

MP_0487. Entornos de desarrollo

UF5. Lenguaje unificado de modelado UML

5.3. Herramientas CASE

Índice

☰	Objetivos	3
☰	Herramientas CASE	4
☰	Dia diagram Editor	5
☰	ArgoUML	7
☰	IBM Rational	9
☰	Office Visio	10
☰	Herramientas UML integradas en IDE	11
☰	Introducción	15
☰	Generación de código a partir de un UML	16
☰	Crear diagramas a partir de código	18
☰	Resumen	21

Objetivos

Con esta unidad perseguimos los siguientes objetivos:

- 1 Conocer cuáles son los tipos de herramientas CASE que se orientan al diseño de aplicaciones bajo la notación UML.
- 2 Conocer las características de algunas de las herramientas de mercado más conocidas.
- 3 El uso de los IDE en el modelado de software.
- 4 Conocer cómo se puede generar código Java desde diagramas y viceversa.

¡Ánimo y adelante!

Herramientas CASE

Las herramientas CASE cubren todo el ámbito de funcionalidades del ciclo de vida software, aunque cada una en su área. Existen pocas que cubran todo el proceso completo. Las que están orientadas a objetos y cubren la parte de requisitos y diseño pueden usar la notación UML para la parte de diseño del sistema.

En la primera unidad de este módulo ya realizamos una clasificación y comprobamos que disponemos de herramientas de creación y asistencia de códigos, realización de pruebas, etc.

El modelado, aunque depende de la metodología que estamos usando, se suele hacer en las fases tempranas del desarrollo y son consideradas herramientas de *Upper Case*.



Las herramientas *Upper Case* ayudan en las fases de planificación, análisis de requisitos y estrategia del desarrollo usando, entre otros, diagramas UML.

Dia diagram Editor

Dia diagram Editor es un software gratuito que nos permite realizar cualquier diagrama UML que deseemos, proporcionando asistentes y ayudas para una correcta implementación.

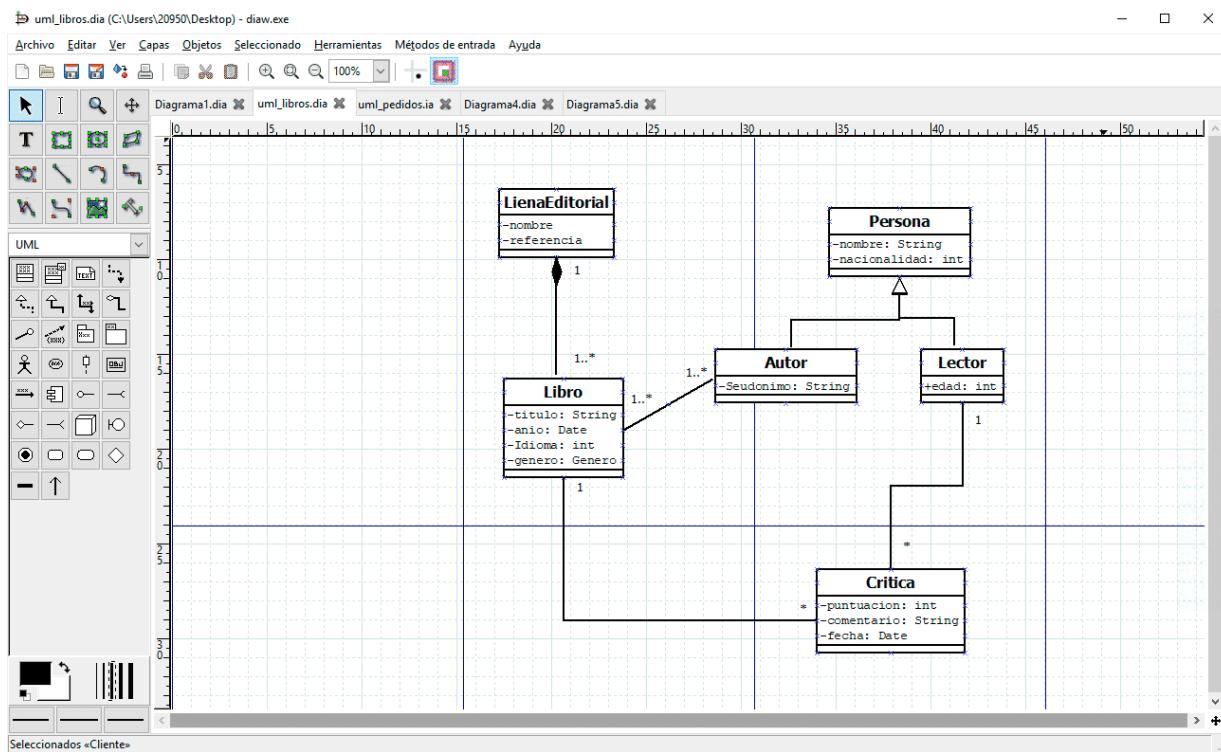
Disponible para las plataformas Windows, Mac y Linux, es una excelente opción para la diagramación UML.

