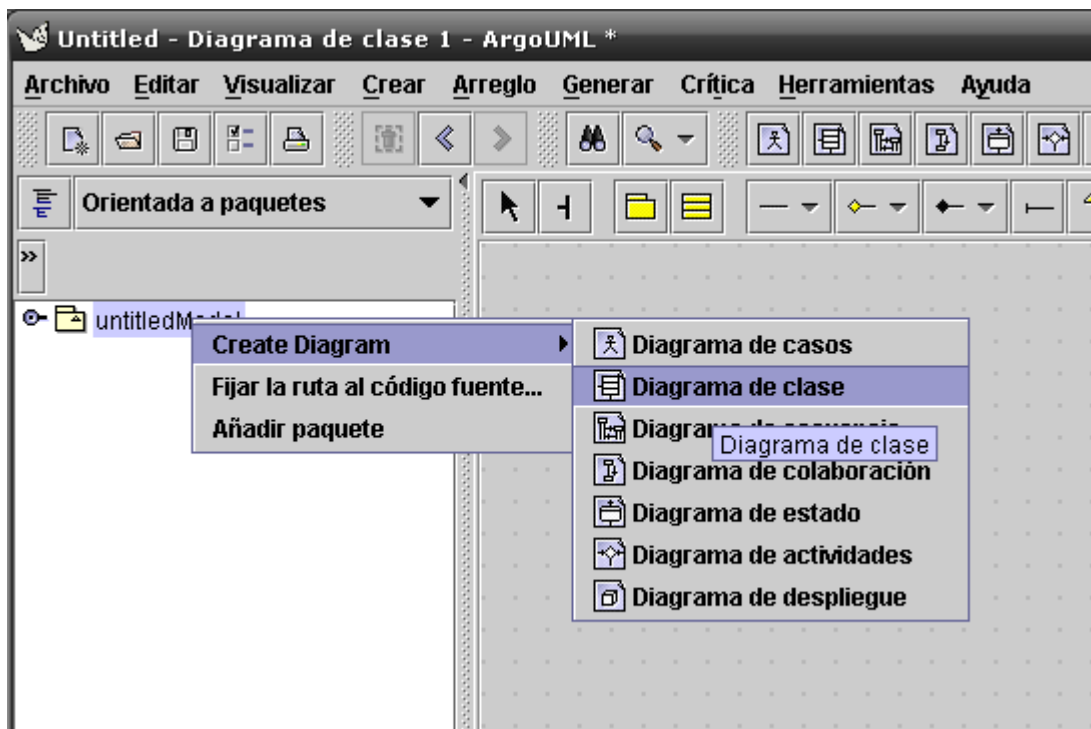
También se valida si cada uno de los archivos XML que forman el Zip están bien formados, en caso de que esto no ocurra el sistema mostrará un error, indicando el fichero, la línea y una descripción del error:



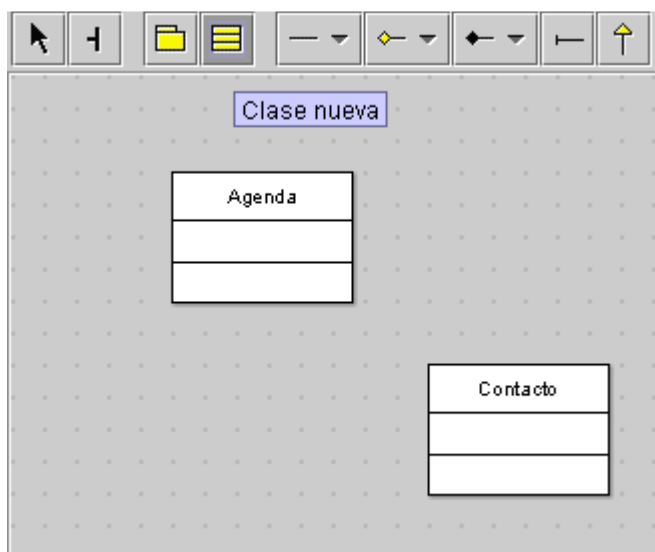
5.1.3 Importar Archivo XMI

Otra posible entrada al sistema es un archivo XMI versión 1.2 creado por una herramienta de diagramación como ArgoUML. A continuación explicaremos el proceso de creación con ArgoUML de un diagrama de clases.

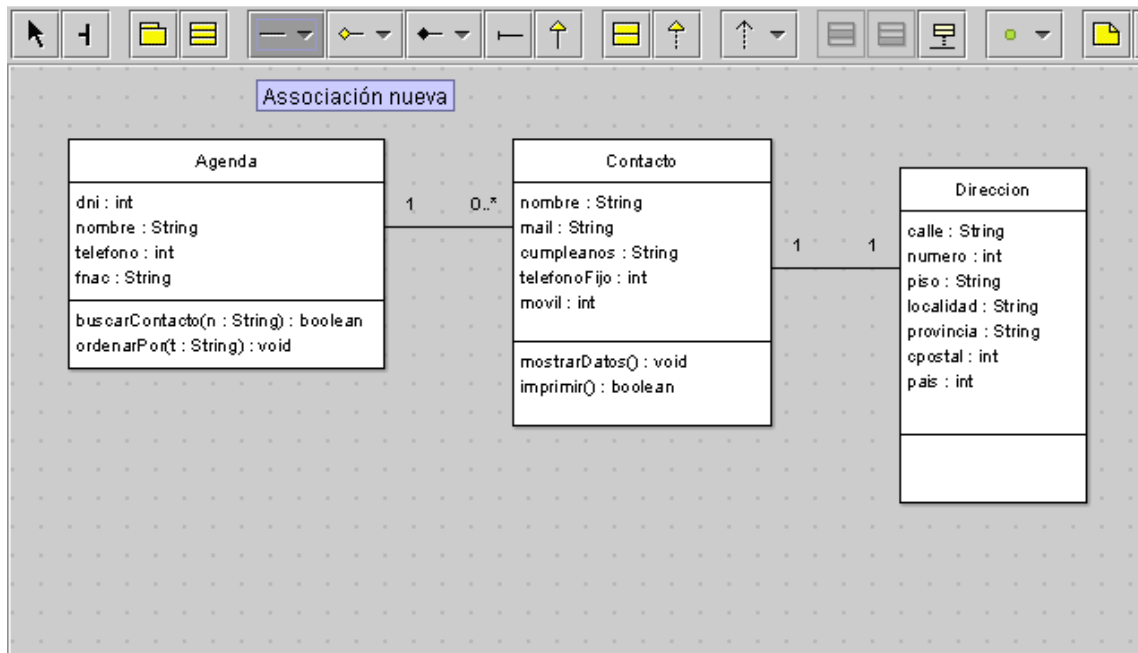
Entramos en la herramienta ArgoUML y se abre automáticamente un espacio de trabajo en el que crearemos un nuevo diagrama de clase.



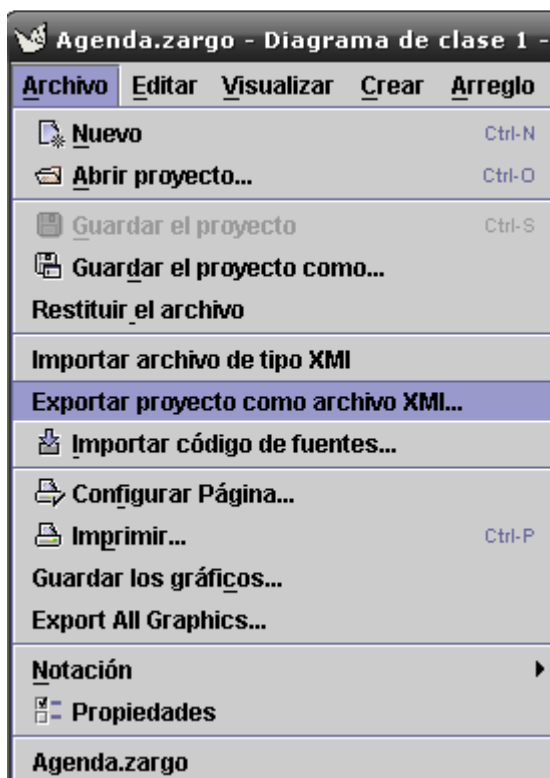
A continuación creamos las clases pulsando el botón 'Clase nueva', los atributos y las operaciones de cada clase se pueden añadir haciendo clic dentro de la clase en su correspondiente cajetín.

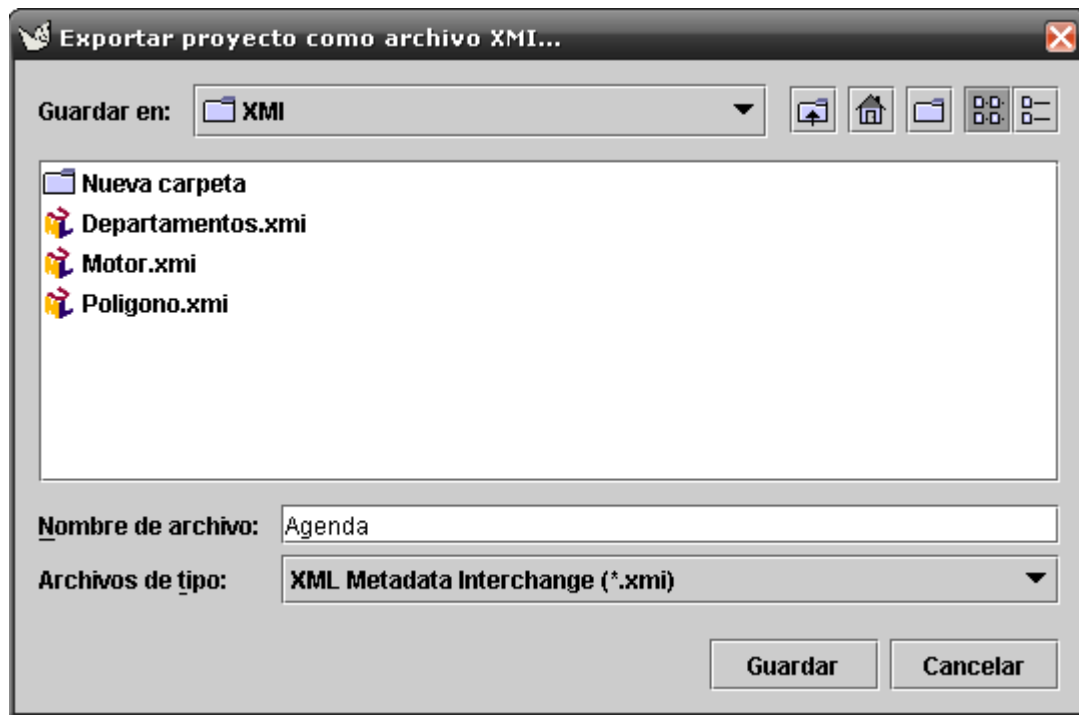


Creamos también las relaciones entre las clases pulsando el botón 'Asociación nueva', por defecto una asociación se crea con multiplicidad 1 a 1 y se edita haciendo clic en el extremo de la asociación.



Una vez creado el diagrama lo exportaremos al formato xmi y le daremos un nombre.

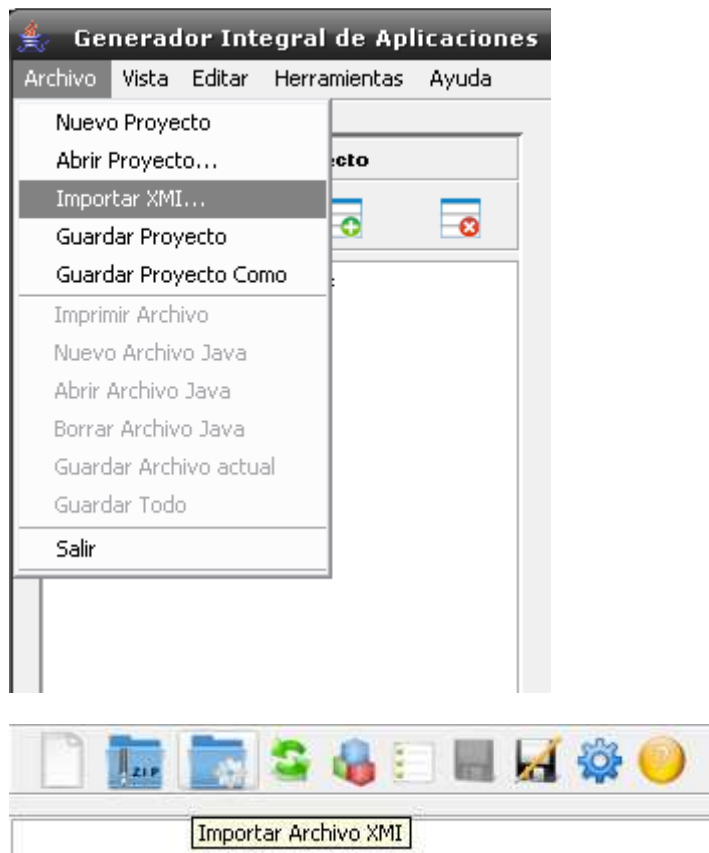




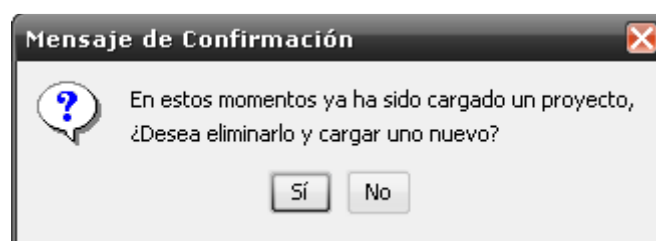
El resultado se puede visualizar y tiene la siguiente estructura.

```
<?xml version = '1.0' encoding = 'UTF-8' ?>
<XMI xmi.version = '1.2' xmlns:UML = 'org.omg.xmi.namespace.UML' timestamp = 'Mon Aug
  <XMI.header>    <XMI.documentation>
    <XMI.exporter>ArgoUML (using Netbeans XMI Writer version 1.0)</XMI.exporter>
    <XMI.exporterVersion>0.24(5) revised on $Date: 2006-11-06 19:55:22 +0100 (Mon, 0
  </XMI.documentation>
  <XMI.metamodel xmi.name="UML" xmi.version="1.4"/></XMI.header>
<XMI.content>
  <UML:Model xmi.id = '10-100-38--23-4866c288:11d1ae9d973:-8000:000000000000077B'
    name = 'untitledModel' isSpecification = 'false' isRoot = 'false' isLeaf = 'fals
    isAbstract = 'false'>
    <UML:Namespace.ownedElement>
      <UML:Class xmi.id = '10-100-38--23-4866c288:11d1ae9d973:-8000:000000000000077C'
        name = 'Agenda' visibility = 'public' isSpecification = 'false' isRoot = 'fa
        isLeaf = 'false' isAbstract = 'false' isActive = 'false'>
        <UML:Classifier.feature>
          <UML:Attribute xmi.id = '10-100-38--23-4866c288:11d1ae9d973:-8000:00000000
            name = 'dni' visibility = 'protected' isSpecification = 'false' ownerSco
            changeability = 'changeable' targetScope = 'instance'>
            <UML:StructuralFeature.multiplicity>
              <UML:Multiplicity xmi.id = '10-100-50-102-a05257d:11d23a72754:-8000:00
                <UML:Multiplicity.range>
                  <UML:MultiplicityRange xmi.id = '10-100-50-102-a05257d:11d23a72754
                    lower = '1' upper = '1' />
                </UML:Multiplicity.range>
              </UML:Multiplicity>
            </UML:StructuralFeature.multiplicity>
          <UML:StructuralFeature.type>
            <UML:DataType xmi.idref = '10-100-38--23-4866c288:11d1ae9d973:-8000:00
          </UML:StructuralFeature.type>
        </UML:Attribute>
        <UML:Operation xmi.id = '10-100-38--23-4866c288:11d1ae9d973:-8000:000000000
          name = 'buscarContacto' visibility = 'public' isSpecification = 'false'
```

Una vez explicado como obtenemos el archivo XMI proseguiremos a importarlo en la herramienta GiAJava, para ello accederemos al menú Archivo – Importar XMI... o bien pulsaremos el botón de la barra de tareas ‘Importar Archivo XMI’.



Si ya existe un proyecto cargado en el espacio de trabajo el sistema muestra el siguiente mensaje de confirmación, como en los casos anteriores debemos guardar si no queremos perder los cambios realizados en el proyecto que está actualmente cargado.



En estos momentos se abre un nuevo formulario en el que se introduce la ruta del archivo XMI que queremos añadir.

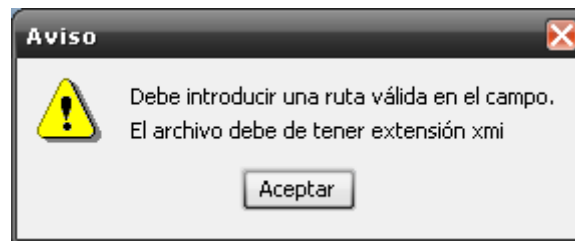


La ruta se puede introducir manualmente o pulsando en el botón de búsqueda de archivos XMI. Este botón filtra por la extensión XMI, lo que permite que sólo se visualicen al usuario los archivos que puede elegir como entrada del sistema.





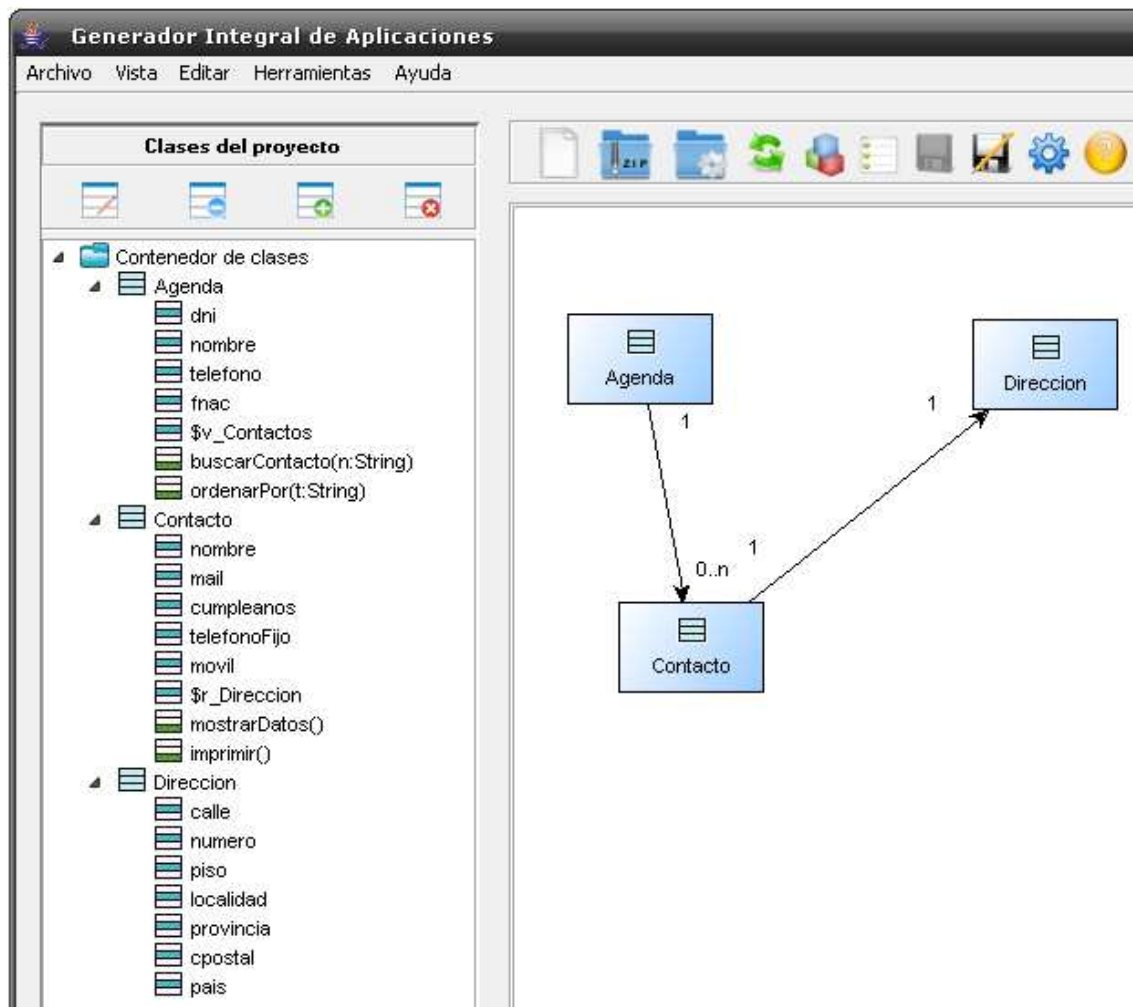
Una vez pulsado el botón aceptar el sistema comprobará que el archivo existe, que tiene extensión XMI.