El modelado es multilenguaje, permitiéndonos realizar los diagramas independientemente del lenguaje de programación que vayamos a utilizar.

Dia diagram Editor

Puedes acceder a la página oficial del programa para descargarlo.

[DESCARGA](#)



Captura pantalla software Dia.

A continuación tienes [un vídeo](#) donde se realiza la implementación de un diagrama de casos de uso y de clases.

ArgoUML



ArgoUML es un proyecto de código abierto para realizar la diagramación UML para proyectos basados en Java.

Disponible para sistemas operativos Windows, es una completa aplicación para la realización de nuestros diagramas. Como peculiaridad, **tiene la opción de generar el código Java partiendo de nuestros diagramas**.

Página oficial de ArgoUML

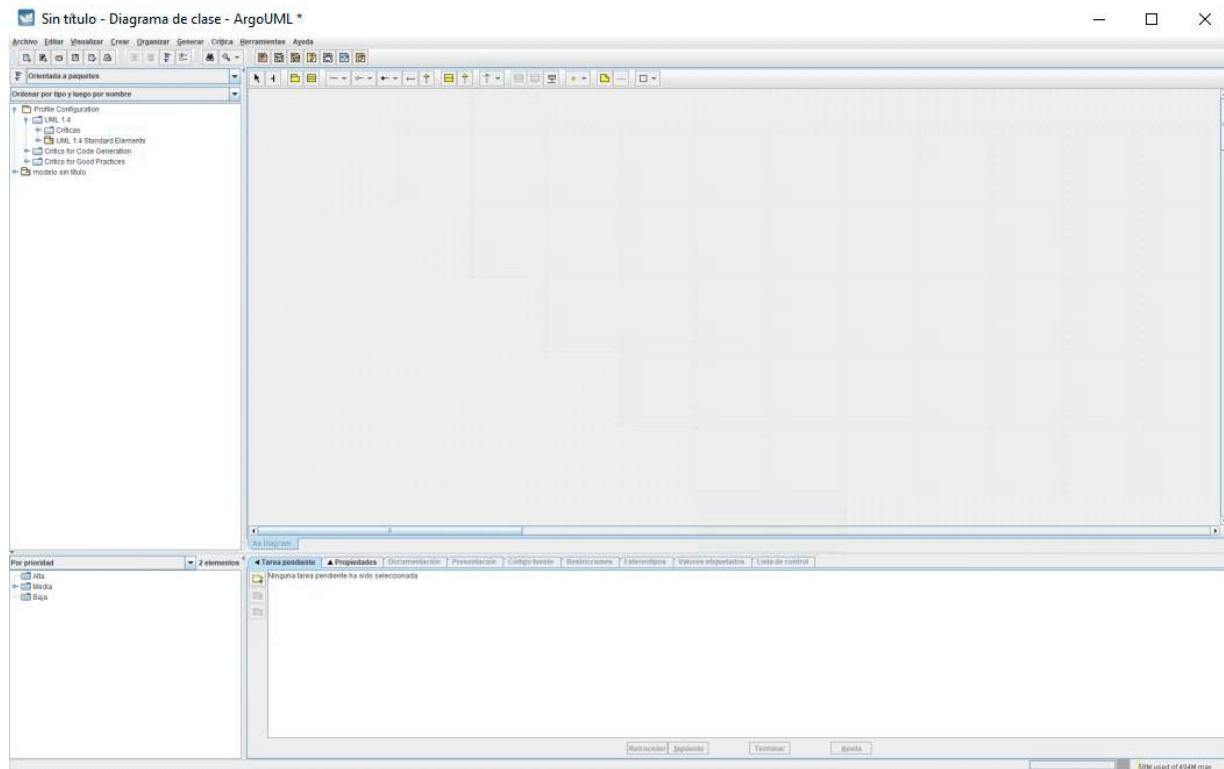
Puedes encontrar documentación, descargar la herramienta, etc.

[ARGOUML](#)

Manual de usuario ArgoUML

Accede al manual de usuario.

[DOCUMENTACIÓN](#)



En el [siguiente vídeo](#) puedes ver cómo se crea un diagrama de clases utilizando ArgoUML.

IBM Rational

IBM Rational es un software **multiplataforma** orientado a desarrolladores de **Java**, totalmente integrado con WebSphere y Eclipse.

Destaca el soporte para JEE, además del uso de patrones e ingeniería de ida y vuelta, **posibilitando la creación de código a partir de diagramas o viceversa**.

Página del producto

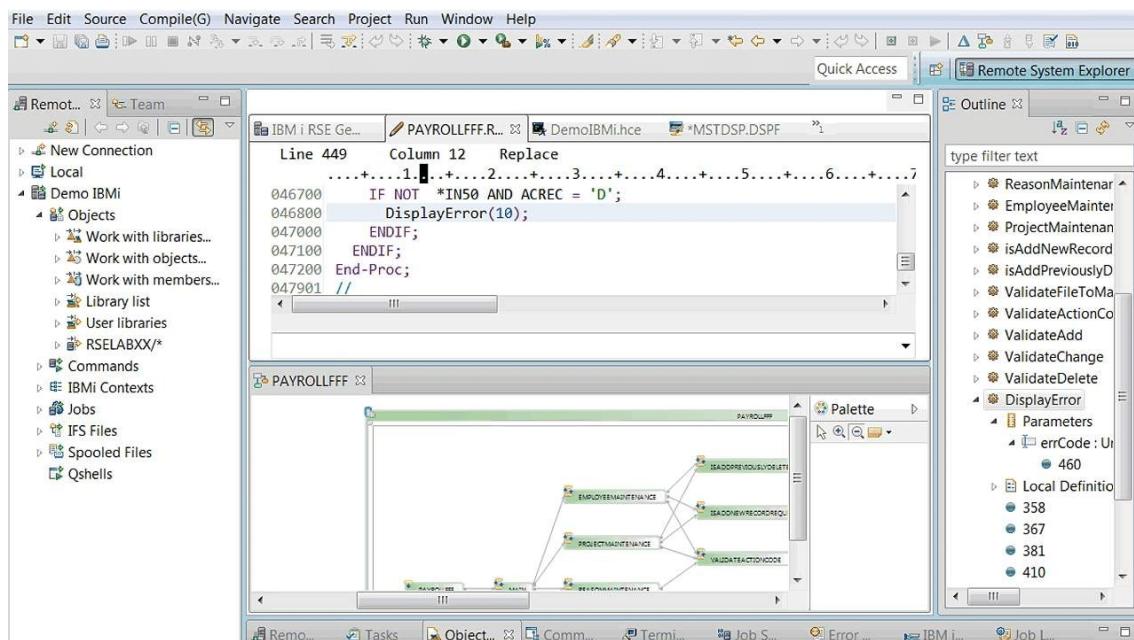
IBM Software solutions.

Manual

IBM Support.

IBM

MANUAL



Office Visio

Office Visio es una herramienta de Microsoft para realizar multitud de diagramas con capacidades de dibujo UML que soporta la ingeniería directa e inversa para C++ y C#, además de Visual Basic .NET.

Esta aplicación es de pago y está muy extendida en las empresas.