Si esto es correcto comprobará que el archivo XMI esté bien formado y en caso de que no lo esté mostrará un mensaje de error:



En caso de que no se haya producido ninguno de estos errores el sistema cargará el archivo:

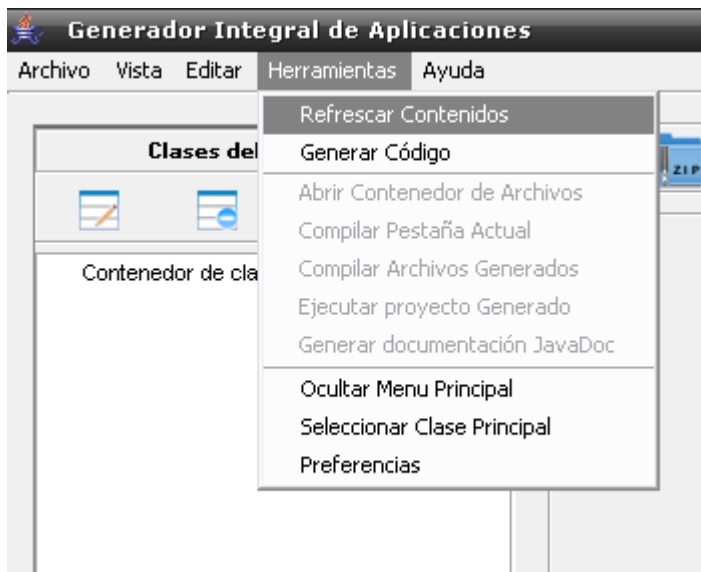


5.1.4 Refrescar contenidos

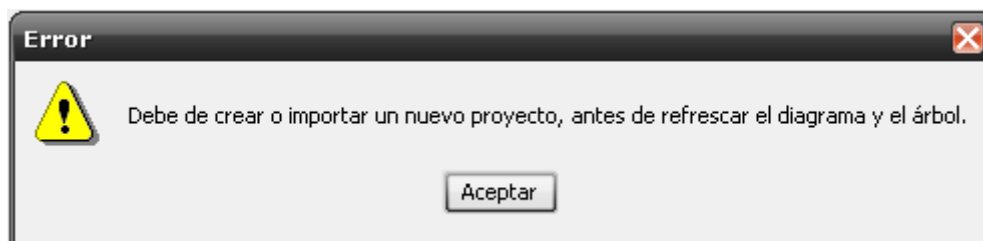
Una vez creado el proyecto podremos hacer uso de la opción refrescar contenidos, esta opción va a volver a cargar el contenido del árbol de clases situado en la parte izquierda y vuelve a pintar el diagrama de clases.

Podremos acceder a esta opción desde el menú Herramientas – Refrescar Contenidos o mediante el botón de la barra de herramientas ‘Refrescar contenidos’.





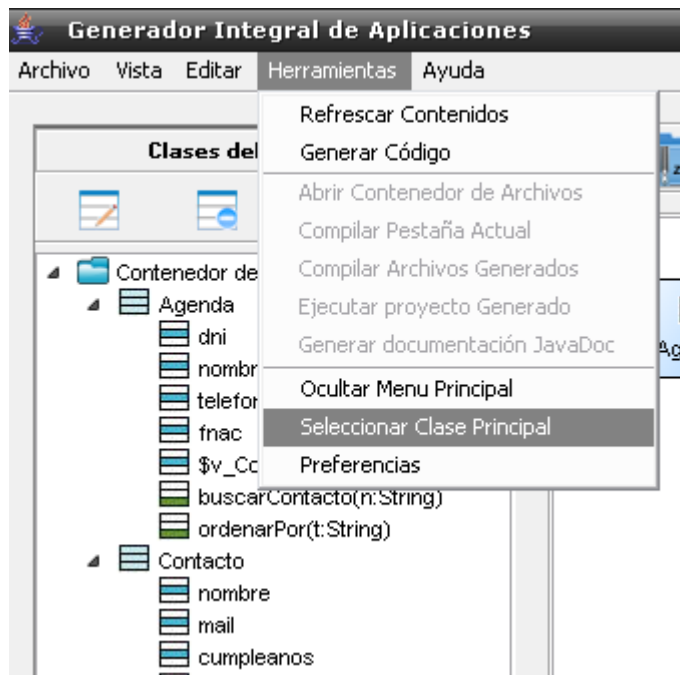
En caso de que no exista ningún proyecto creado y se haya pulsado el botón 'Refrescar contenidos' se mostrará el siguiente mensaje de error.



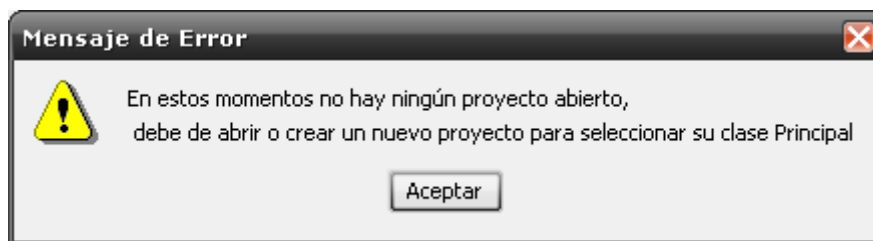
5.1.5 Seleccionar Clase Principal

Previamente a la generación de código se debe de seleccionar la clase principal del sistema. Esta clase es elegida por el usuario y debe ser una clase a la que no llegue ninguna relación. A esta opción se puede acceder desde el menú Herramientas - Seleccionar clase principal, o desde el botón de la barra de herramientas 'Seleccionar clase principal'.

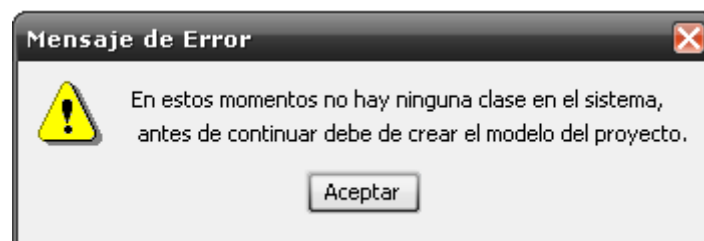




En caso de que no se haya abierto ningún proyecto se mostrará el siguiente mensaje de error:

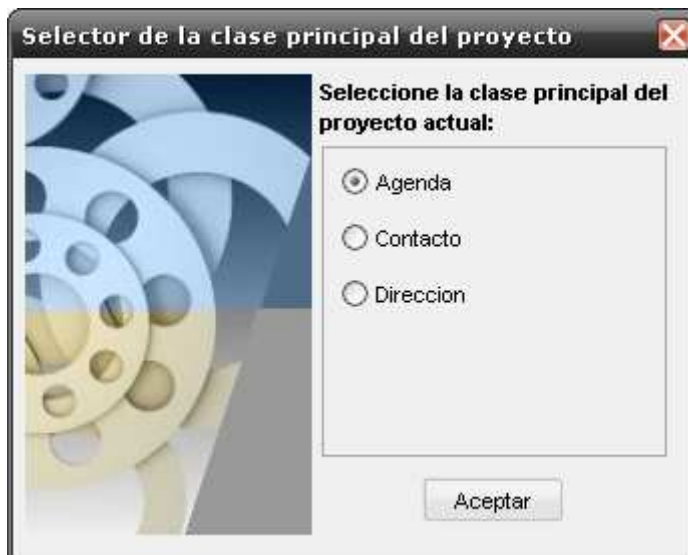


En caso de que no exista ninguna clase en el sistema se mostrará el siguiente mensaje de error:



Como se puede verse en la siguiente imagen, se puede seleccionar cualquiera de las clases existentes en el sistema. Cuando se abre la ventana de selección por primera vez, veremos que se encuentra marcada la clase que el sistema considera como clase principal.

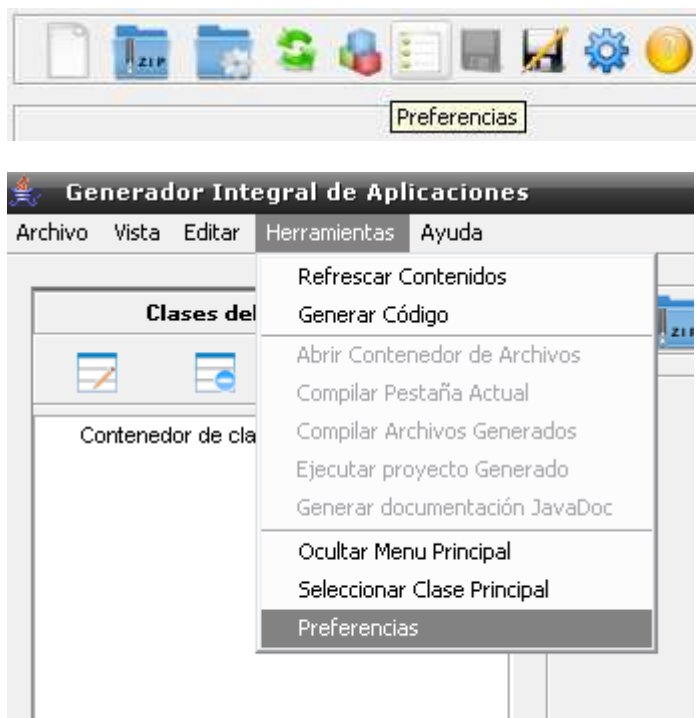
Al pulsar el botón aceptar, el sistema muestra un mensaje en la barra de información situada en la parte inferior del sistema.



Se ha marcado con éxito la clase principal: Agenda

5.1.6 Modificar Preferencias

Continuando con los botones situados en la barra de herramientas podremos acceder al botón de preferencias, o si lo preferimos accederemos desde el menú Herramientas - Preferencias.



Al pulsar este botón se abre un formulario en el que aparecen dos casillas en las que deberemos de rellenar los datos de los directorios por defecto de la aplicación. Estos directorios son:

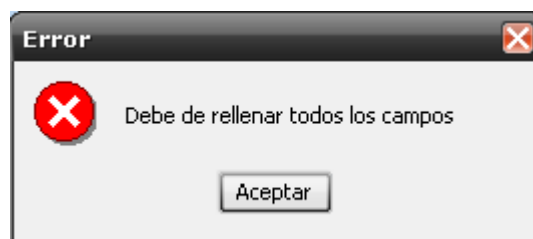
- Directorio de trabajo: en él se almacenará el archivo Zip creado por el sistema cuando se pulse la opción guardar como.
- Directorio java: en él se almacenarán los archivos java generados.

Los directorios se pueden introducir manualmente o mediante el botón '...' que filtra los ficheros por directorio.

Estos directorios son considerados como directorios por defecto, cuando pulsamos el botón guardar de esta ventana el sistema no valida que las rutas sean correctas, porque estas rutas aparecerán en la ventana guardar como y en la ventana generar código, que es donde se utilizarán y ahí si se validan.



En caso de que alguno de las casillas no se haya rellenado el sistema mostrará el siguiente mensaje de error:



Al pulsar el botón aceptar, el sistema muestra un mensaje en la barra de información situada en la parte inferior del sistema.

Se han almacenado correctamente los directorios por defecto.

5.1.7 Guardar Proyecto

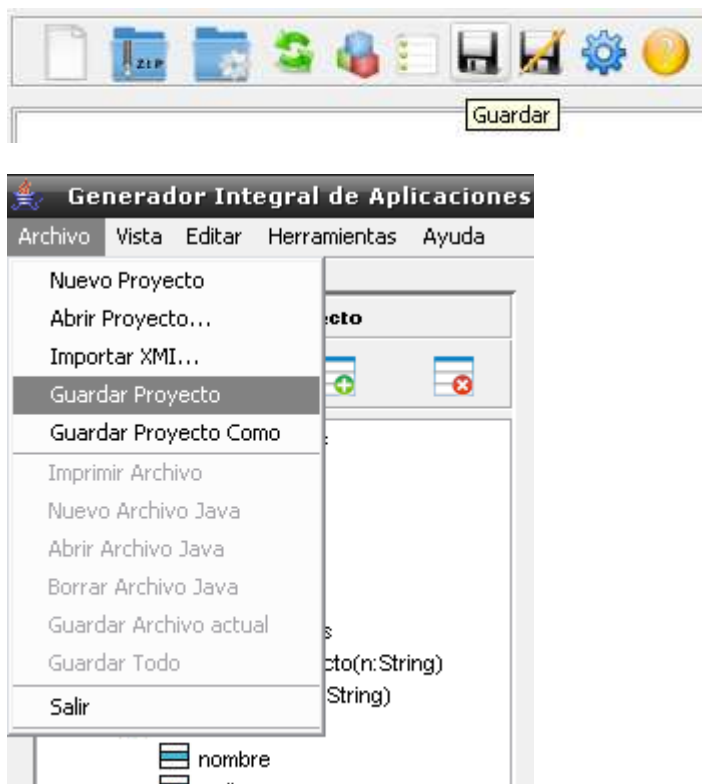
La opción de guardar proyecto sólo aparecerá habilitada en caso de que el sistema ya haya guardado el archivo Zip previamente, los posibles casos para que esto ocurra son los siguientes:

- Cuando el usuario haya pulsado la opción 'Guardar como' antes de la opción guardar. El sistema entenderá que el usuario desea guardar los cambios en el directorio que se haya indicado en la opción 'Guardar como'

- Cuando se haya creado un 'Nuevo Proyecto' desde cero, en este caso existe un archivo Zip al que se le da nombre y ubicación en la opción nuevo proyecto. El sistema sobrescribirá el proyecto con el nombre y la ubicación que se le ha dado en la opción 'Nuevo proyecto'.

- Cuando se haya abierto un proyecto Zip ya creado con anterioridad mediante la opción 'Abrir Proyecto'. En el momento en el que se abre un nuevo proyecto tenemos la ubicación del archivo Zip en el que se almacenará el proyecto.

Para acceder a la opción guardar podremos pulsar el botón 'Guardar' de la barra de herramientas o podemos ir a través del menú Archivo - Guardar Proyecto.



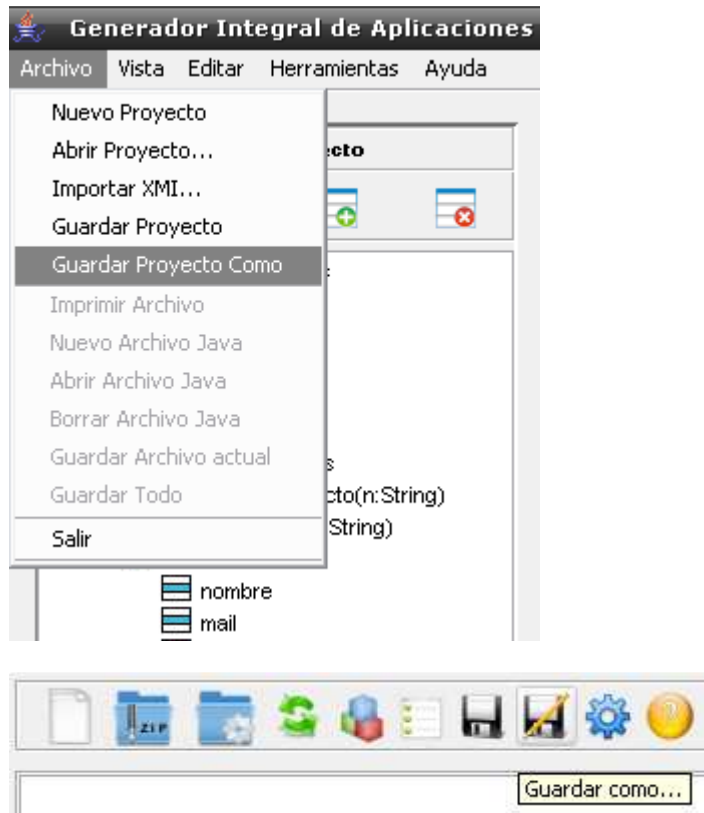
Si el sistema ha guardado correctamente el proyecto se mostrará el siguiente mensaje:

Se ha guardado correctamente el proyecto en el directorio: D:\Workspace\Proyecto_Agenda

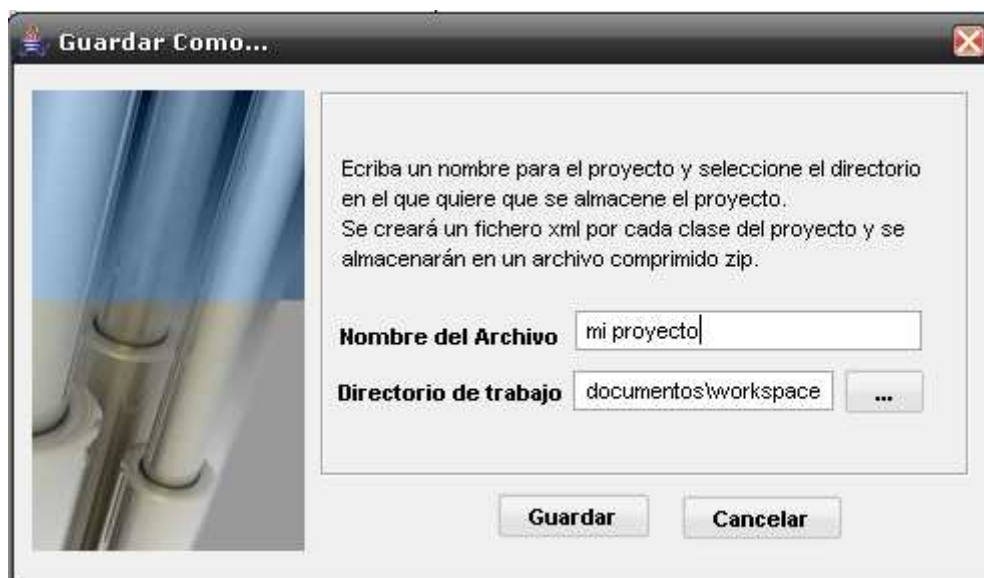
En disco se ha creado un archivo Zip y se exportado una imagen en formato png con el diagrama actual situándola en la misma carpeta que el Zip.

5.1.8 Guardar Proyecto como

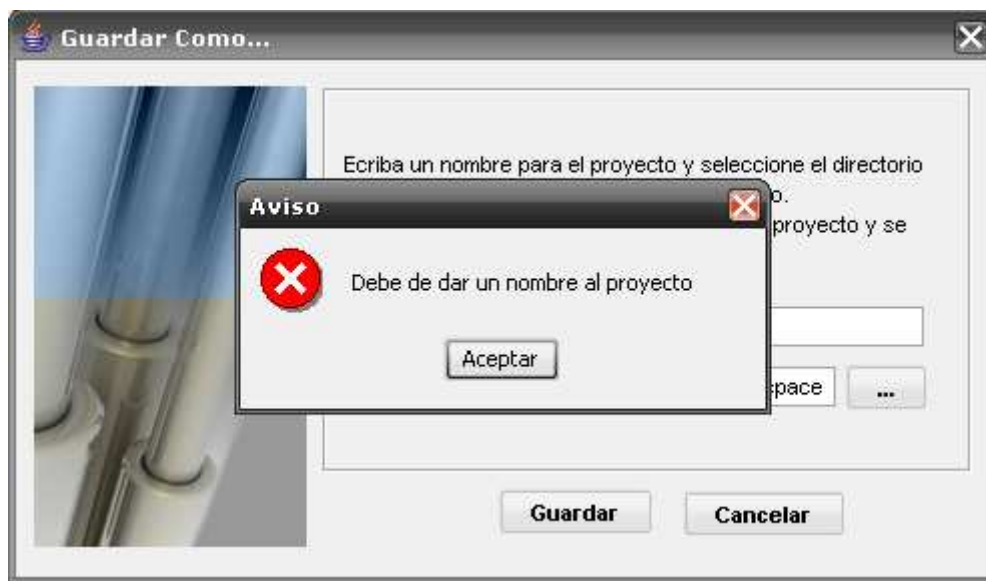
A diferencia de la opción guardar, si deseamos escoger la ubicación y el nombre que queremos darle al proyecto deberemos acceder a la opción guardar como mediante el menú Archivo - Guardar Proyecto Como o mediante el botón 'Guardar como' de la barra de herramientas.



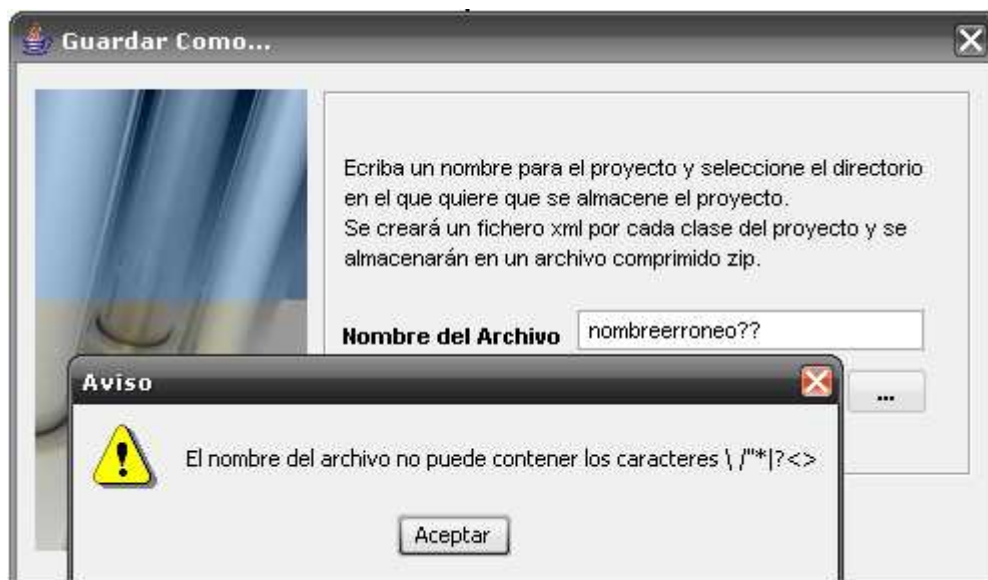
Se abre un formulario en el que daremos un nombre al proyecto. El campo del directorio de trabajo aparece relleno con el directorio de trabajo por defecto introducido en la ventana de preferencias. Este directorio se puede modificar manualmente o a través del botón '...' situado en esta ventana.