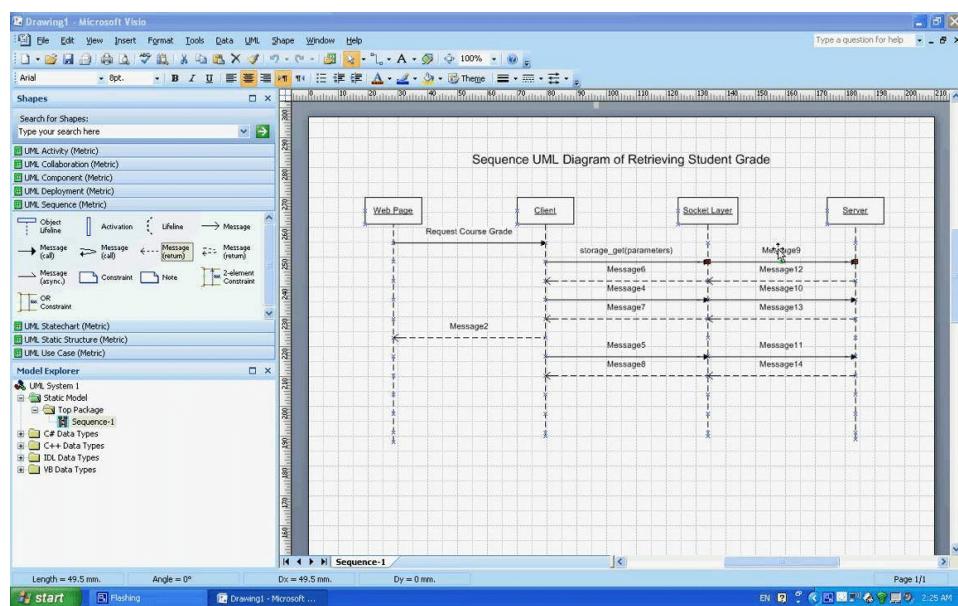
Página de producto

Trabajo de forma visual. Diagramación de forma sencilla.



Manual de Visio

Guía de Visio para principiantes



Vista del programa Office Visio.

Herramientas UML integradas en IDE

Las herramientas **UML que funcionan en los IDE** suelen estar construidas como *plugins*.

Existen muchas herramientas en el mercado que cubren todas las capacidades de desarrollo de la metodología UML. Tales herramientas se fabrican específicamente para generar los diagramas y documentación de los proyectos y están especializadas en estas funciones.

El **modelado UML** permite realizar el **diseño del sistema** a implementar siempre que esté orientado a objetos.

Los **IDE o entornos de desarrollo integrados** son las herramientas en donde los programadores desarrollan el código fuente de los proyectos y, por lo tanto, conforman el siguiente paso dentro del ciclo de vida del software, justo después del diseño del proyecto.

Algunos de los más conocidos son:

Para NetBeans

Visual Paradigm for NetBeans IDE

El *plugin* [Visual Paradigm for NetBeans IDE](#) ofrece una amplia gama de características que incluyen:

- Utilidad de diseño gráfico para diagramas de clases y diagramas de actividad.
- Capacidades de colaboración en equipo.
- Características de generación de código para Java, C ++ y PHP .

JDeveloper

[**JDeveloper**](#) es un IDE gratuito suministrado por Oracle Corporation. Ofrece funciones para el desarrollo en Java, XML, SQL y PL/SQL, HTML, JavaScript, BPEL y PHP. **JDeveloper** cubre el ciclo de desarrollo completo, desde el diseño hasta la codificación, depuración, optimización y creación de perfiles de implementación.

Las capacidades completas de modelado que proporciona JDeveloper incluyen:

- Diagramas UML como casos de uso, de clase, de actividad y los diagramas de secuencia.
- Modeladores de dominio específico, como diagramas de flujo JSF, EJB diagramas, modelado de base de datos y editor de esquemas.

PlantUML

[**PlantUML**](#) es una herramienta de código abierto que permite a los usuarios crear diagramas UML a partir de un lenguaje de texto sin formato. El lenguaje de PlantUML es un ejemplo de un lenguaje específico de aplicación.

Se utiliza el software Graphviz para diseñar los diagramas. Se ha utilizado para permitir que los estudiantes ciegos trabajan con UML. PlantUml también ayuda a los ingenieros de software ciegos a leer los diagramas UML.

NetBeans UML Plugin

[**PUML**](#) es un *plug-in* para NetBeans y está basado en una biblioteca visual.

El objetivo es crear una herramienta fácil de usar y útil para los desarrolladores. Los diseñadores se centran en el diseño de la aplicación y los desarrolladores se centran en el código. Los diseñadores y analistas de aplicaciones de diseño, usando UML, pueden generar código directamente desde la propia herramienta.

Para Eclipse

Eclipse es uno de los IDE que dispone de más *plugins* para el modelado UML, algunos de los que aparecen disponibles en *Eclipse Market Place* son:

UML Designer

Utiliza el metamodelo estándar UML2 proporcionado por la **Fundación Eclipse** e implementa los siguientes diagramas UML genéricos:

- Jerarquía de paquetes.
- Diagrama de clase.
- Diagrama de componentes.
- Diagrama de estructura compuesta.
- Diagrama de despliegue.
- Diagrama de casos de uso.
- Diagrama de actividad.
- Diagrama de estados.
- Diagrama de secuencia.
- Diagrama de perfiles.