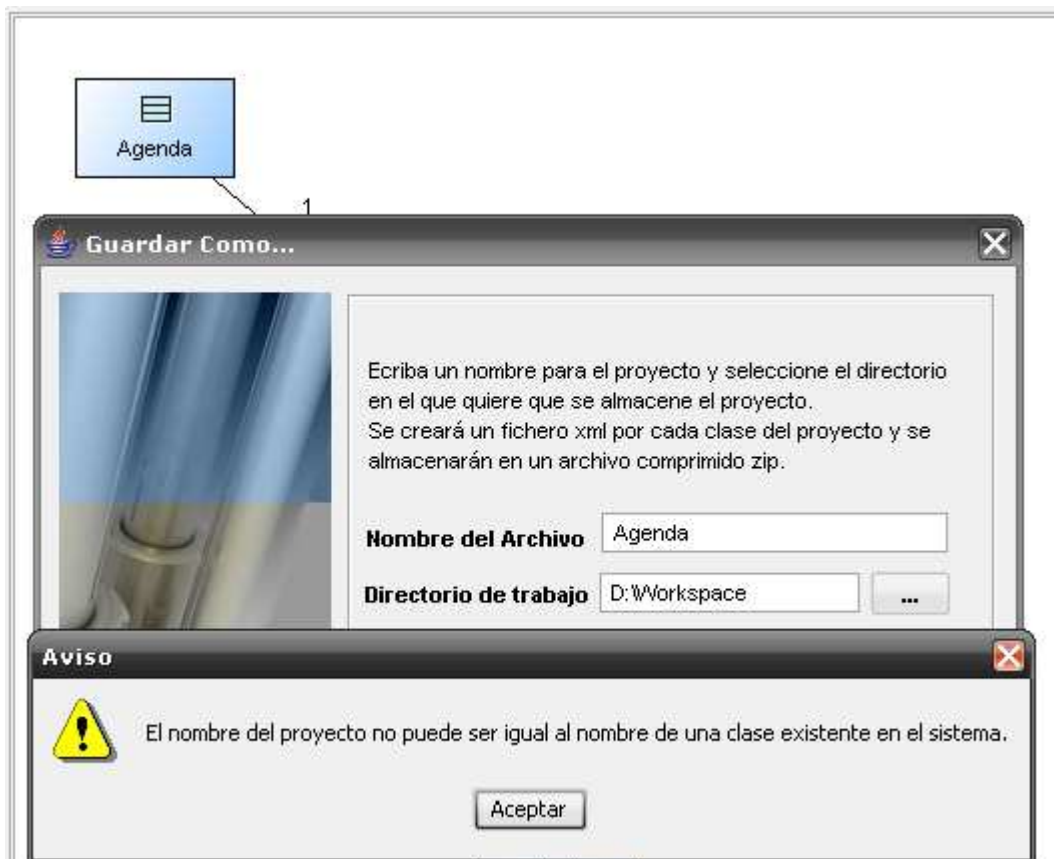


En caso de que no se introduzca un nombre para el proyecto el sistema mostrará el siguiente error:



Si se rellenan ambos campos y se pulsa el botón guardar el sistema comprueba que el nombre que se le quiere dar al proyecto no contiene caracteres inválidos y que no se corresponde con el nombre de ninguna clase. En caso de que esto ocurra muestra los siguientes errores:





También se valida que el directorio exista, en caso de que no exista se mostrará el siguiente error:



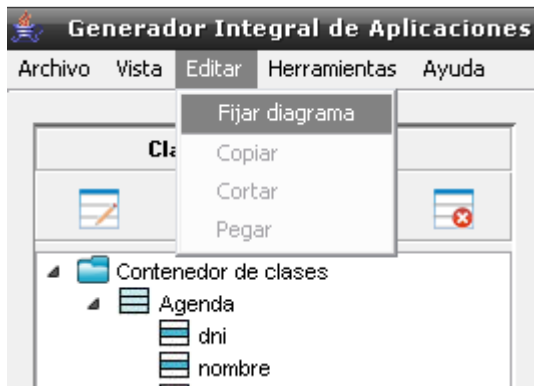
Si no se produce ningún error se muestra el siguiente mensaje en la barra de información situada en la parte inferior de la aplicación.

Se ha guardado correctamente el proyecto en el directorio:D:\Workspace

5.1.9 Fijar Diagrama

En cualquier momento el usuario va a poder fijar el diagrama actual, esta opción hace que el diagrama de clases no sea editable por el usuario directamente, es decir no se van a poder mover las clases de lugar.

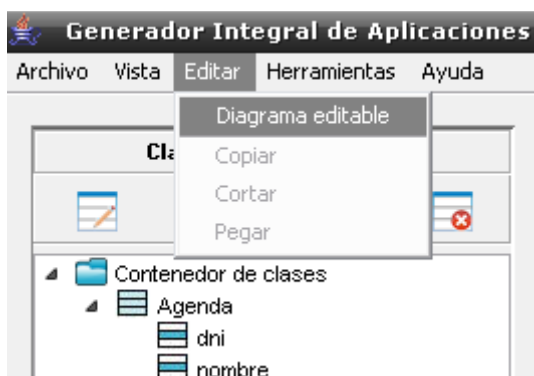
Para acceder a esta opción se podrá escoger el menú Editar - Fijar diagrama.



Si no existe ningún proyecto creado el sistema mostrará el siguiente mensaje:

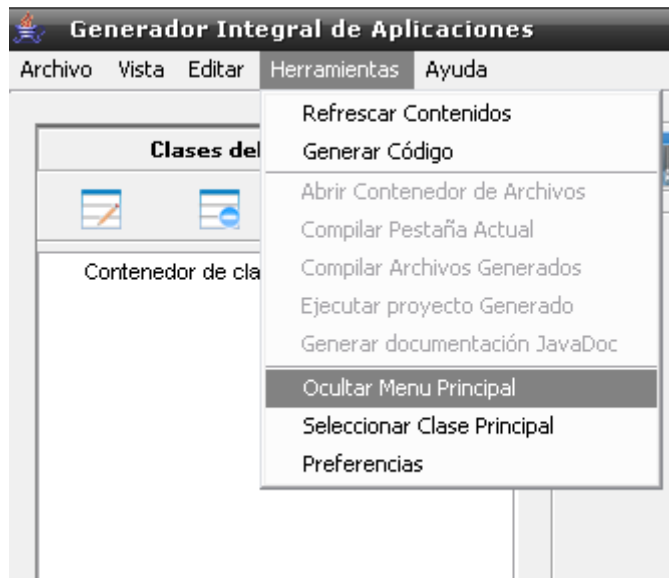


En el momento en el que se fija el diagrama se podrá volver a poner editable bien modificando el diagrama al añadir una nueva clase o bien desde el menú Editar - Diagrama editable.

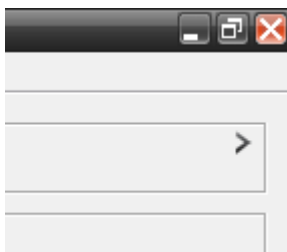


5.1.10 Ocultar Menú Principal

Si se desea tener un mayor espacio tanto en el área del diagrama como en el árbol lateral de clases el sistema el usuario podrá ocultar el menú principal, para ello podrá hacer clic en el menú Herramientas - Ocultar Menú principal.

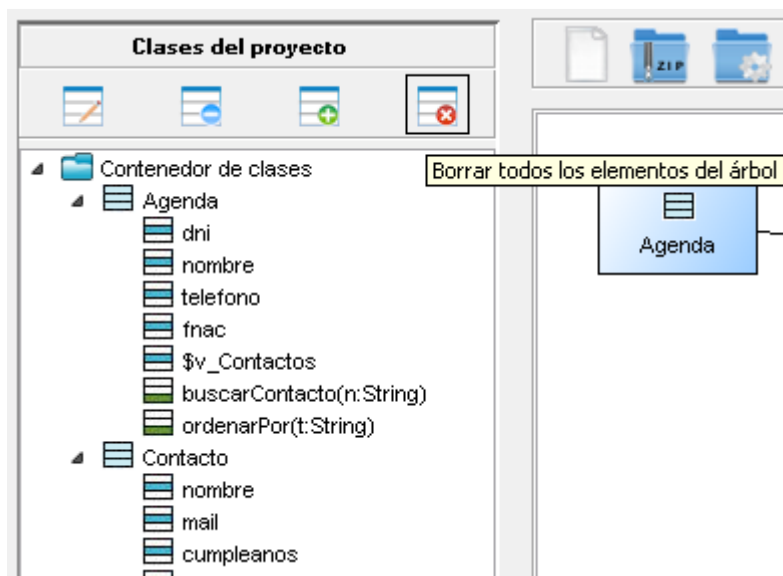


Para volver a mostrar el menú haremos clic en el botón situado en la esquina izquierda de la aplicación.

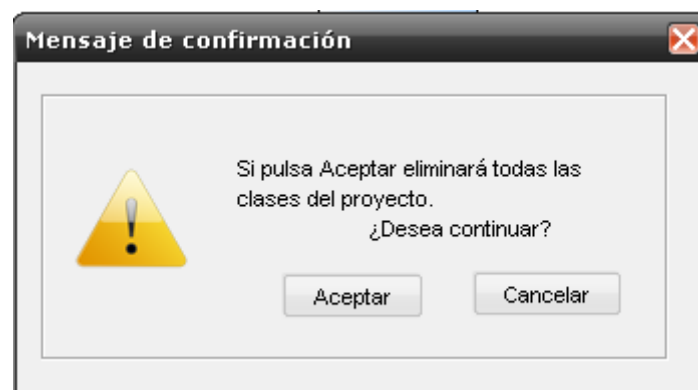


5.1.11 Borrar todos los elementos

Si se desean eliminar todos los datos del árbol de clases, es decir todas las clases con sus atributos y métodos se puede hacer uso del botón borrar todos los elementos del árbol.



La aplicación mostrará un mensaje de confirmación y tras aceptar este mensaje el árbol se presentará vacío así como el diagrama de clases estará en blanco.

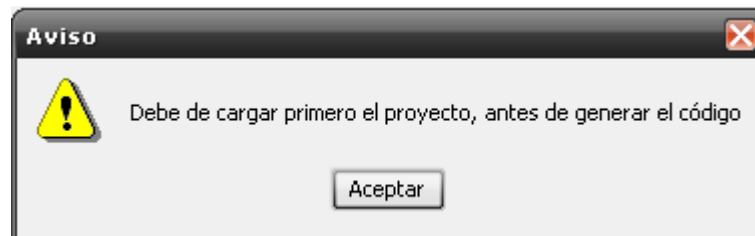


5.1.12 Generar código

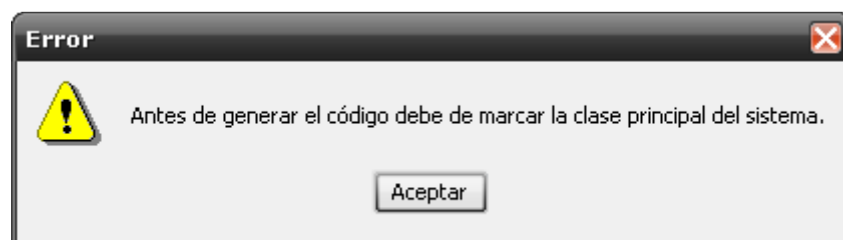
Para generar el código java a partir del modelo debemos de acceder al menú Herramientas- Generar código o pulsar el botón 'Generar código' situado en la barra de herramientas.



En caso de que no se haya creado ningún proyecto el sistema mostrará el siguiente mensaje de error.

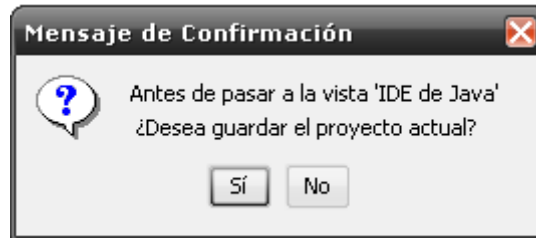


Si existe un proyecto abierto en la aplicación, se comprueba si se ha marcado o no la clase principal. En caso de que no se haya marcado la clase principal se muestra la siguiente advertencia y automáticamente se abre la ventana 'Selector de clase principal'.



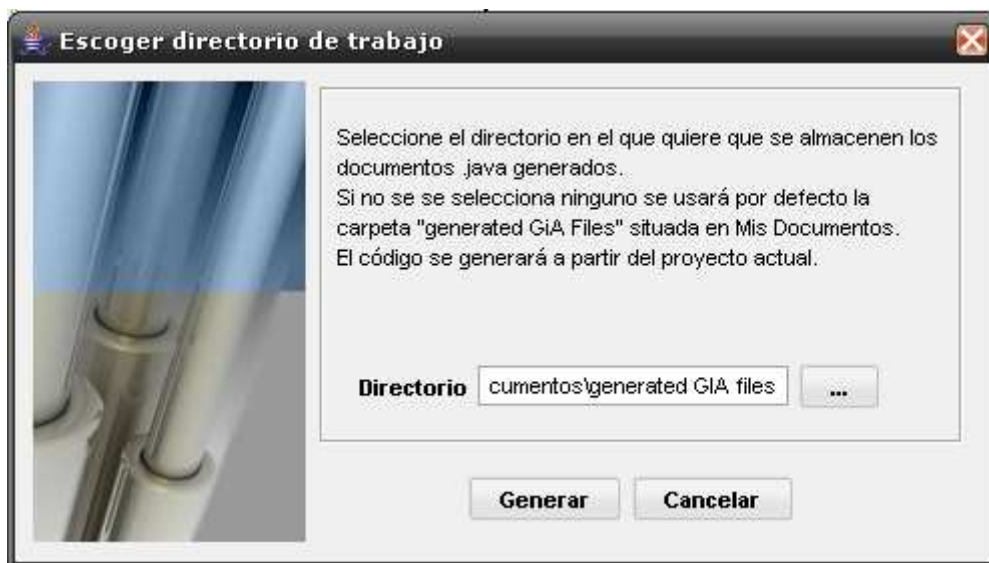
Si no se produce ningún error el sistema le pregunta al usuario si desea guardar el proyecto actual antes de pasar a la vista de 'IDE de java'. Si el usuario decide guardar los

cambios tendrá una versión en disco del proyecto hasta ese momento. De todas maneras aunque el usuario decida seguir adelante sin guardar los cambios realizados en el modelo, podrá volver a la vista de diseño en cualquier momento (si no ha salido de la aplicación) y recuperar el modelo que había hasta ese momento en la herramienta.



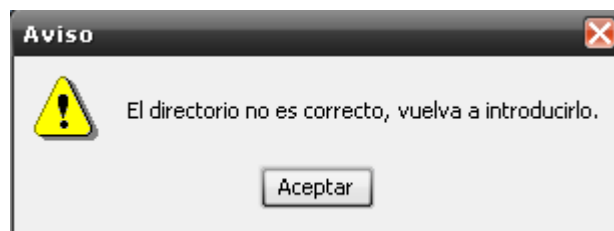
Si el usuario decide guardar los cambios el sistema comprueba si existe un archivo Zip sobre el que realizar los cambios, si es así el sistema guardará las modificaciones sobre el mismo. Si no existe ningún archivo Zip el sistema muestra la ventana 'Guardar como...' para que el usuario introduzca una ruta y de un nombre al proyecto.

A continuación el sistema muestra un formulario para introducir un directorio en el que guardar los archivos java generados. El formulario aparece rellenado con el directorio por defecto que se haya introducido desde la ventana de 'Preferencias'.

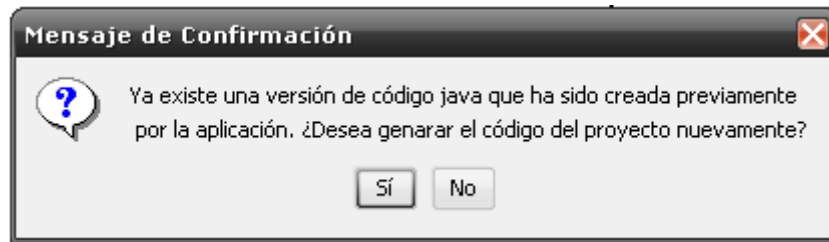


Una vez introducida la ruta el sistema comprueba si la ruta no contiene espacios y si es una ruta válida. En caso de que no exista la ruta pero que sea válida el sistema la creará.

Si la ruta no es válida el sistema mostrará el siguiente mensaje:



En caso de que no se produzca ningún error el sistema verifica si ya existe un archivo con el mismo nombre en esa dirección en caso de que exista muestra el siguiente mensaje:

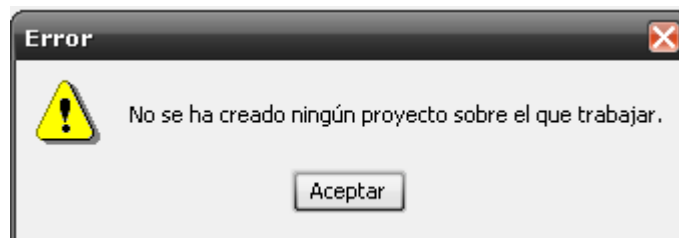


Si el usuario pulsa la opción sí el sistema eliminará todos los archivos existentes en ese directorio y creará los nuevos archivos de código java.

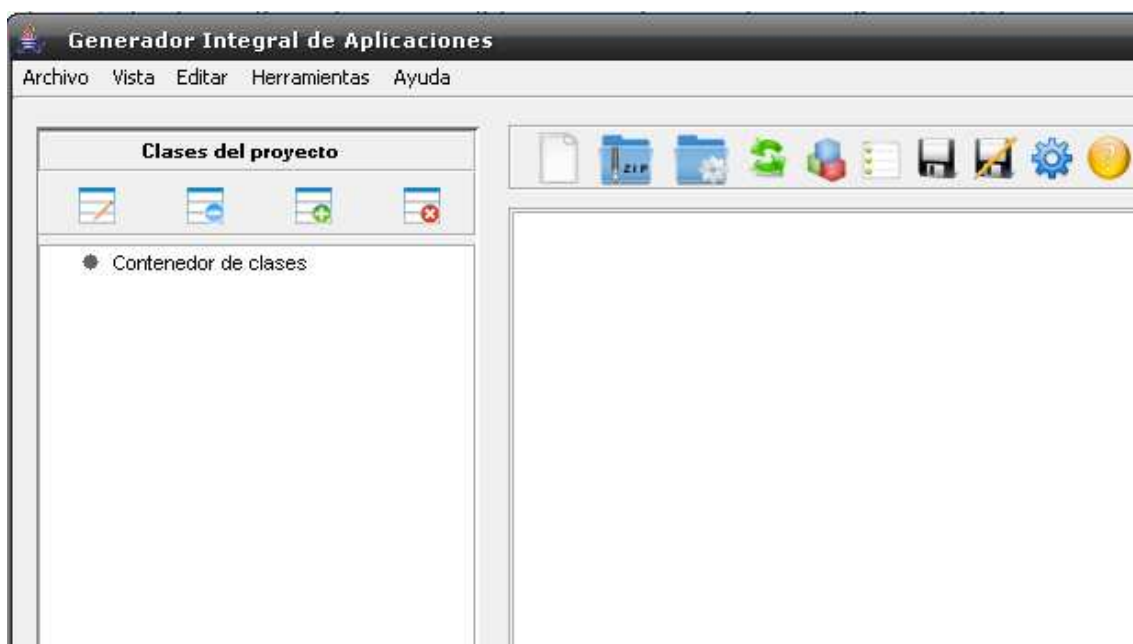
5.1.13 Gestionar Clase

A continuación se expondrán todas las acciones relacionadas con la gestión de clases, atributos, métodos y relaciones.

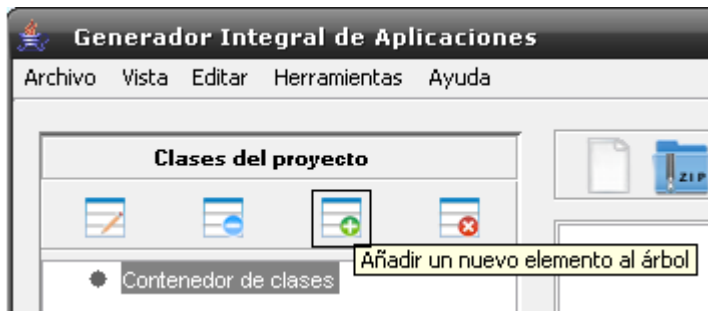
Si antes de acceder a cualquiera de las opciones que se muestran a continuación no se ha creado ningún proyecto el sistema mostrará el siguiente mensaje de error:



Añadir una clase



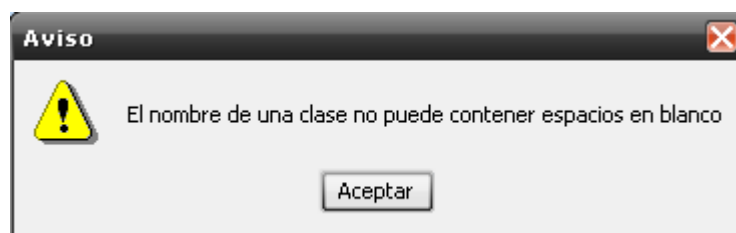
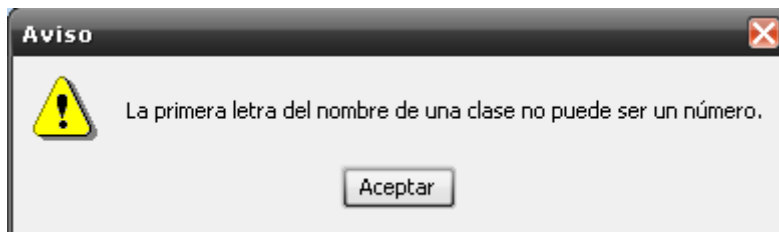
En la imagen superior podemos comprobar que el árbol de clases está vacío y que por tanto no existe ninguna clase en el sistema.

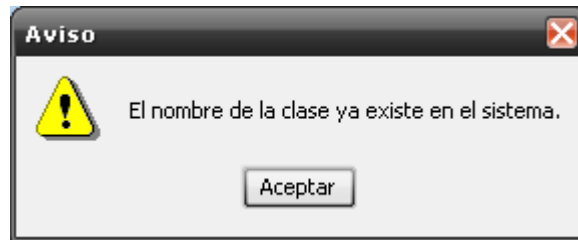


Para añadir un nuevo elemento al árbol debemos de seleccionar la raíz del árbol y a continuación pulsar el botón 'Añadir un nuevo elemento al árbol'. Se abrirá una nueva ventana en la que podremos introducir los datos de la clase. Tenemos que tener en cuenta que el nombre de una clase no puede empezar por un número y no puede contener espacios.

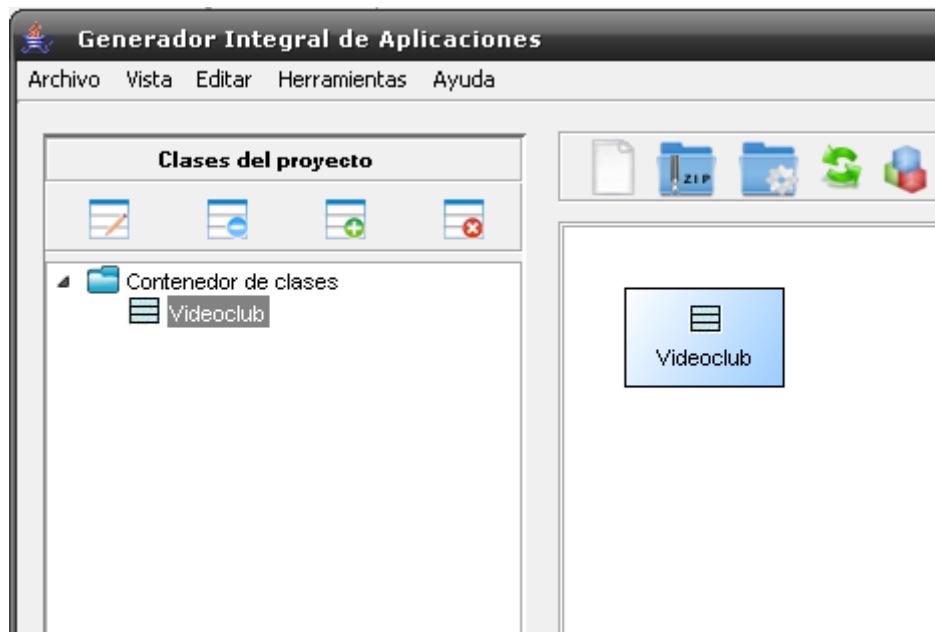


Si se introduce un nombre incorrecto para la clase el sistema mostrará alguno de los siguiente errores:



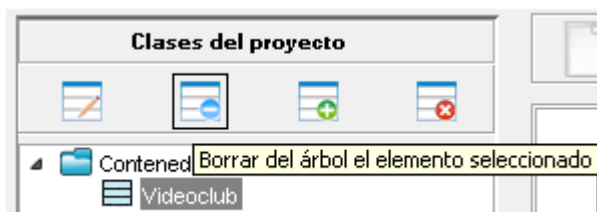


En caso de que no se produzca ningún error la clase se añadirá al árbol y al diagrama.

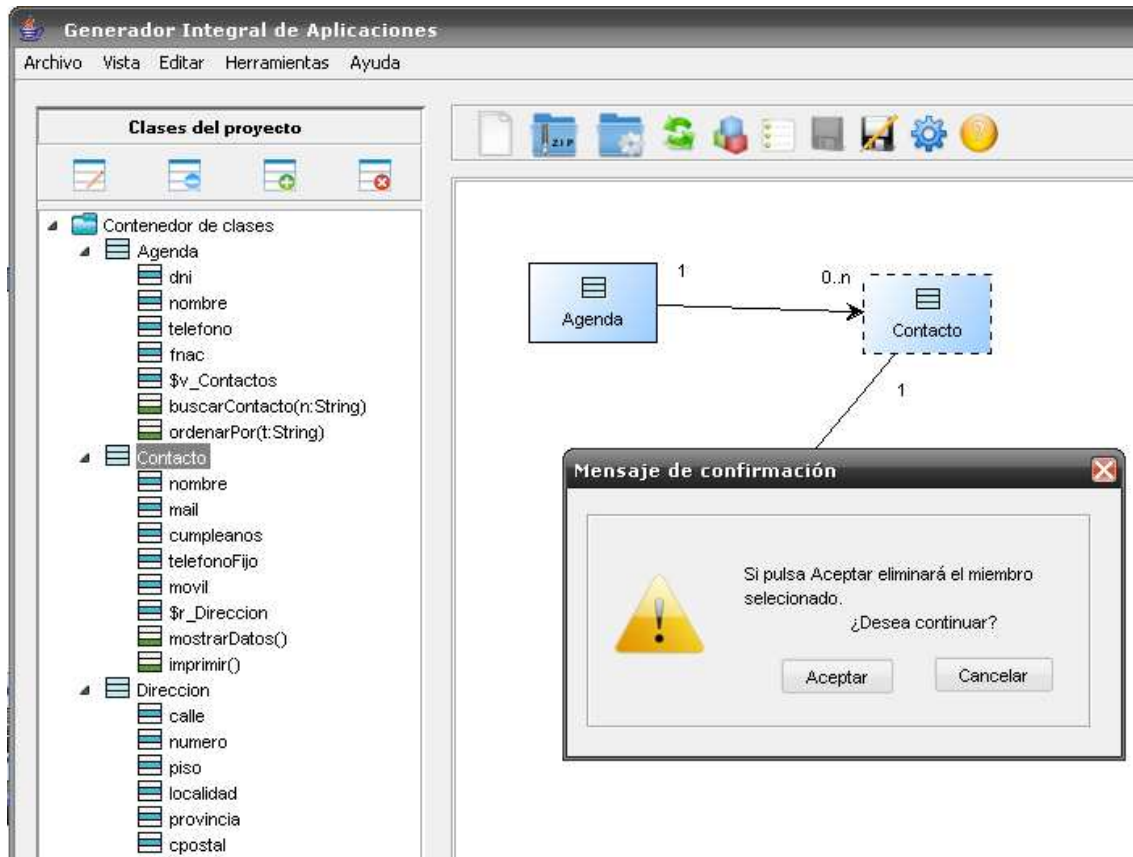


Borrar Clase

Si deseamos borrar una clase deberemos hacer clic en su nombre en el árbol de clases situado en la parte izquierda y pulsar el botón 'Borrar del árbol el elemento seleccionado'.

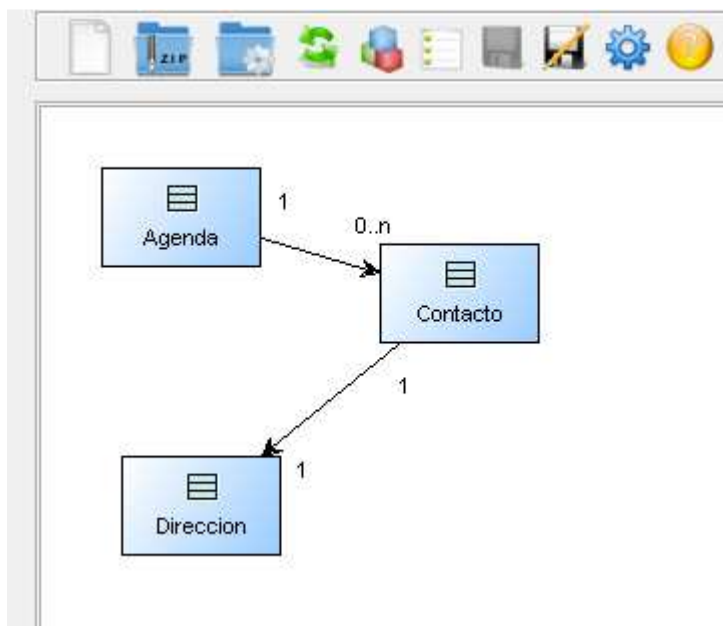


Si se decide eliminar una clase se eliminan todos sus atributos y métodos. En el ejemplo siguiente veremos como al eliminar la clase Contacto con ella se eliminan a parte de sus atributos y métodos las relaciones que tiene esa clase con otras clases del sistema.



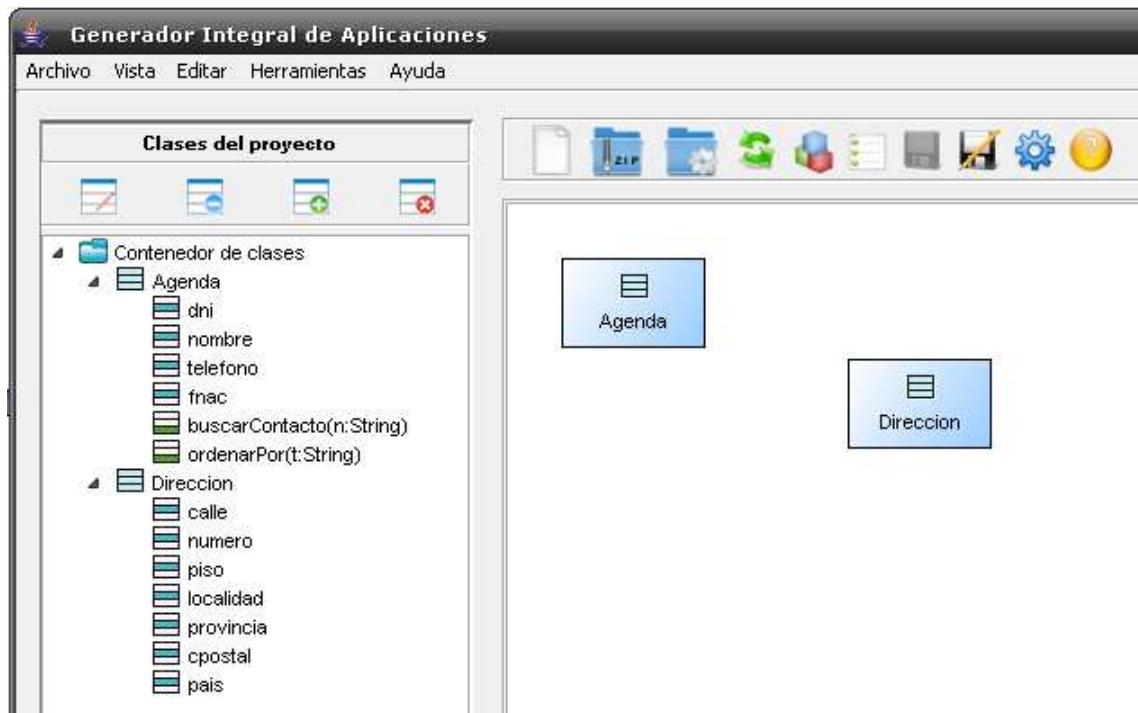
Como podemos ver en el ejemplo la clase Agenda está relacionada con la clase Contacto. Una agenda tiene n contactos. Y la clase Contacto está relacionada con la clase Dirección. Si aceptamos el mensaje superior el modelo cambiará completamente.

Este es el diagrama de clases antes de realizar el borrado del elemento contacto.



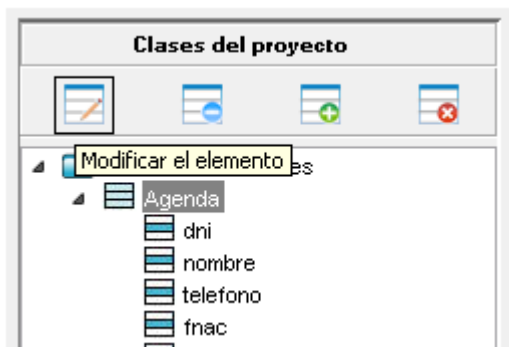
Después de borrar el elemento Contacto podemos comprobar que se ha borrado la clase del diagrama y del árbol y que además han desaparecido las relaciones de asociación del

diagrama. Además se puede comprobar que en la imagen que también se ha eliminado el atributo \$v_Contactos de la clase agenda que era el atributo que creaba la relación entre la clase Agenda y Contacto.

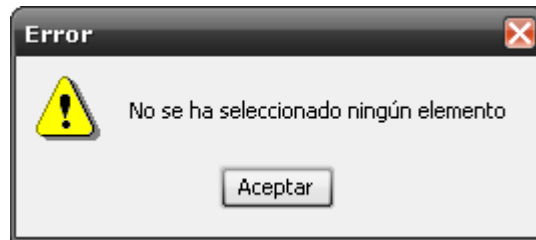


Modificar Clase

Si se desea modificar alguna clase del sistema en primer lugar se selecciona el nombre de la clase del árbol de clases y luego se pulsa el botón 'Modificar el elemento'.



Si se pulsa el botón modificar sin haber seleccionado ninguna clase, el sistema muestra un mensaje de error.



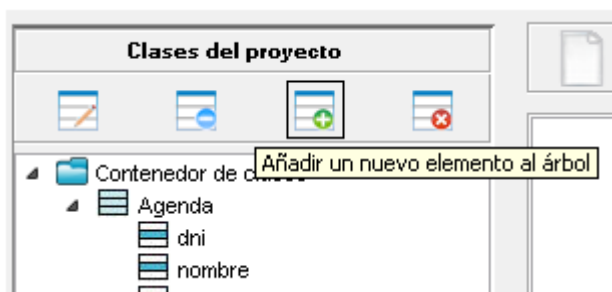
Si no se produce ningún error se abrirá un formulario desde el cual podremos modificar los datos de una clase. Las validaciones que se hacen después de pulsar el botón aceptar son las mismas que cuando se realiza el alta de una clase, por lo tanto los errores que se pueden producir son los mismos.



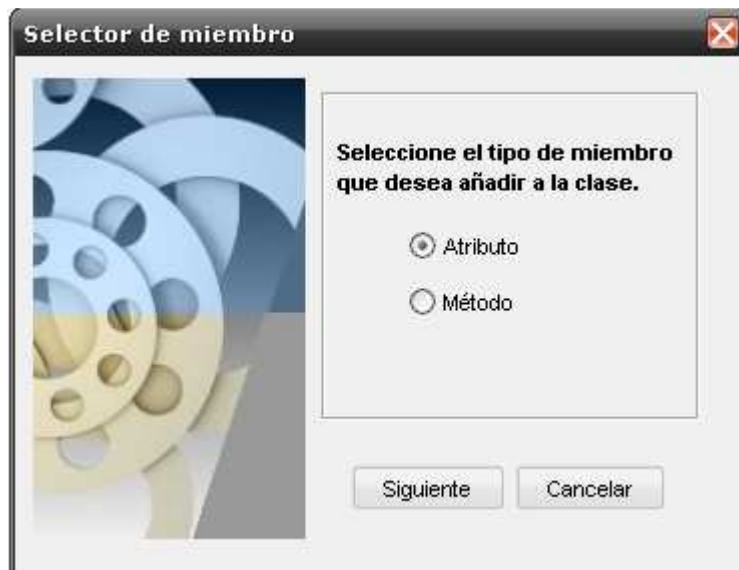
5.1.14 Gestionar Atributos

Añadir Atributo

Para añadir un atributo debemos seleccionar el nombre de la clase a la cual queremos añadir el atributo. Y a continuación pulsar el botón 'Añadir un nuevo elemento al árbol'.



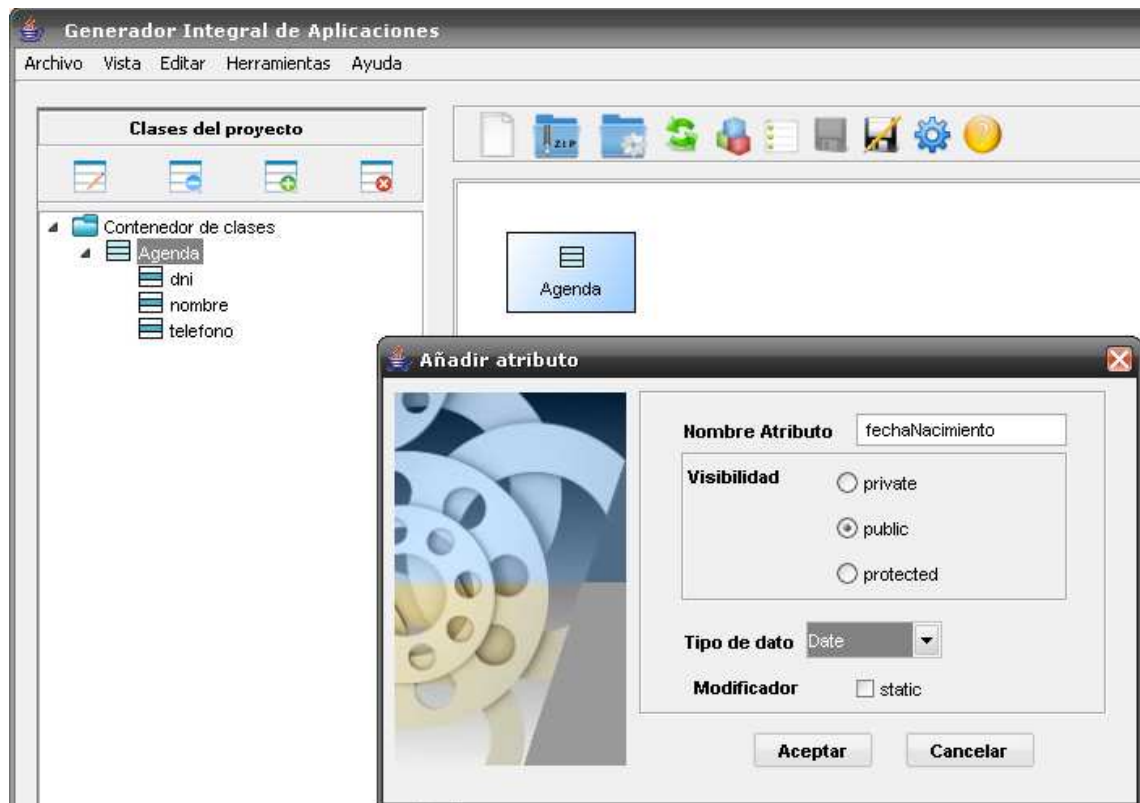
Se abrirá una nueva ventana en la que tendremos que seleccionar que tipo de elemento queremos añadir al árbol, en este caso añadiremos un atributo. Seleccionamos atributo y pulsamos el botón aceptar.



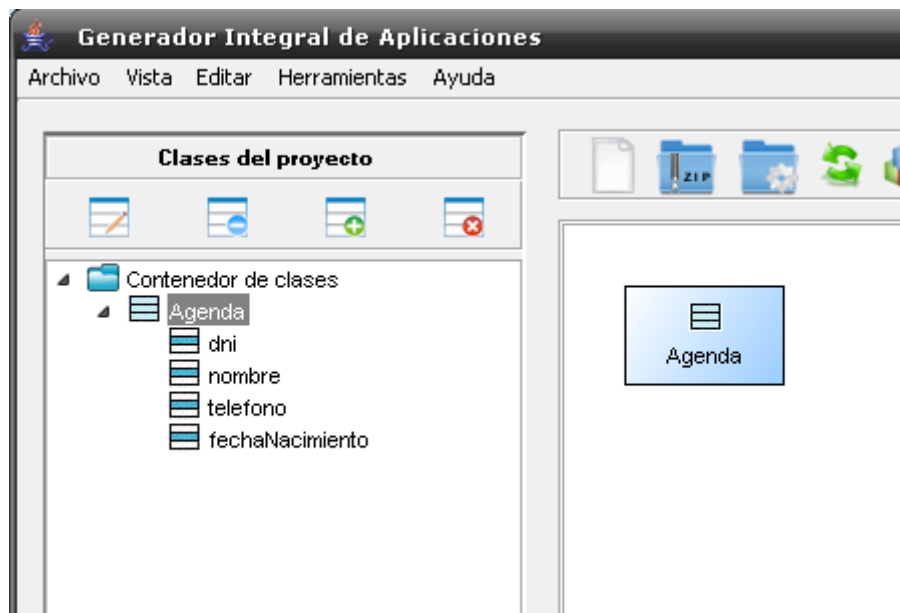
A continuación se abre una nueva ventana con todos los datos relativos a un atributo. A igual que en una clase el nombre de un atributo no puede empezar por un número y no puede tener espacios en blanco. Y para una misma clase no pueden existir dos atributos con el mismo nombre.



En este caso lo veremos con un ejemplo para comprobar cómo se añade el atributo creado al árbol de clases.

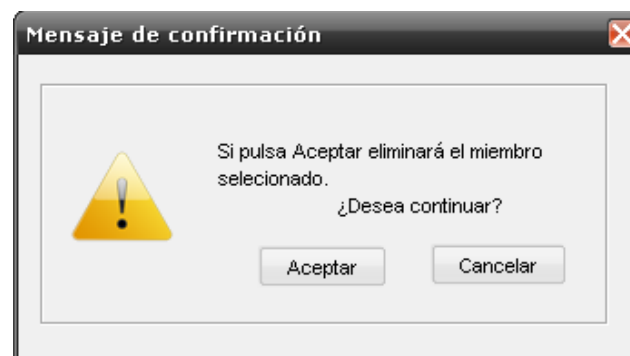


Añadimos un nuevo atributo a la clase Agenda de tipo Date. Al pulsar el botón aceptar comprobaremos que el elemento se ha añadido correctamente al árbol de clases:

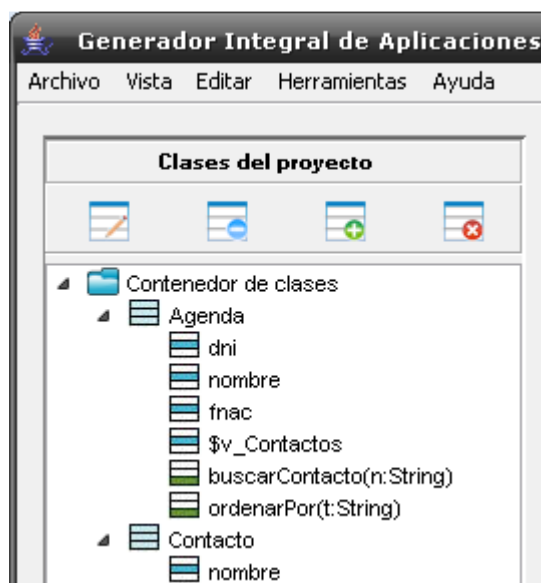


Borrar Atributo

Para borrar un atributo solo tenemos que seleccionarlo del árbol de clases y pulsar el botón 'Borrar del árbol el elemento seleccionado'.