Este paquete genera el código de los diagramas UML y es compatible con el generador de código UML to Java.

Puedes acceder al manual desde el siguiente enlace:

<http://www.umldesigner.org/tutorials/tuto-getting-started.html>

UML to Java Generator

Este generador permite producir código fuente de Java a partir de un modelo UML. Este módulo utiliza el metamodelo UML2 estándar proporcionado por la **Fundación Eclipse** y, como tal, es compatible con cualquier herramienta UML que utilice este estándar como UML Designer.

Puedes acceder al manual desde el siguiente enlace:

<https://www.eclipse.org/umlgen/>

MDT: UML2 Modeler

MDT: UML2 Modeler es una herramienta que se usa para el modelado de sistemas integrales con el uso del lenguaje UML. Su funcionalidad principal es **modelar arquitecturas de sistemas en el entorno MDT**, así como **generar dinámicamente código fuente de Magik**.

Ayuda a combinar el trabajo de los arquitectos que trabajan con el lenguaje UML con las tareas asignadas a los desarrolladores Magik. Los arquitectos pueden ampliar los sistemas existentes por nuevos elementos mediante la creación de artefactos específicos para los desarrolladores directamente del entorno de MDT.

Puedes acceder al manual desde el siguiente enlace:

[https://www.mdt.net/pe/help/index.jsp?
topic=%252Fpl.com.astec.mdt.uml.help%252Fhtml%252Fintro%252Ftoc.html](https://www.mdt.net/pe/help/index.jsp?topic=%252Fpl.com.astec.mdt.uml.help%252Fhtml%252Fintro%252Ftoc.html)

UML Generators 1.0.0

Es un nuevo proyecto Eclipse que proporciona componentes que unen automáticamente la brecha entre los modelos UML y el código fuente. Ya sea por extracción de datos de los modelos UML (UML y perfiles o modelos de decoración) para producir el código fuente o el código fuente de ingeniería inversa para producir modelos UML.

Esta versión proporciona un generador de código Java y C (de ida y vuelta incluido). UML Generator 1.0.0 utiliza Acceleo, una implementación del Object Management Group (OMG) que integra el **Modelo de texto** en el idioma estándar MTL (MOF).

Es gradual y se puede ampliar gracias a su sistema de primer orden. El Unified Modeling Language (UML) es la especificación más utilizada en el OMG.

Puedes acceder al manual desde el siguiente enlace:

https://wiki.eclipse.org/Eclipse_UML_Generators

UML Class Diagram Editor for Eclipse

Proporciona una herramienta ligera, pero altamente profesional, de software para la creación y edición de clases y paquetes de diagramas UML con el IDE de Eclipse.

- Editor UML 2.5 de diagramas de paquetes y clases.
- Validación del modelo (OCL).
- Navegación y exploración de modelos.
- Modelado de tareas con Mylyn.
- Importación / exportación del estándar XMI (Rational Software Architect, Topcased, Eclipse UML2).
- Diagrama de exportación (SVG, PNG, JPG, BMP) y edición de clases y paquetes diagramas UML con el IDE de Eclipse.

Puedes acceder al manual desde el siguiente enlace:

<http://www.uml-lab.com/en/uml-lab/tutorials/>

Introducción

Las herramientas UML son capaces de coordinar los elementos programáticos con los diagramas de UML, de forma que una vez diseñado el modelo y generadas las clases o el código correspondiente, estos se coordinen.

No todas las herramientas funcionan de esa forma, hay algunas que solo generan UML, otras que generan código a partir de los diagramas y otras más sofisticadas que sincronizan los cambios realizados en cada una de las partes, el mundo del diseño o la codificación.

De las muchas funcionalidades que se derivan de la inclusión de herramientas UML, una de las más valiosas en equipos de desarrollo modernos es la posibilidad de:

Ingeniería inversa de código a UML

Consiste en revertir una instantánea de la base del código de clases UML y diagramas de clases. Al traer el contenido de código en el modelo UML visual, ayuda a los programadores o ingenieros de software a revisar la aplicación e identificar los posibles errores o deficiencias y también buscar posibles mejoras.