Al aceptar el mensaje de confirmación el árbol se refrescará y el atributo teléfono habrá sido eliminado.



Modificar Atributo

Para modificar un atributo primero debemos de seleccionarlo del árbol de clases y a continuación pulsar el botón modificar

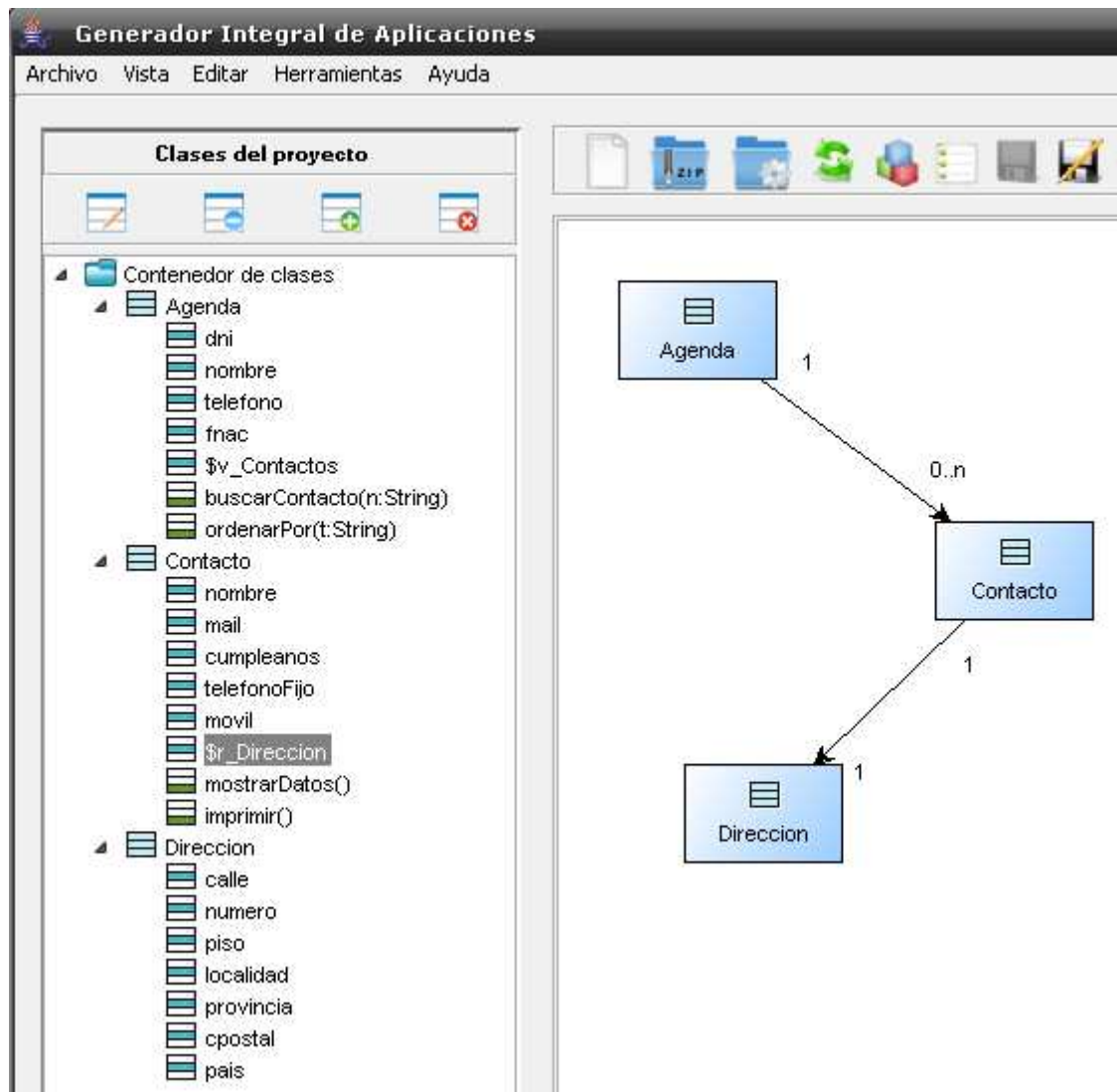


Se abrirá una nueva ventana desde la que se podrán modificar los datos del atributo.



Pongamos por ejemplo que en vez de un elemento normal queremos modificar una referencia a una clase dentro de otra, para ello volvemos al ejemplo de la Agenda de contactos.

En primer lugar tenemos en la clase Contacto una referencia a la clase Dirección. Seleccionamos ese atributo y pulsamos el botón de modificar.

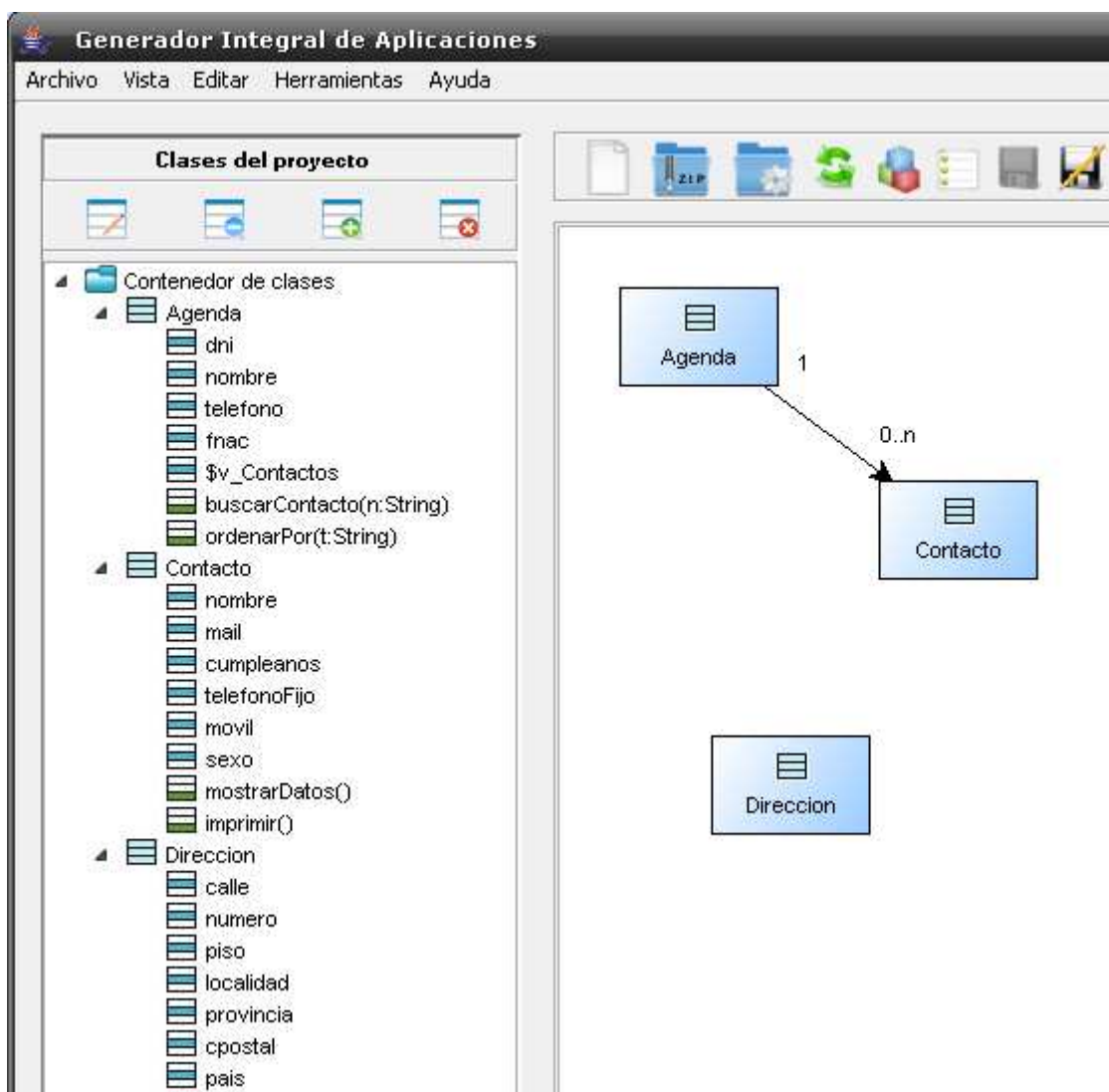


Se abre la ventana siguiente y cambiamos por completo ese atributo, cambiando en primer lugar el tipo que es 'Contacto' por 'char' y el nombre '\$r_Dirección' que se habilitará en el momento en el que se cambie el tipo por 'sexo' como en la imagen:





Una vez hecho esto pulsamos el botón aceptar y refrescamos el contenido de los paneles. El resultado es el siguiente.

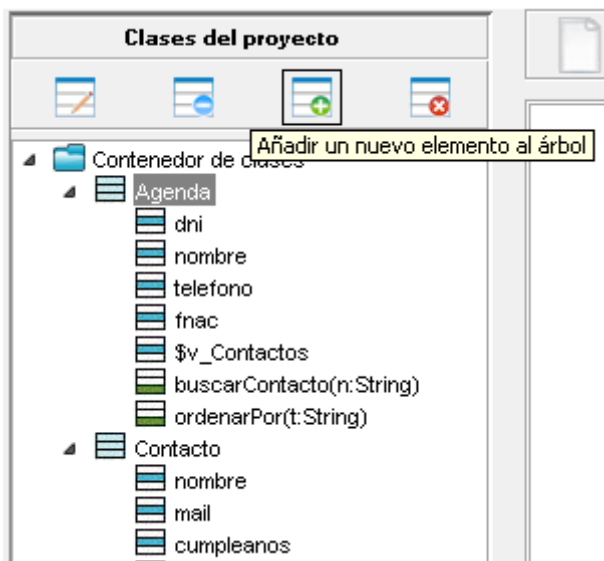


Como se puede comprobar el atributo ha cambiado y también ha desaparecido la relación que originaba el atributo '\$r_Contacto', pues era una referencia a la clase contacto.

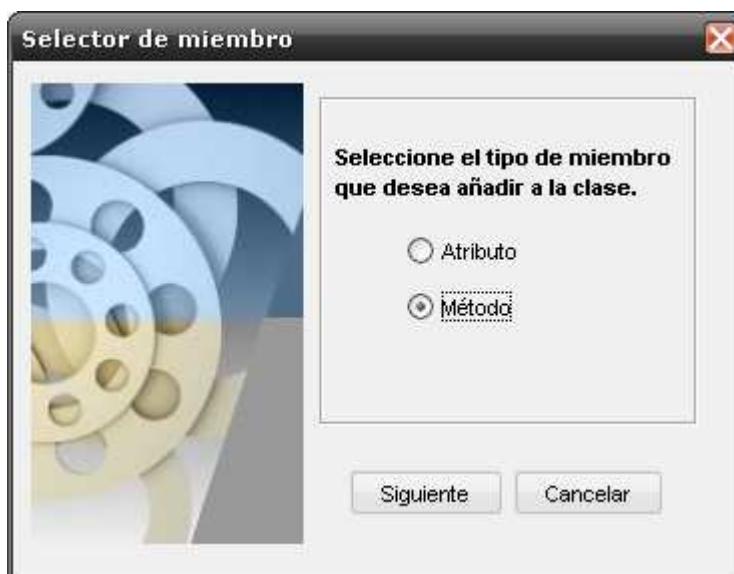
5.1.15 Gestionar Métodos

Añadir Métodos

Para añadir un atributo debemos seleccionar el nombre de la clase a la cual queremos añadir el atributo. Y a continuación pulsar el botón 'Añadir un nuevo elemento al árbol'.



Se abrirá una nueva ventana en la que tendremos que seleccionar que tipo de elemento queremos añadir al árbol, en este caso añadiremos un método. Seleccionamos método y pulsamos el botón aceptar.



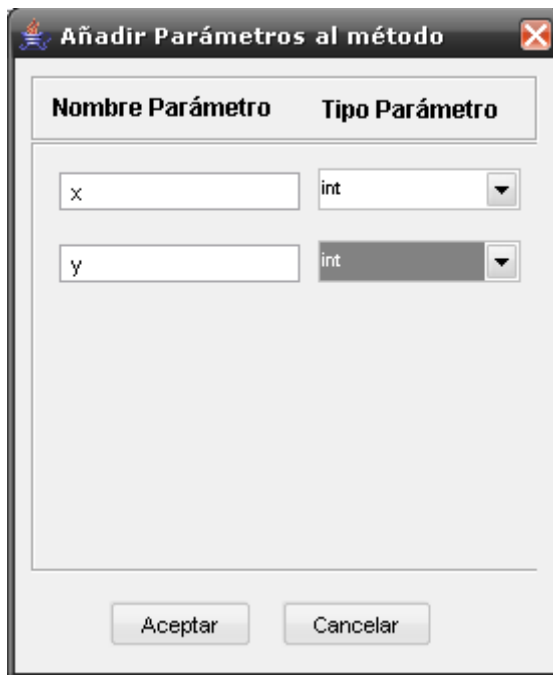
A continuación se abre una nueva ventana con todos los datos relativos a un método. A igual que en una clase el nombre de un método no puede empezar por un número y no puede tener espacios en blanco.



Cuando añadimos un método si queremos podemos añadir los parámetros del método. Para ello rellenaremos el campo número de parámetros con un valor entero.



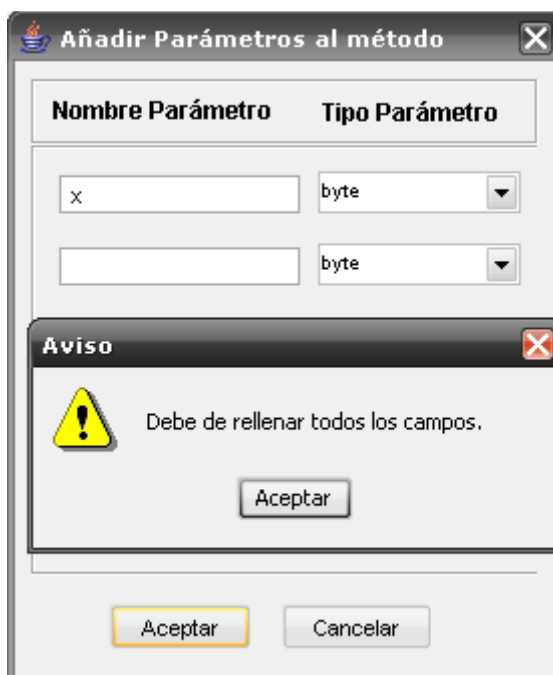
Y pulsaremos el botón siguiente, el sistema comprueba que los datos introducidos son correctos y abre una nueva ventana en la que se añadirán los parámetros del método. Como podemos ver, en este caso se abre una nueva ventana en la que se pueden rellenar los datos de dos parámetros que es el número que hemos añadido en la ventana añadir método.



| Nombre Parámetro | Tipo Parámetro |
|------------------|----------------|
| x | int |
| y | int |

Aceptar Cancelar

Pulsamos el botón aceptar y el sistema comprueba que se han rellenado todos los parámetros, en caso contrario muestra el siguiente error:



| Nombre Parámetro | Tipo Parámetro |
|------------------|----------------|
| x | byte |
| | byte |

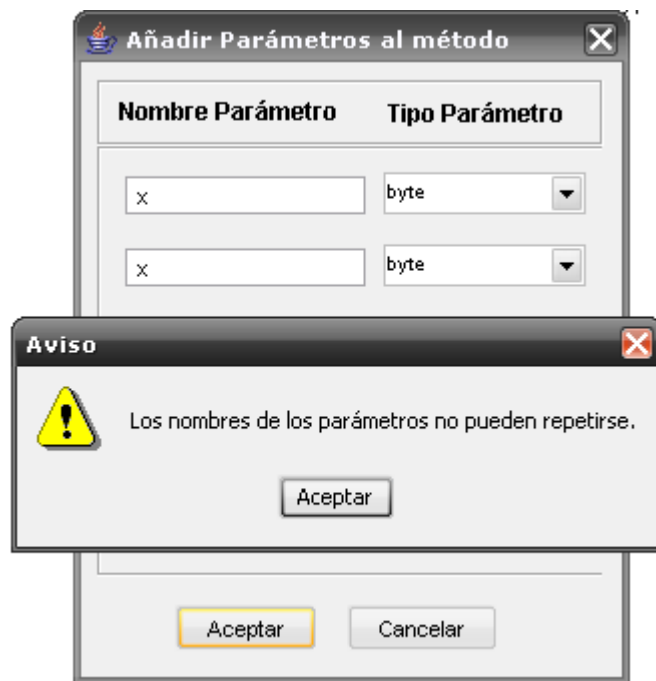
Aviso

⚠ Debe de rellenar todos los campos.

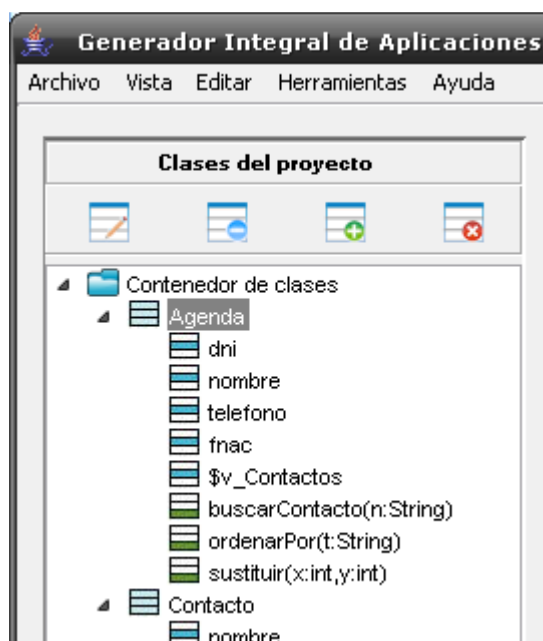
Aceptar

Aceptar Cancelar

En caso de que se repita el nombre de dos parámetros para el mismo método el sistema mostrará el siguiente error.

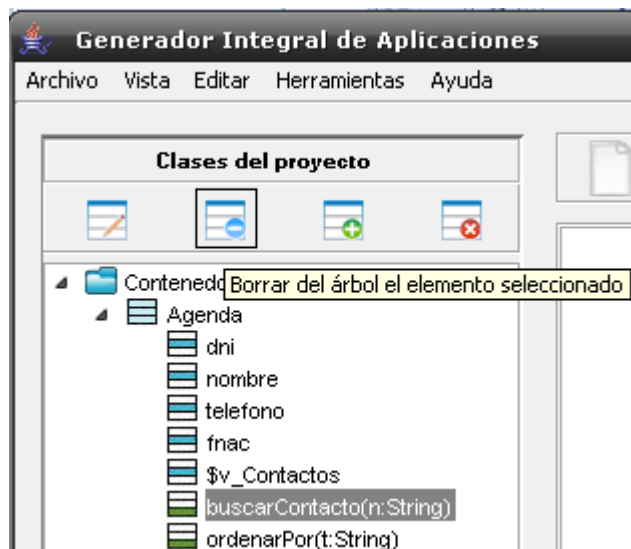


Si no se ha producido ningún error se añadirá el método al árbol. En este caso que se ha añadido el método sustituir a la clase Agenda.

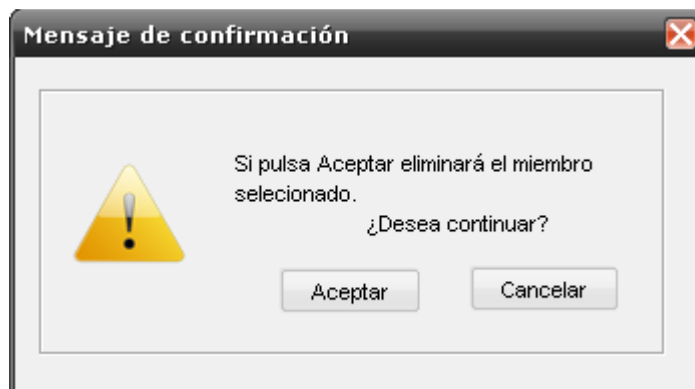


Borrar Método

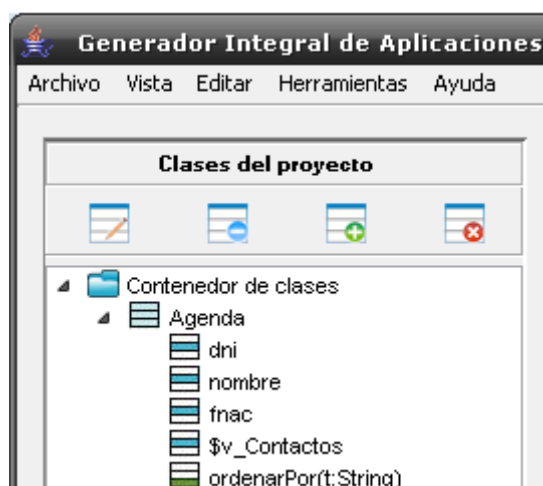
Para borrar un método solo tenemos que seleccionarlo del árbol de clases y pulsar el botón 'Borrar del árbol el elemento seleccionado'.



Confirmamos el siguiente mensaje haciendo clic en el botón aceptar.

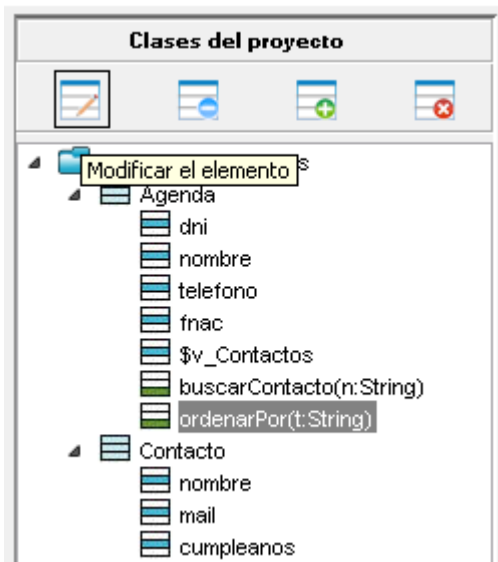


Comprobamos que se ha eliminado el método del árbol de clases.



Modificar Método


Para modificar un método hay que seleccionarlo del árbol de clases y pulsar el botón 'Modificar elemento'



Cuando pulsamos el botón el sistema abre una nueva ventana con los datos del método. Desde esta ventana podremos modificar los atributos del método y también de sus parámetros.



Si queremos modificar los parámetros debemos de hacer clic en el botón 'Modificar parámetros' situado a la izquierda en la parte inferior. El sistema abre una nueva ventana desde la que se pueden modificar o eliminar los parámetros. En caso de que queramos añadir un nuevo parámetro al método pulsaremos el botón 'Añadir parámetro'.

Si queremos eliminar un parámetro debemos de hacer clic en el botón  situado al lado de cada parámetro.

| Nombre Parámetro | Tipo Parámetro |
|------------------|----------------|
| t | String |

Modificar Cancelar Añadir parámetro

En este ejemplo añadimos un nuevo parámetro y aceptamos.

| Nombre Parámetro | Tipo Parámetro |
|------------------|----------------|
| b | byte |

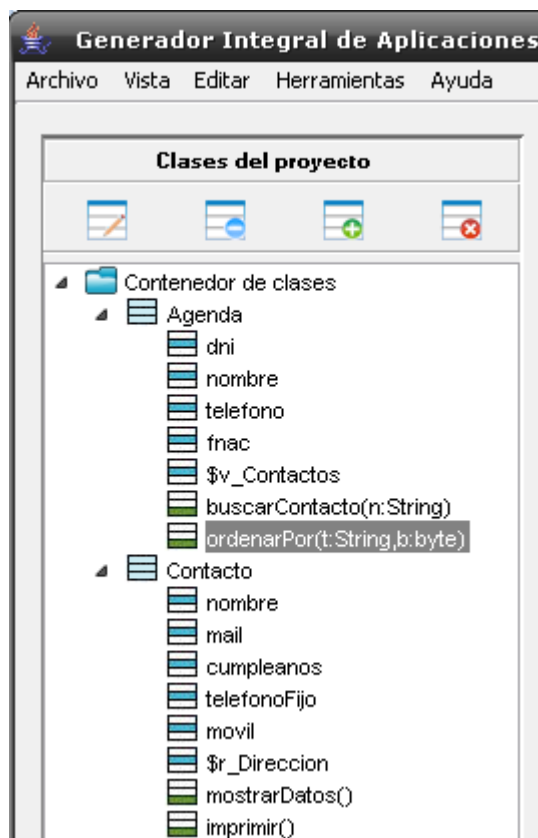
Aceptar Cancelar

Vemos que el parámetro se ha añadido a la lista de parámetros del método:

| Nombre Parámetro | Tipo Parámetro |
|------------------|----------------|
| t | String |
| b | byte |

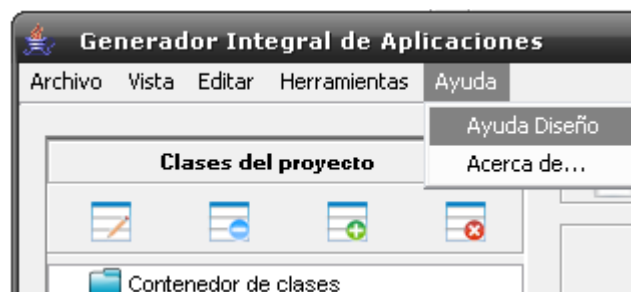
Modificar Cancelar Añadir parámetro

A continuación pulsamos el botón 'Modificar', se cierra la ventana y se puede ver como los parámetros se han modificado correctamente en el árbol de clases.

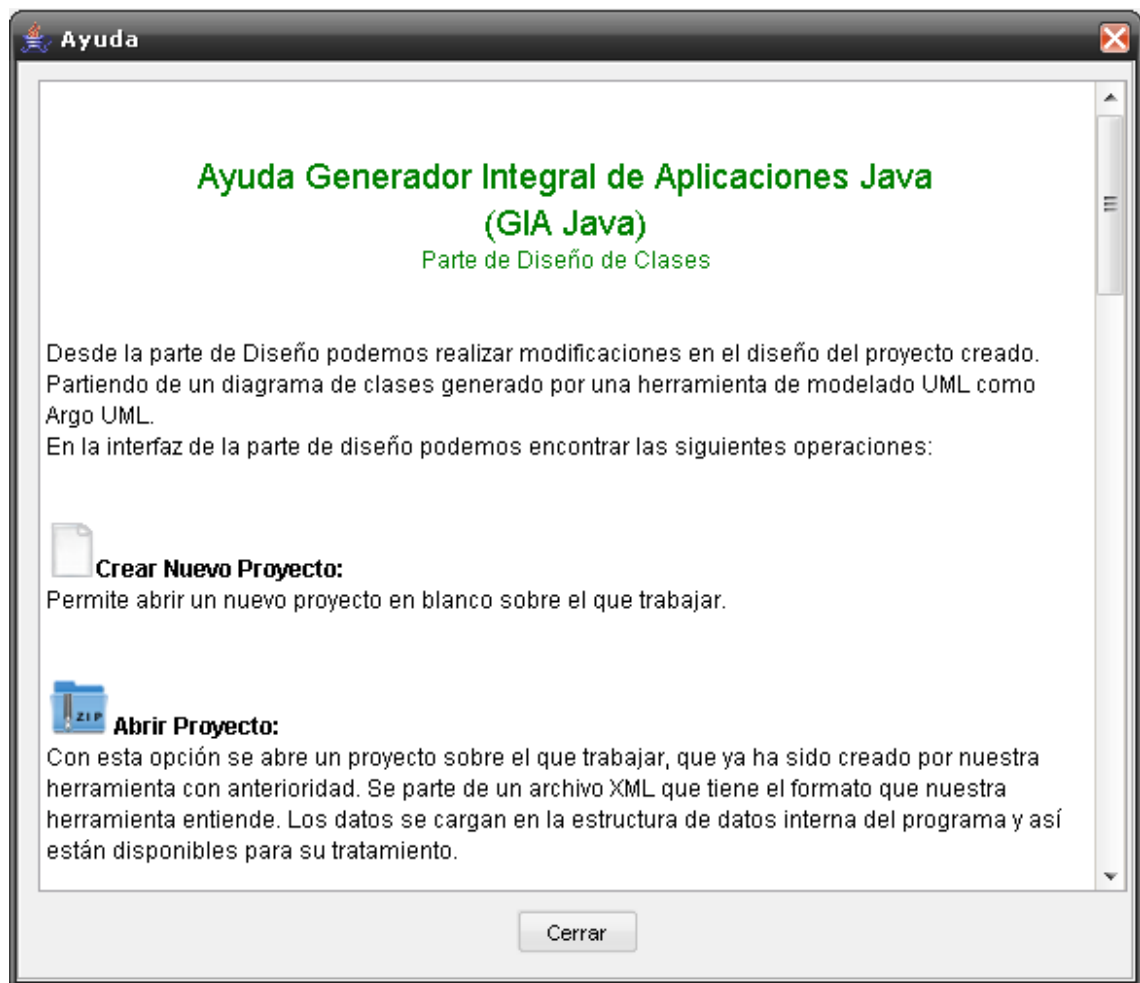


5.1.16 Ayuda de Diseño

Si se desea entrar en la ayuda de la aplicación estando en la vista de Diseño, deberemos acceder al menú Ayuda- Ayuda Diseño, o pulsar el botón 'Ayuda' situado entre los botones de la barra de herramientas.



El sistema abrirá una ventana desde la que se verá una breve explicación de cada una de las opciones del menú y botones de la interfaz de la aplicación.



5.2 PARTE DE IDE DE JAVA

Una vez que se ha generado el código de la interfaz se abre una nueva vista dentro de la aplicación GiAJava, denominada 'IDE de Java'. Recibe este nombre porque es un entorno de desarrollo desde el que se podrá acceder al código generado para cada clase.

Para cada clase se han generado clases completas: constructores, métodos para acceder y modificar las propiedades de un objeto, métodos para acceder a las relaciones entre objetos, métodos para guardar, buscar y recuperara los objetos en formato XML.

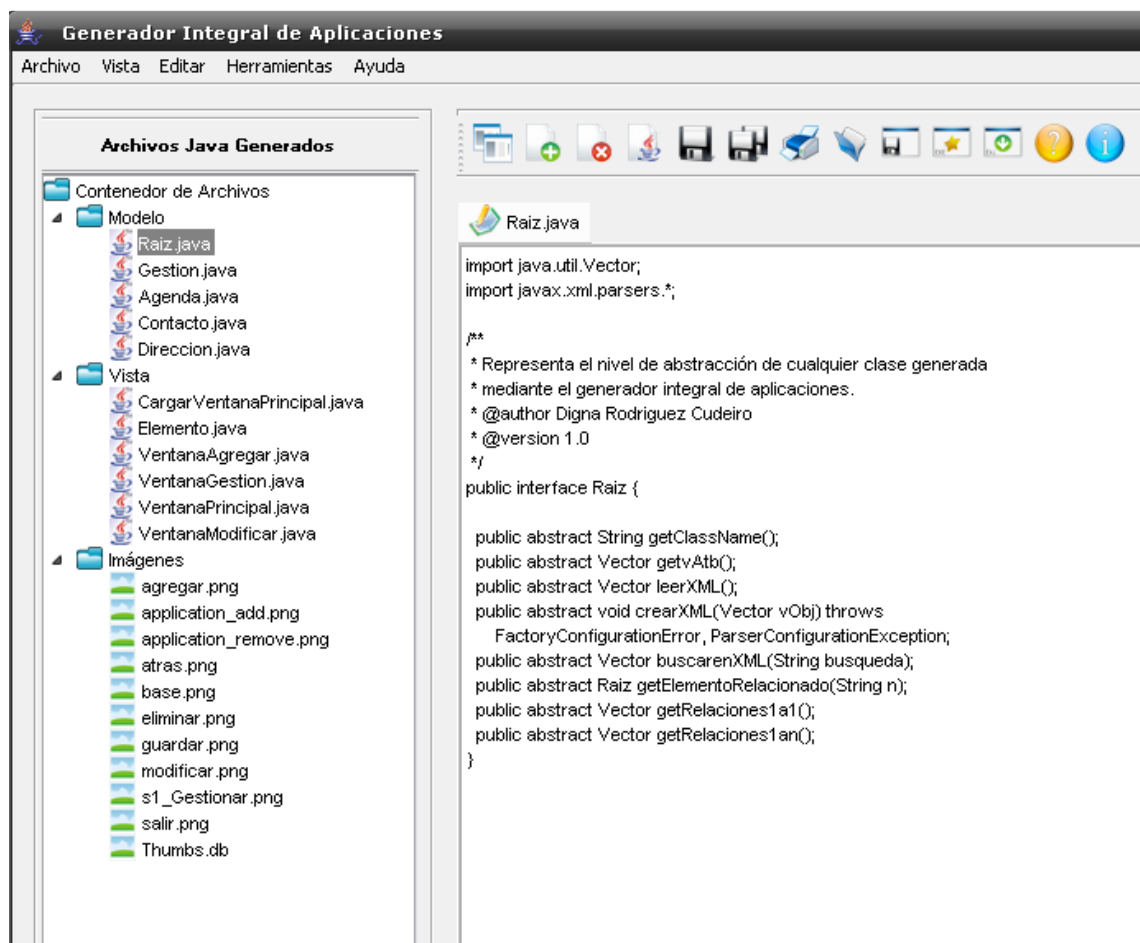
Además del código de las clases se genera la GUI de la aplicación, desde la que el usuario interactuará con la aplicación generada y podrá añadir, modificar o borrar datos de los objetos. Esta interfaz será la base de la aplicación software que deseemos implementar.

Los objetos se almacenarán en archivos externos y en formato XML. Cada clase tiene su correspondiente archivo XML, con todos los datos de los objetos de la clase.

Desde la vista 'IDE de java' se podrán editar los archivos java generados además de compilar y ejecutar el proyecto generado.

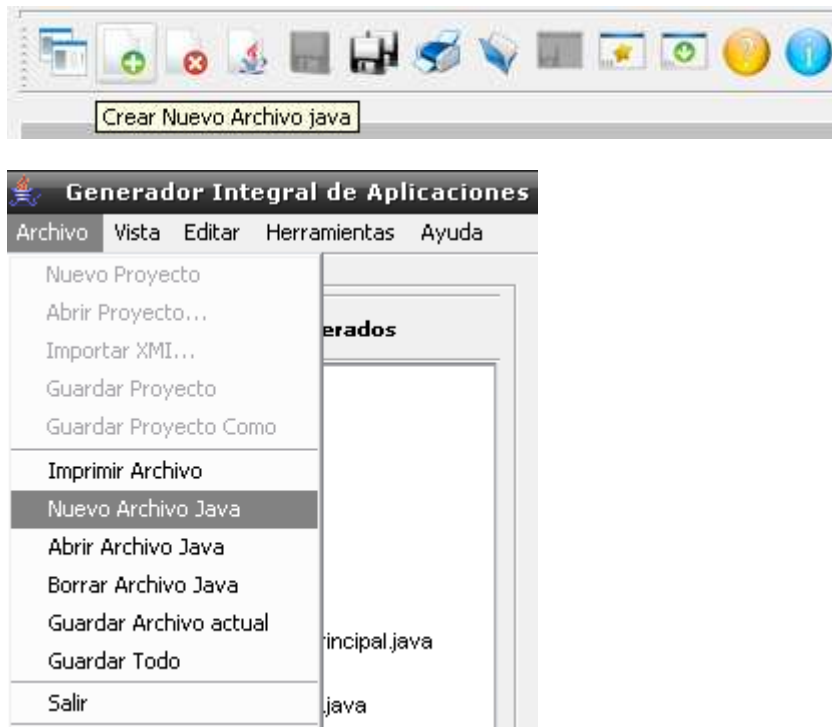
Las ventanas que forman parte de la interfaz se encuentran debajo de una carpeta denominada Vista dentro del árbol del archivos java generados.

Los archivos java de las clases se encuentran debajo de la carpeta Modelo dentro del árbol de archivos java generados.



5.2.1 Nuevo Archivo Java

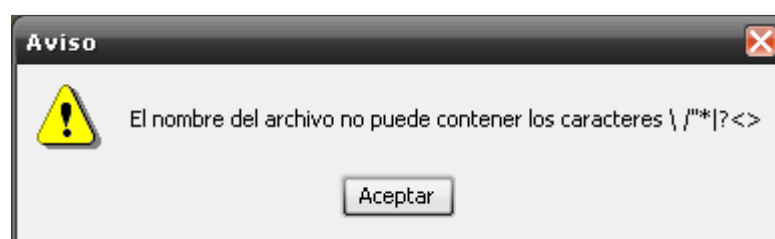
Si queremos añadir un nuevo archivo java al proyecto debemos acceder al menú Archivo – Nuevo Archivo Java, o pulsar el botón 'Crear Nuevo Archivo java' situada en la barra de herramientas de la parte de IDE de java.



El sistema abre una nueva ventana en la que se puede dar un nombre al nuevo archivo y se escoge el directorio debajo del que se quiere añadir el nuevo archivo, dependiendo de si forma parte del modelo o de la interfaz.



Si se introducen caracteres erróneos en el nombre del fichero el sistema mostrará el siguiente mensaje de error:



En caso de que no se pulse el botón aceptar sin haber rellenado el nombre del archivo se muestra el siguiente aviso:



Si se da un nombre al archivo que ya existe en el sistema se muestra el siguiente mensaje de error:



En caso de que no ocurra ninguno de los errores anteriormente mencionados se añadirá el nuevo archivo al árbol y se creará un archivo vacío en el directorio en el que se han generado los archivos java.

5.2.2 Abrir Archivo Java

Si queremos añadir al proyecto un archivo de código java situado en nuestro disco duro deberemos acceder al menú Archivo – Abrir Archivo Java o pulsar el botón 'Añadir archivo java ya existente'.

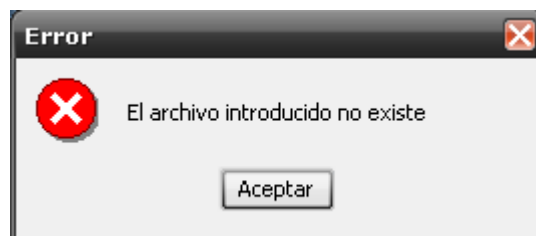




El sistema abre una nueva ventana desde la que se debe de buscar un archivo java existente en el disco duro. El nombre del archivo se puede introducir manualmente o mediante el botón '...' que filtrará los archivos por los que tienen extensión .java.

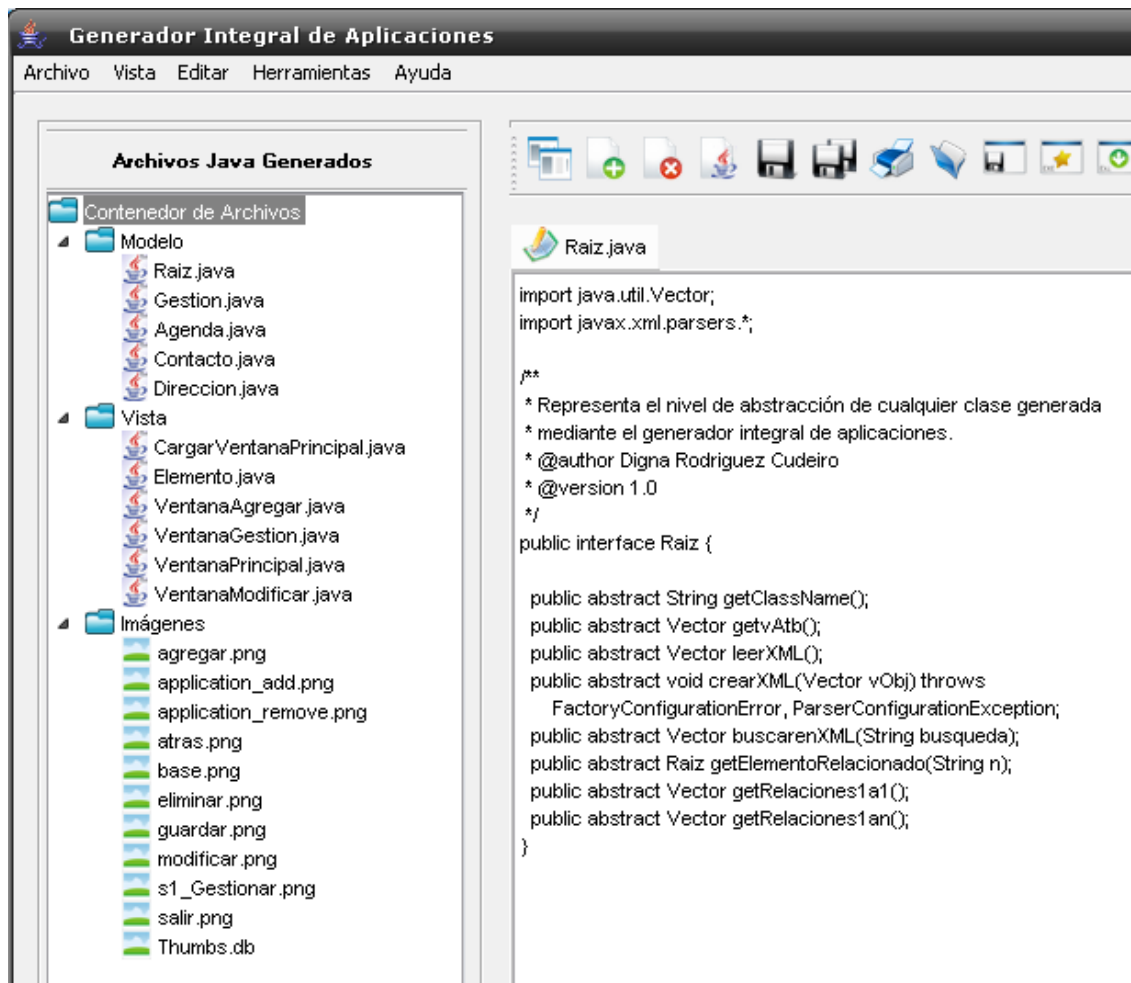


Si se introduce una ruta errónea el sistema muestra el siguiente mensaje de error:

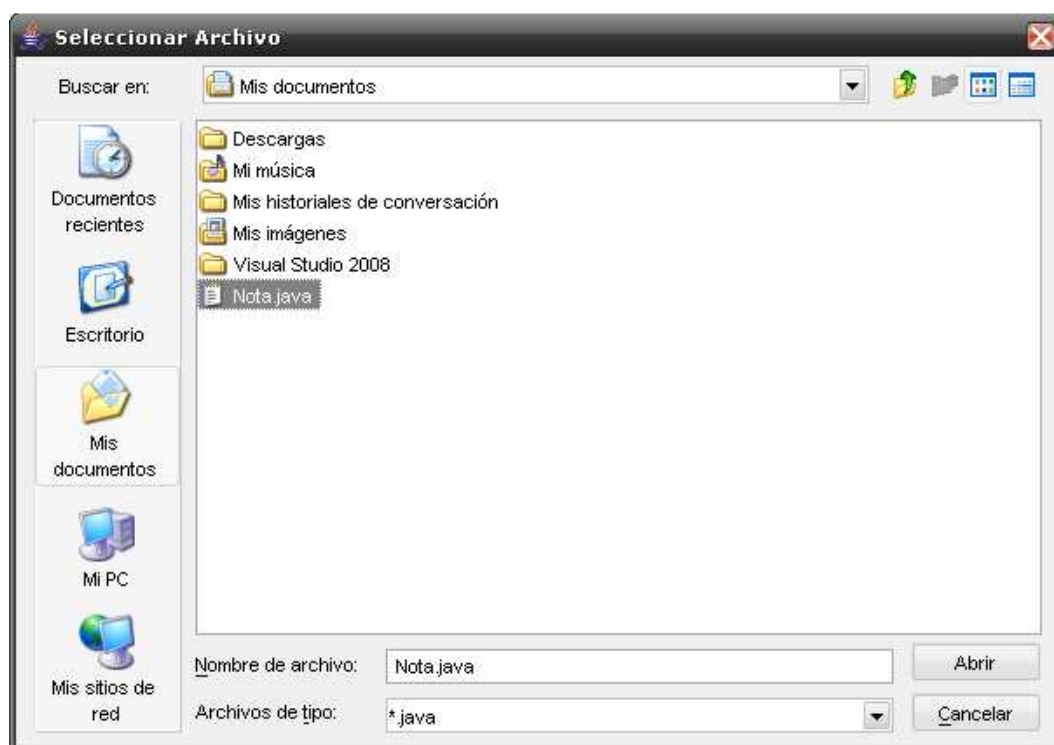


En caso de que no se haya producido ningún error se seleccionará el archivo java y se añadirá al árbol.