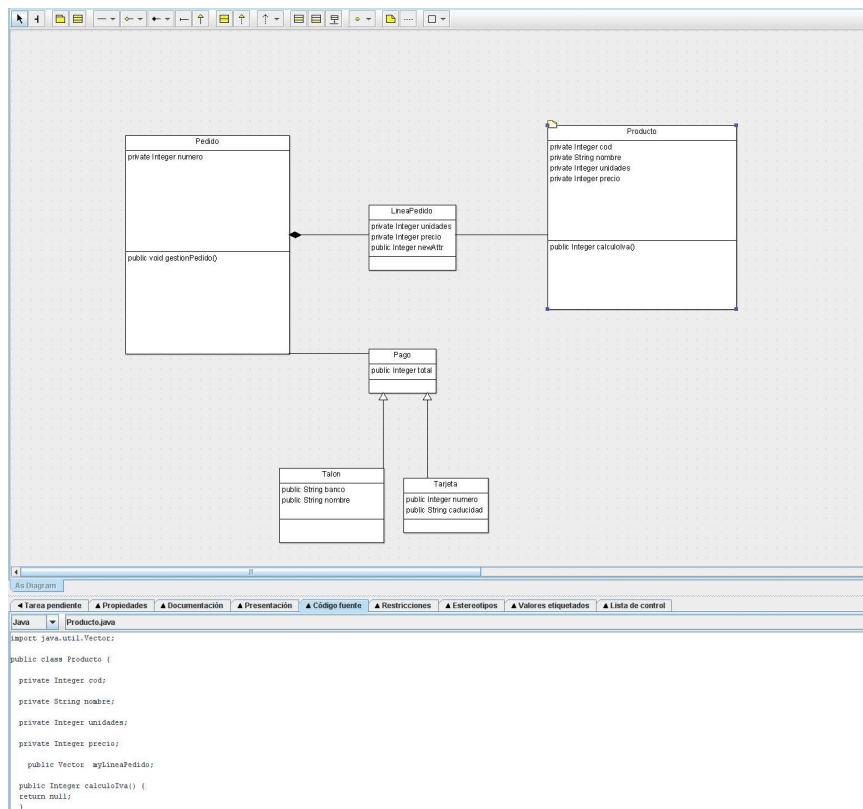
Generación de código UML

Generar código fuente en el lenguaje de programación de trabajo a partir de los diagramas UML definidos.

Generación de código a partir de un UML

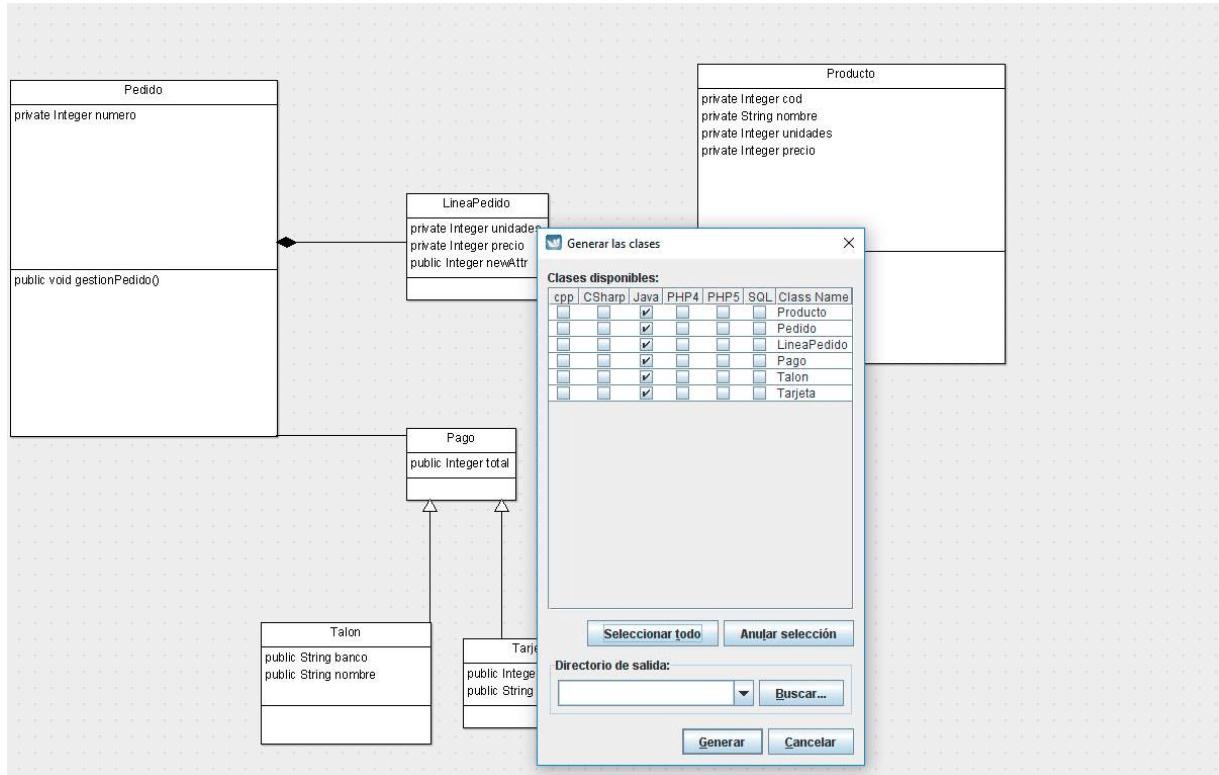
Una de las utilidades que más nos llama la atención de las herramientas CASE de UML, es la característica de **creación de código automático** partiendo de la representación de una clase con sus atributos y métodos.