Podemos verlo en el siguiente ejemplo:

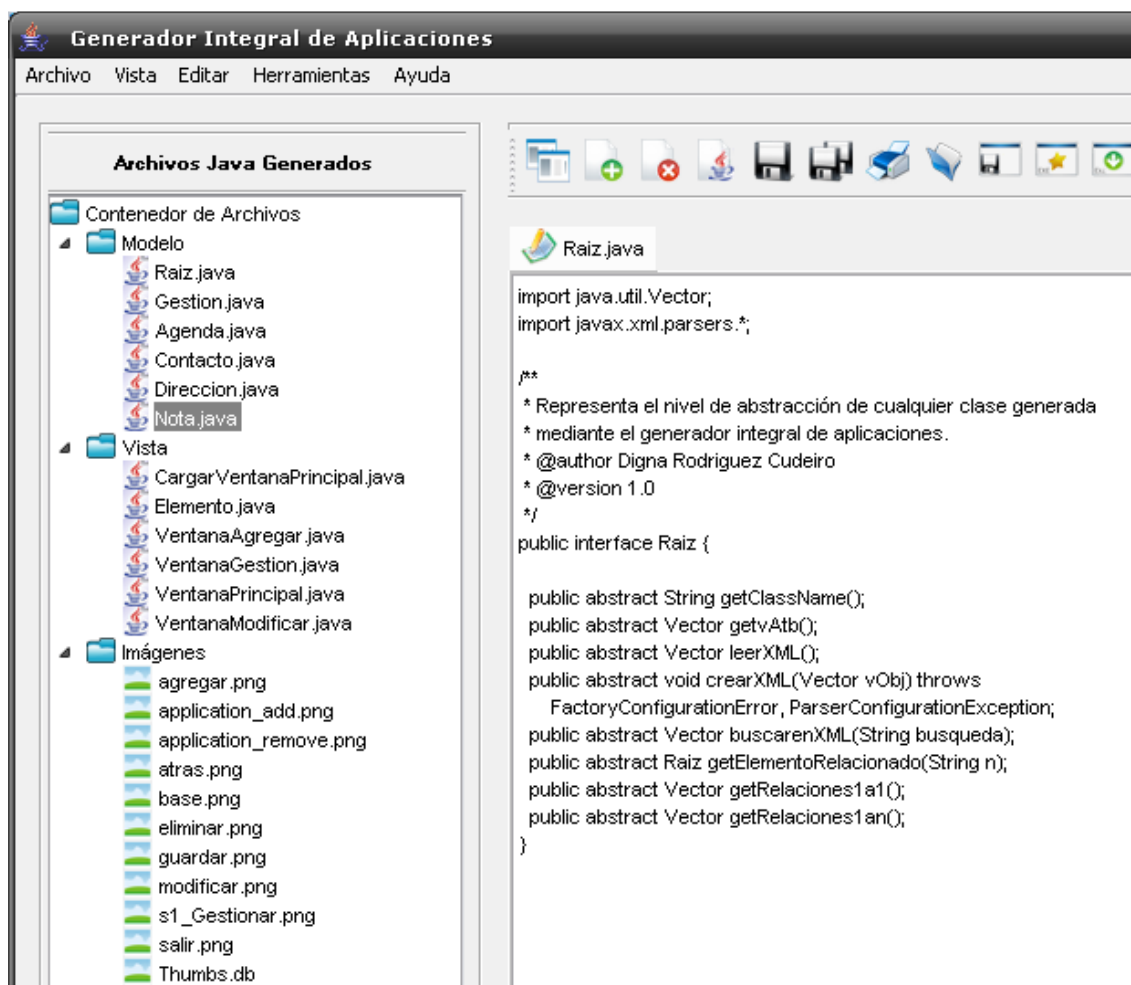


Inicialmente en la herramienta se encuentran los archivos que cuelgan del árbol de archivos java generados.





Pulsamos el botón aceptar y el sistema valida que el fichero exista y si existe se copiará en el directorio en el que se encuentran el resto de archivos java y se añadirá al árbol de archivos java.

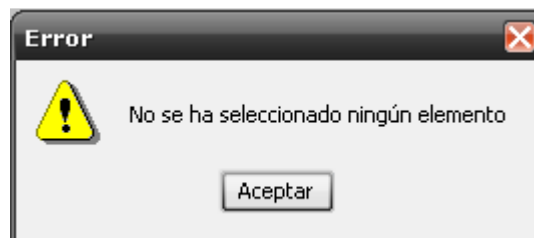


5.2.3 Borrar Archivo Java

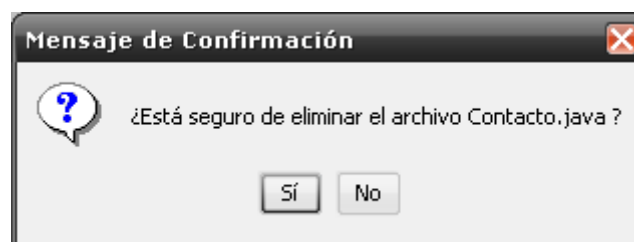
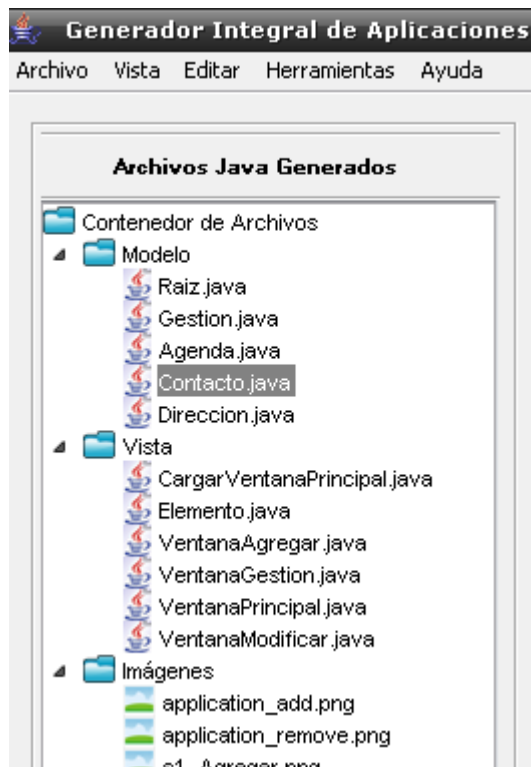
Si se desea borrar alguno de los ficheros java se debe de seleccionar el fichero del árbol y se accederá al menú Archivo - Borrar archivo java, o desde el botón 'Borrar fichero seleccionado' de la barra de herramientas.



En caso de que no se seleccione ningún nodo del árbol y se pulse el botón eliminar el sistema mostrará el siguiente mensaje de error:



Veremos con un ejemplo todo el proceso de borrado. En primer lugar seleccionamos la clase Contacto del directorio Modelo, luego se pulsa el botón borrar y a continuación el sistema muestra un mensaje de confirmación. El usuario acepta el mensaje de confirmación y el sistema se encarga de eliminar el archivo de disco y del árbol de java generado.

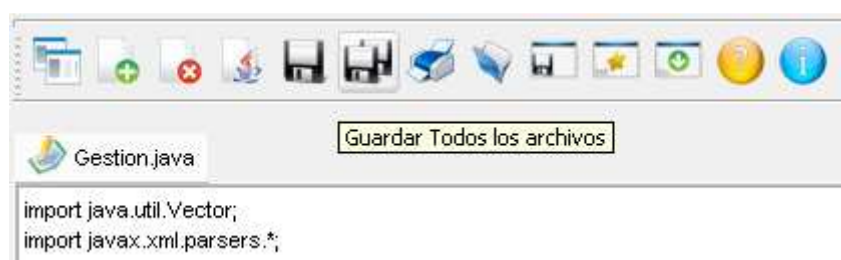


Una vez confirmado el borrado del archivo el sistema mostrará en la barra inferior de información del sistema el siguiente mensaje:

Se ha borrado el archivo Contacto.java

5.2.4 Guardar Todo

Si queremos guardar a la vez todos los archivos que estén abiertos en pestañas debemos de acceder al menú Archivo – Guardar todo, o desde el botón ‘Guardar Todos los archivos’





Si no hay ningún archivo abierto en una pestaña, en la barra inferior de información del sistema se mostrará el siguiente mensaje:

No hay ningún archivo que guardar.

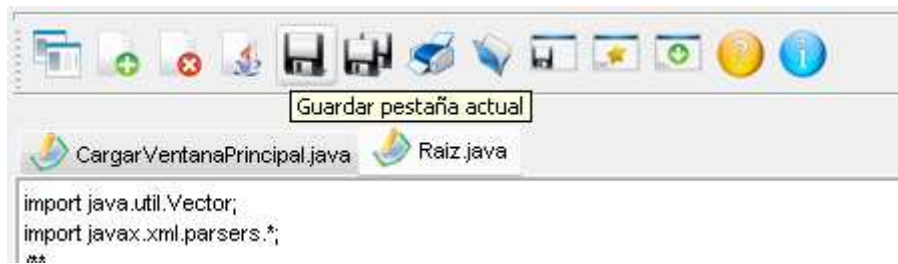
Si hay archivos abiertos en pestañas, en la barra inferior de información del sistema se mostrará el siguiente mensaje:

Se han guardado con éxito todos los archivos

5.2.5 Guardar Pestaña

Si queremos guardar sólo el archivo abierto en la pestaña seleccionada debemos acceder al menú Archivo – Guardar Archivo actual, o desde el botón 'Guardar pestaña actual' de la barra de herramientas.





Si el archivo se guarda con éxito en la barra de información del sistema se muestra el siguiente mensaje:

Se ha guardado con éxito el archivo Raiz.java

El botón guardar pestaña actual sólo está activo si existe algún archivo abierto en una pestaña.

5.2.6 Abrir contenedor de archivos

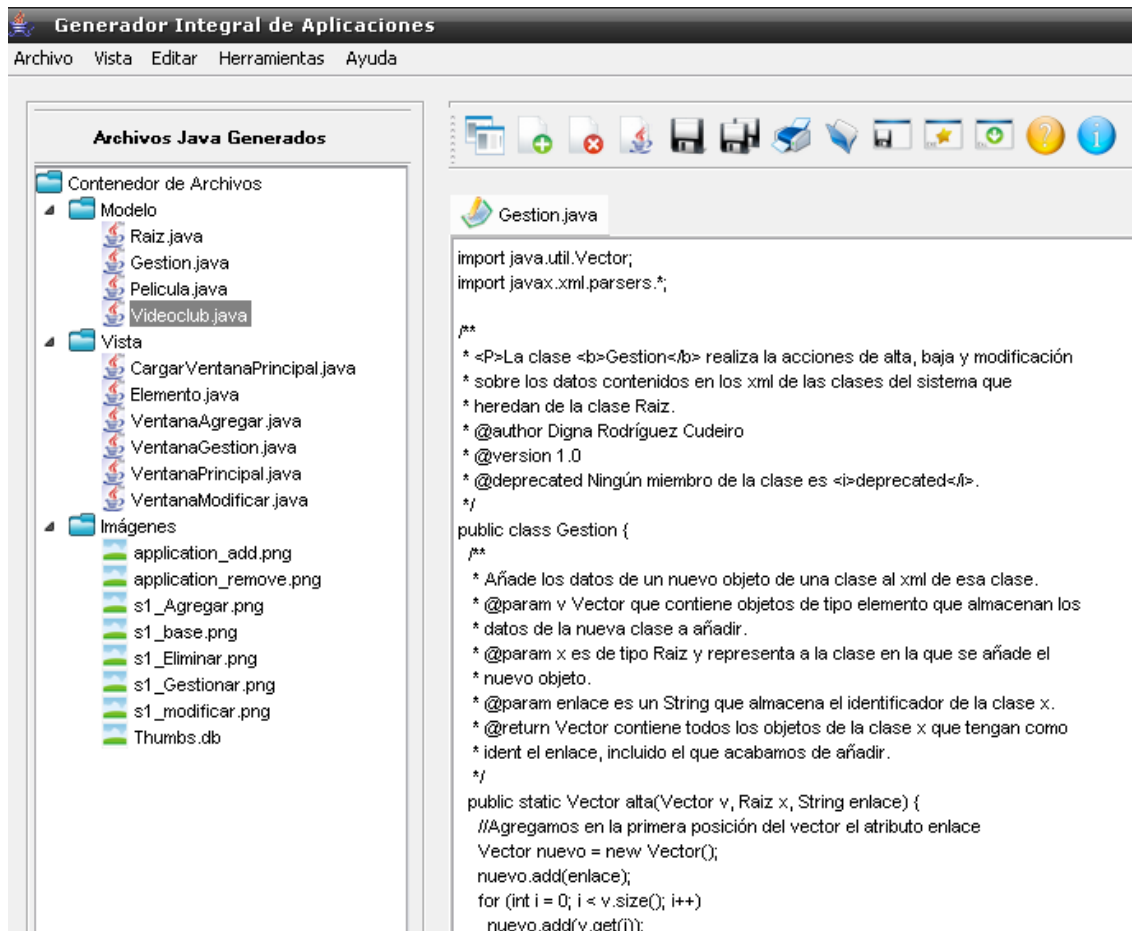
Si queremos acceder directamente al directorio en el que se encuentran todos los ficheros java generados y creados desde el IDE de java podremos acceder al menú Herramientas – Abrir Contenedor de Archivos o pulsando el botón ‘Abrir Contenedor de Archivos’ de la barra de herramientas.



Esta opción abre en el explorador de archivos el directorio que contiene todos los archivos generados.

5.2.7 Abrir Archivo en pestaña

Si queremos abrir un archivo en una pestaña debemos de hacer doble clic en el nombre del archivo.

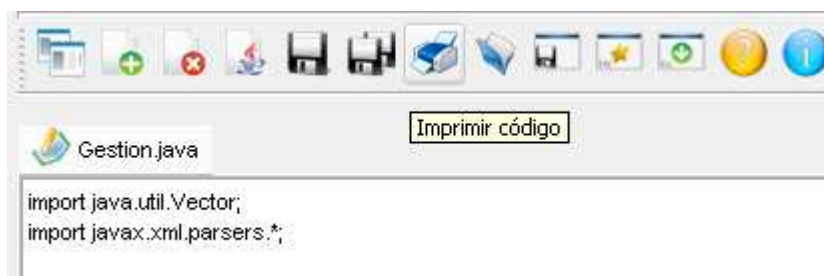


En estos momentos se activa el botón de 'cerrar' situado en la parte derecha de la aplicación.



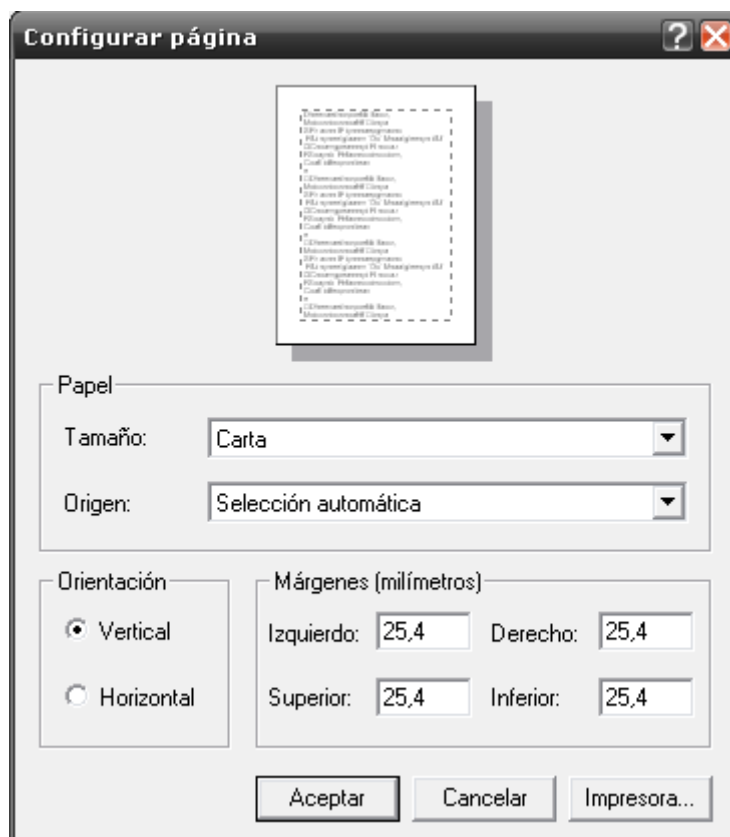
5.2.8 Imprimir Archivo

Si queremos imprimir el código de un archivo abierto debemos de acceder al menú Archivo – Imprimir archivo o al botón 'Imprimir código' de la barra de herramientas.

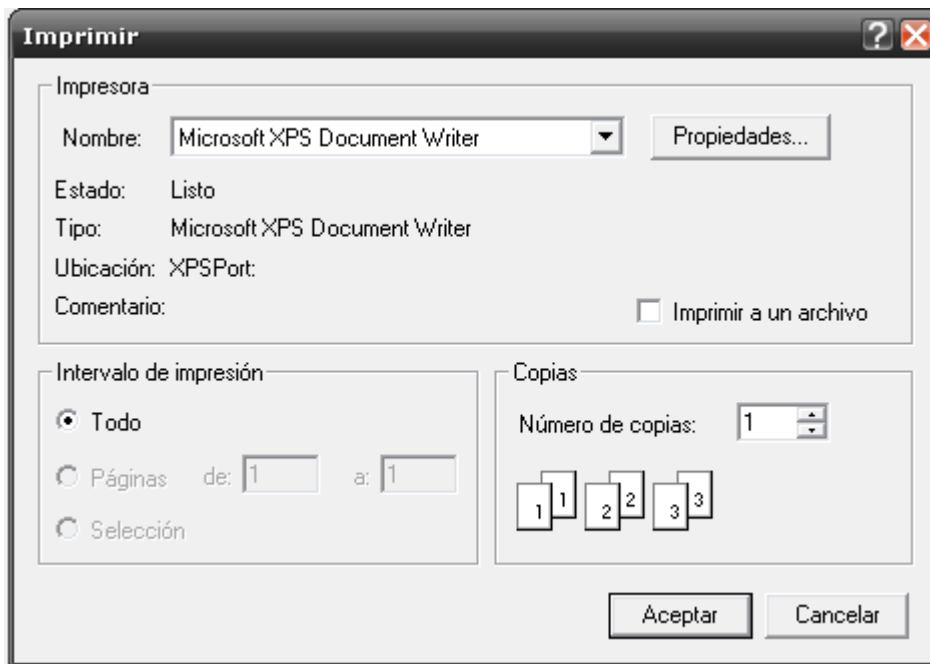




A continuación se abre un asistente desde el que podremos editar las propiedades de página.



También podremos seleccionar la impresora desde la siguiente ventana.



5.2.9 Compilar Pestaña

Si queremos compilar y guardar un solo archivo, en primer lugar lo abriremos en una nueva pestaña haciendo doble clic en su nombre en el árbol de archivos java y a continuación accederemos al menú Herramientas – Compilar pestaña actual. También podremos acceder desde el botón ‘Guardar y Compilar pestaña actual’.



En caso de que no exista ningún archivo abierto el sistema muestra en la barra de información del sistema el siguiente mensaje:

No hay ningún archivo abierto que compilar.

Si hay un archivo abierto se muestran los resultados de la compilación en el panel de mensajes situado en la parte inferior. En caso de que la compilación se haya realizado con éxito se muestra el siguiente mensaje:



The screenshot shows a code editor window titled 'Raiz.java'. The code defines an abstract interface 'Raiz' with several methods. Below the code, a message box indicates that the current tab was compiled successfully.

```

import java.util.Vector;
import javax.xml.parsers.*;

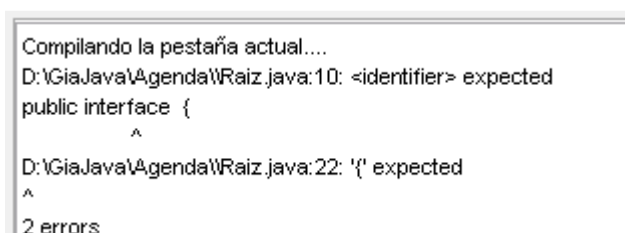
/**
 * Representa el nivel de abstracción de cualquier clase generada
 * mediante el generador integral de aplicaciones.
 * @author Digna Rodriguez Cudeiro
 * @version 1.0
 */
public interface Raiz {

    public abstract String getClassName();
    public abstract Vector getvAtb();
    public abstract Vector leerXML();
    public abstract void crearXML(Vector vObj) throws
        FactoryConfigurationError, ParserConfigurationException;
    public abstract Vector buscarenXML(String busqueda);
    public abstract Raiz getElementoRelacionado(String n);
    public abstract Vector getRelaciones1a1();
    public abstract Vector getRelaciones1an();
}

```

Compilando la pestaña actual....
 archivo Raiz.java.java compilado.