No todas las herramientas disponen de esta función. En nuestro caso vamos a trabajar con **ArgoUML**. Con este software ya hemos visto que podemos representar diagramas de clases con sus relaciones, al mismo tiempo que podemos ir viendo el código generado en cada clase.



En la pestaña de código fuente, Argo nos muestra el código base de la clase que seleccionemos.

Si deseamos exportar todo el proyecto representado, en el menú tenemos la opción de generar todas las clases (F7). Al hacerlo, nos pedirá una carpeta para guardar los archivos. Esta carpeta puede ser nuestro espacio de trabajo de Eclipse.



Opción de generar todas las clases.

En el [siguiente vídeo](#) puedes ver cómo realizar este proceso.

Crear diagramas a partir de código

Otra operación que puede resultar útil con este tipo de herramientas es **la generación de los diagramas de clases de un código ya creado**.

Imaginemos que tenemos un desarrollo complejo, difícil de entender mirando el código de las clases, ya sea por su extensión o porque está mal documentado. Realizar la conversión a un diagrama puede facilitarnos su comprensión.

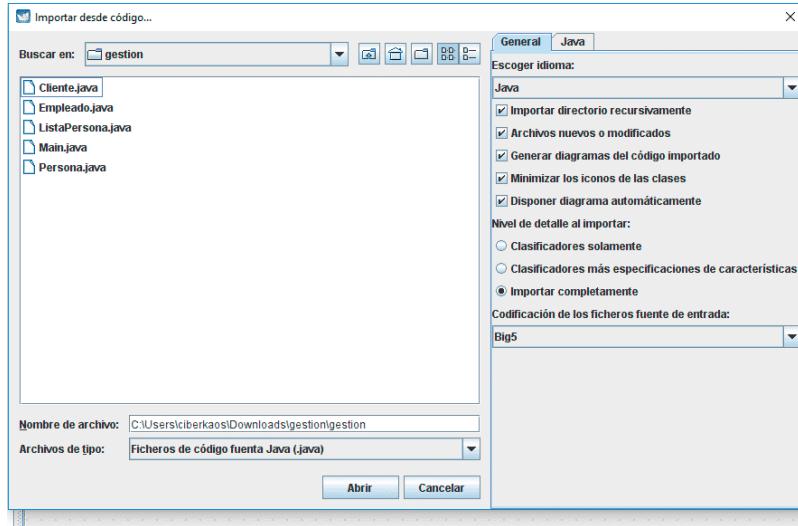
Esta operación resulta extremadamente sencilla con el uso de herramientas como ArgoUML, dado que lo único que tenemos que hacer es abrir el proyecto partiendo de un código (Menú Archivo > Importar desde código).

Vamos a partir de un ejemplo donde tenemos cinco clases con sus relaciones.