Si se han producido errores estos se mostrarán también en el panel de mensajes:



The screenshot shows the same code editor window, but the message box now displays two compilation errors. The first error is on line 10, where the compiler expected an identifier but found an opening curly brace. The second error is on line 22, where the compiler expected a closing curly brace but found an opening curly brace.

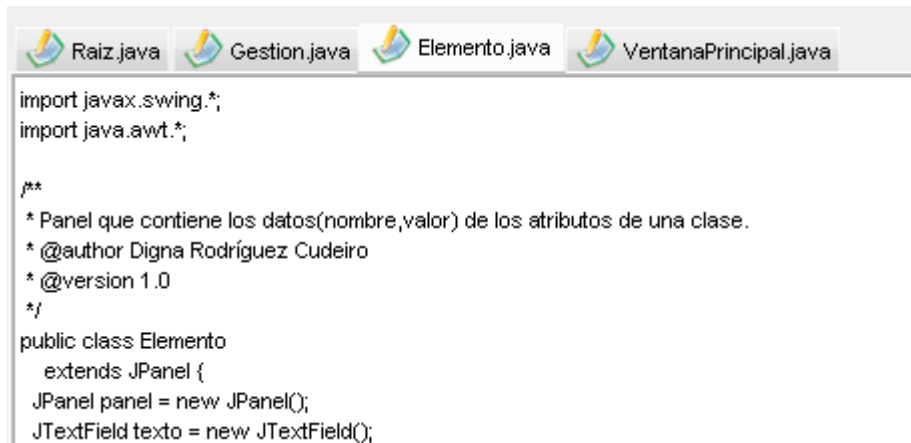
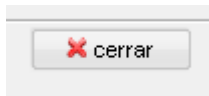
```

Compilando la pestaña actual....
D:\GiaJava\Agenda\Raiz.java:10: <identifier> expected
public interface {
      ^
D:\GiaJava\Agenda\Raiz.java:22: '{' expected
      ^
2 errors

```


5.2.10 Cerrar Pestaña

Si queremos cerrar una pestaña simplemente debemos de seleccionar la pestaña a cerrar y pulsar el botón cerrar situado en la parte derecha.

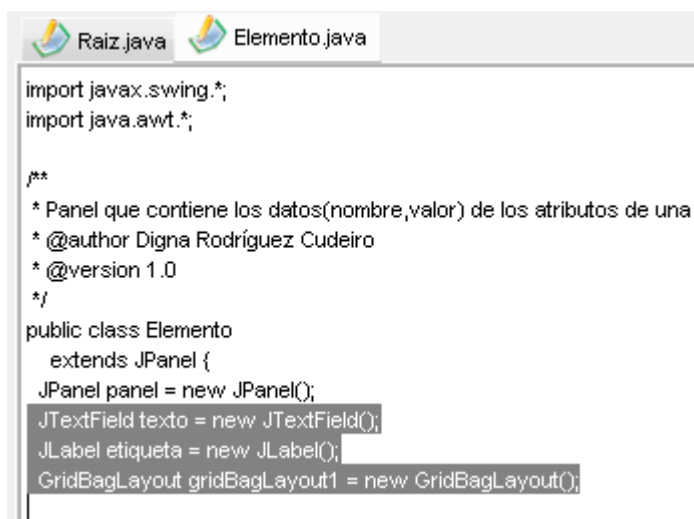


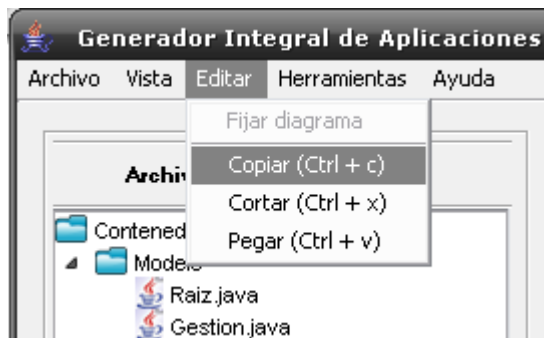
Si no hay ninguna pestaña abierta el botón aparece deshabilitado.



5.2.11 Copiar selección de código

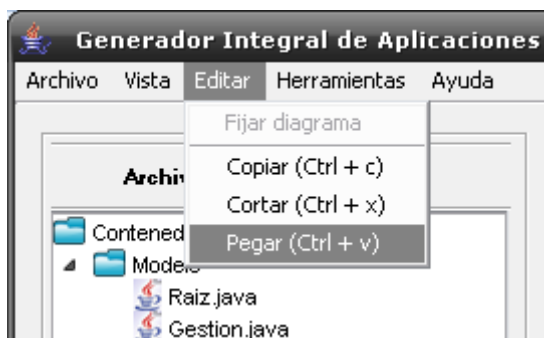
Si queremos copiar código para pegarlo en otro lugar seleccionaremos el código y accederemos al menú Editar- Copiar o pulsaremos en el teclado las teclas 'Ctrl + c'.





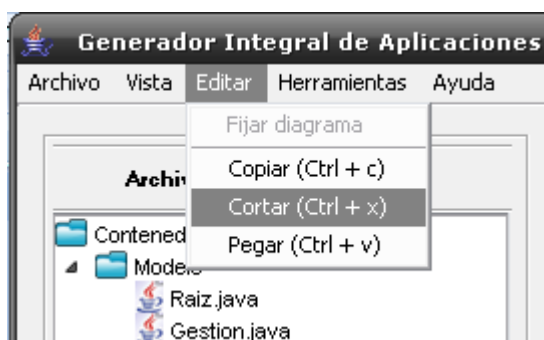
5.2.12 Pegar selección de código

Para pegar el código que hemos copiado previamente, situaremos el cursor en el panel de código y accederemos al menú Editar – Pegar o pulsaremos en el teclado las teclas 'Ctrl + v',



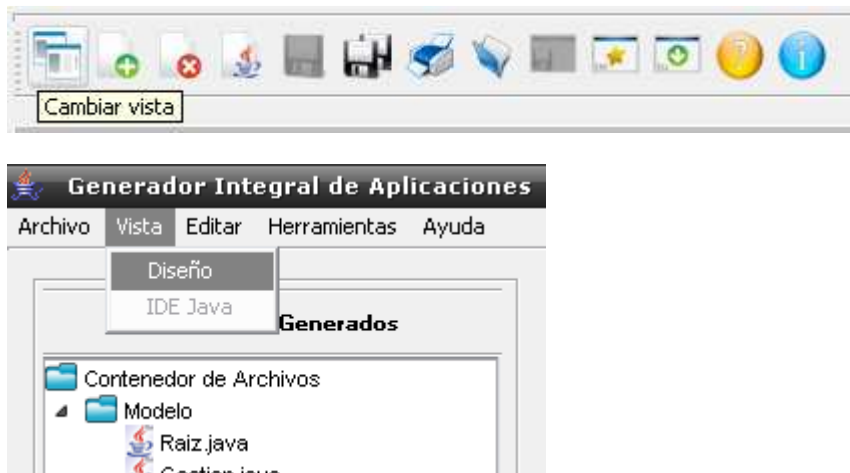
5.2.13 Cortar selección de código

Si queremos cortar código para pegarlo en otro lugar seleccionaremos el código y accederemos al menú Editar- Copiar o pulsaremos en el teclado las teclas 'Ctrl + x'.

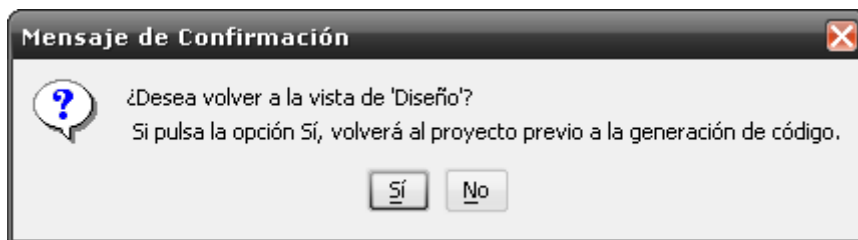


5.2.14 Cambiar a Vista Diseño

Si en algún momento decidimos que queremos modificar el modelo del cual partimos en el inicio, debemos de pulsar el botón 'Cambiar vista', o acceder al menú Vista – Diseño.

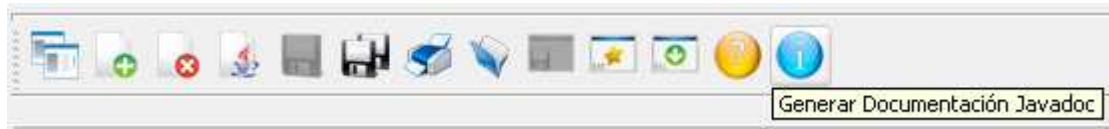


Una vez activada la opción de cambiar de vista, el sistema mostrará el siguiente mensaje de confirmación, y una vez confirmado se volverá al modelo previo a la generación de código.

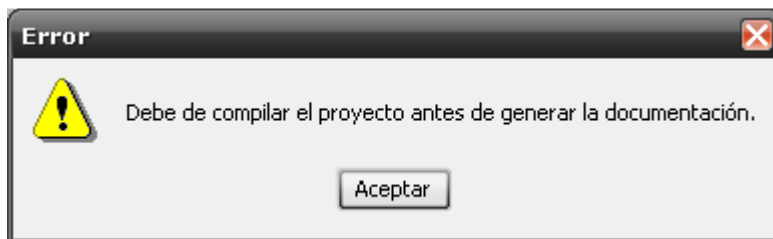


5.2.15 Generar Documentación Javadoc

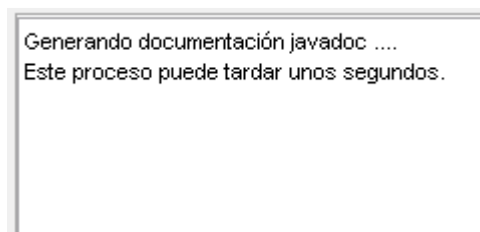
Si queremos generar la documentación javadoc relativa a las clases generadas debemos de pulsar el botón 'Generar Documentación Javadoc' de la barra de herramientas o acceder al menú Herramientas – Generar documentación Javadoc.



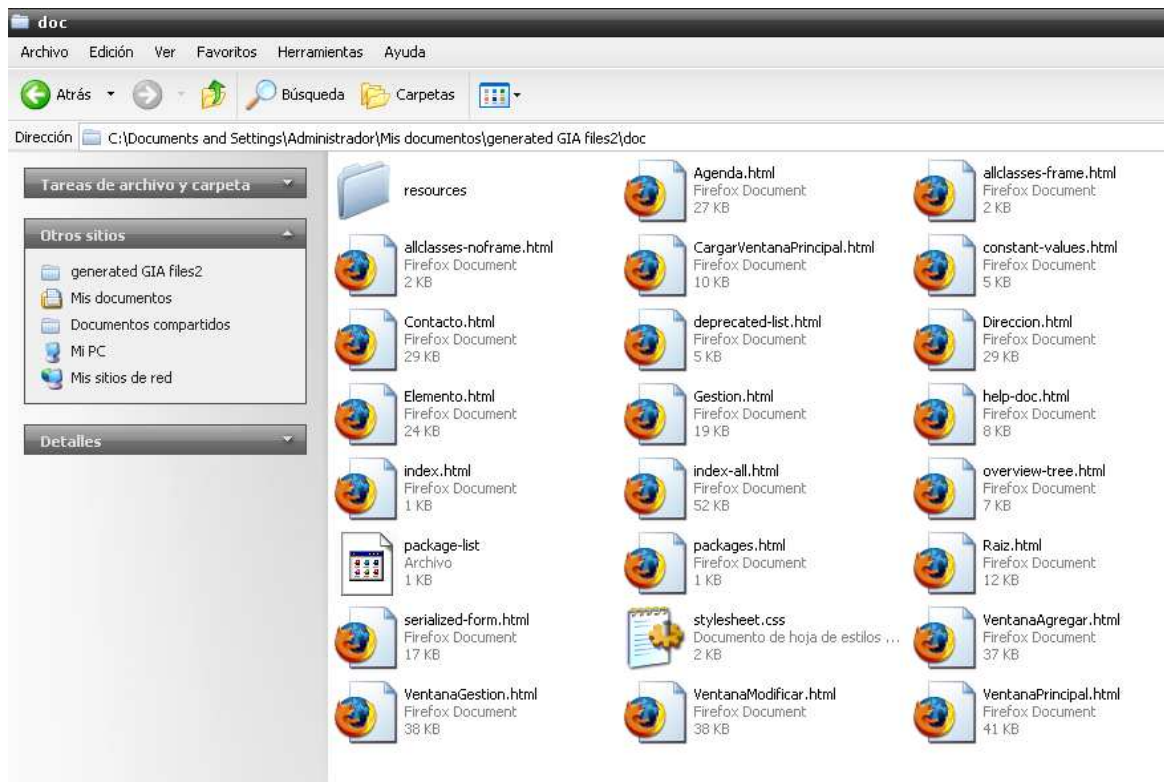
Si no se ha compilado el proyecto antes de generar la documentación la herramienta mostrará el siguiente mensaje.



La herramienta mostrará el siguiente mensaje en el panel inferior de mensajes:



Se crearán los archivos de documentación javadoc en una carpeta doc situada en la carpeta de generación de código del sistema.



Se abre el navegador y se muestra la documentación javadoc de la clase principal del sistema.

Agenda

All Classes
[Agenda](#)
[CargarVentanaPrincipal](#)
[Contacto](#)
[Direccion](#)
[Elemento](#)
[Gestion](#)
[Raiz](#)
[VentanaAgregar](#)
[VentanaGestion](#)
[VentanaModificar](#)
[VentanaPrincipal](#)

Package **Class** Tree Deprecated Index Help

PREV CLASS [NEXT CLASS](#)

SUMMARY: NESTED | [FIELD](#) | [CONSTR](#) | [METHOD](#)

FRAMES [NO FRAMES](#)

DETAIL: [FIELD](#) | [CONSTR](#) | [METHOD](#)

Class Agenda

java.lang.Object
 └ **Agenda**

All Implemented Interfaces:
[Raiz](#)

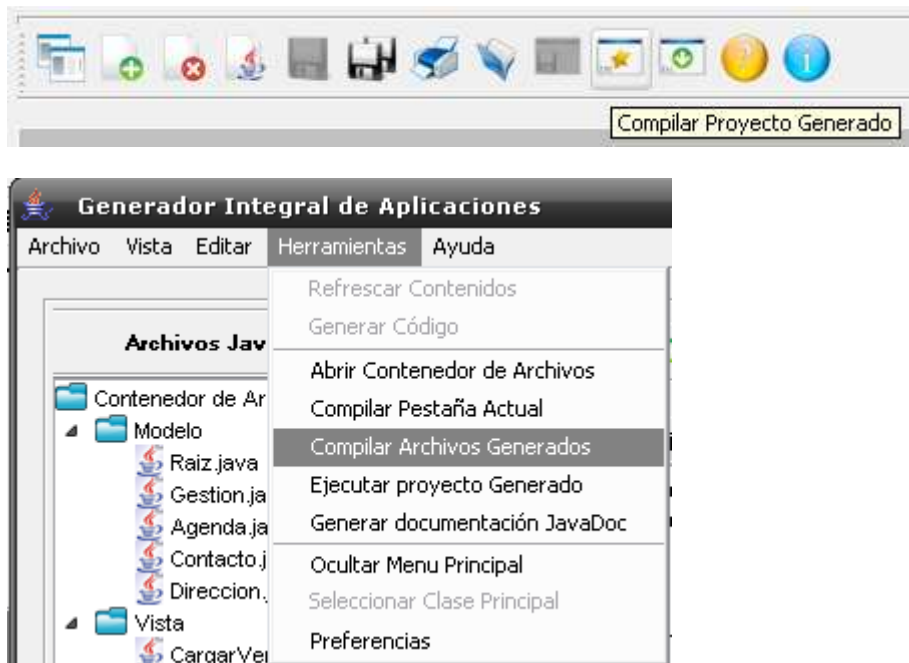
```
public class Agenda
extends java.lang.Object
implements Raiz
```

Field Summary

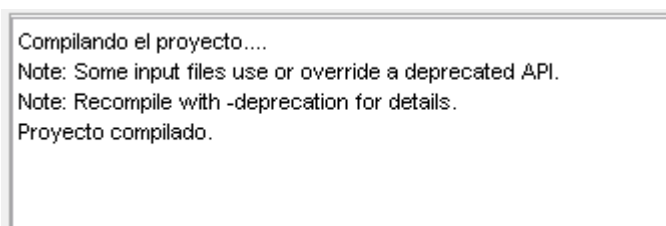
| | |
|----------------------------|-------------------------------|
| protected java.util.Vector | \$v Contactos |
| protected int | dni |
| java.lang.String | fnac |

5.2.16 Compilar Proyecto

Si se quieren compilar todos los archivos del proyecto debemos de acceder al menú Herramientas – Compilar Archivos Generados o acceder al botón ‘Compilar Proyecto completo’ de la barra de herramientas.



El resultado de compilar el proyecto se muestra en el panel de mensajes existente en la parte inferior de la aplicación.

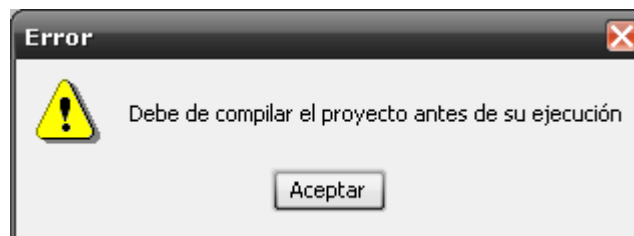


5.2.17 Ejecutar Proyecto

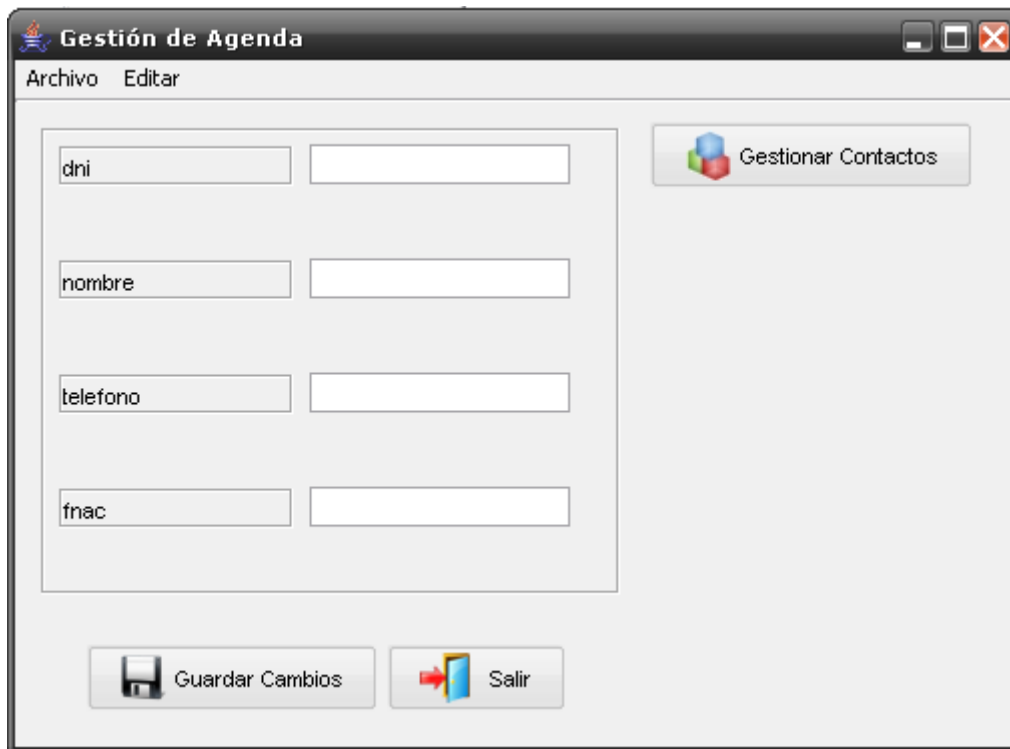
Una vez compilado podremos ejecutar el proyecto accediendo al menú Herramientas – Ejecutar proyecto generado o desde el botón ‘Ejecutar el Proyecto Generado’ situado en la barra de herramientas.



Antes de ejecutar el proyecto debe de haberse compilado en caso de que el usuario haya ejecutado directamente el proyecto se mostrará el siguiente mensaje de error:



A continuación se mostrará la ventana principal del sistema generado desde la cual se podrá acceder a los datos de la clase principal y del resto de clases relacionadas con ella. Para el ejemplo usado en este manual la ventana principal será la siguiente:



Si el usuario desea ver más ejemplos de proyectos generados debe hacer uso del manual técnico que acompaña a este manual de usuario, en la sección ejemplos.

5.2.18 Ayuda de IDE de Java

Si se desea entrar en la ayuda de la aplicación estando en la vista de IDE de Java, deberemos acceder al menú Ayuda- Ayuda IDE de Java, o pulsar el botón 'Ayuda' situado entre los botones de la barra de herramientas.



El sistema abrirá una ventana desde la que se verá una breve explicación de cada una de las opciones del menú y botones de la interfaz de la aplicación.