Nombre ↑	
	Cliente.java
	Empleado.java
	ListaPersona.java
	Main.java
	Persona.java

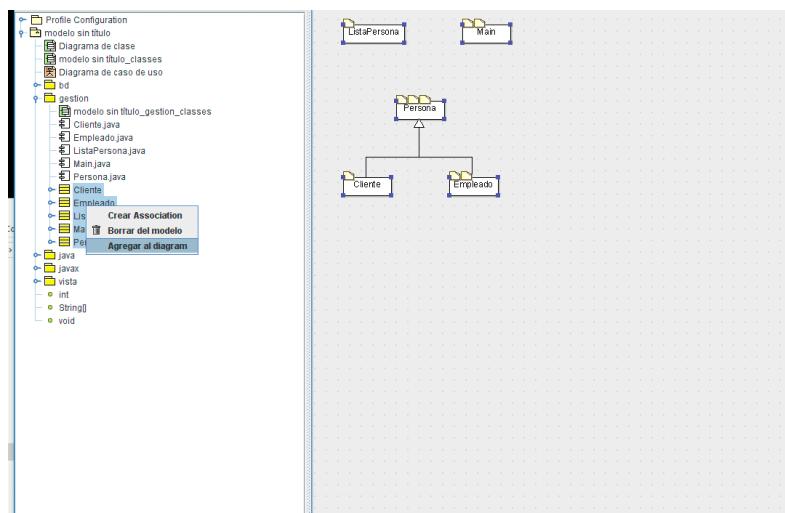
Archivos con el código .java.

Desde ArgoUML, seleccionamos el archivo de código que queremos importar, Menú Archivo > Importar desde código.



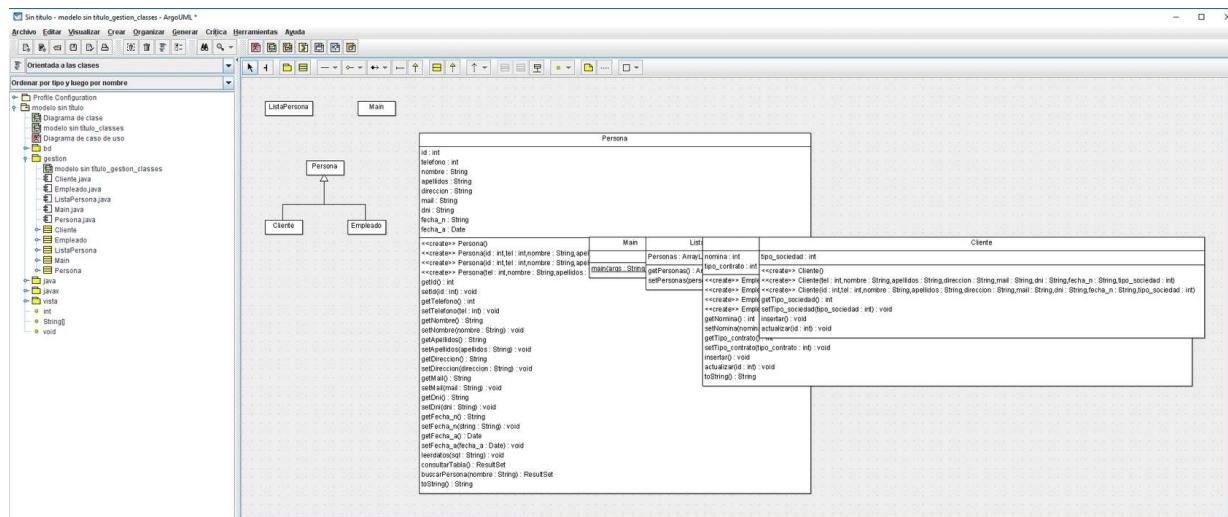
Podemos seleccionar archivos sueltos o carpetas.

Ahora ya tenemos nuestras clases en ArgoUML. A partir de aquí podemos insertar las clases en el área de trabajo y verlas o modificarlas según convenga.



Agregamos las clases que deseamos examinar, Argo nos informa de la relación que tiene cada una de las clases importadas.

Al importar una clase partiendo del código nos muestra todos los métodos y atributos implementados.



Resumen

Has finalizado esta lección.

En esta unidad hemos visto cómo las herramientas UML están preparadas para componer todo tipo de diagramas UML. Muchas de ellas tienen conectores con los entornos de desarrollo y pueden trabajar conjuntamente con ellos. Además, disponen de funciones de ingeniería inversa entre otras muchas opciones.



PROEDUCA