

UD 8

DIAGRAMAS DE ESTRUCTURA

ENTORNOS DE DESARROLLO 20/21

CFGS DAW

PRÁCTICA 00

DIAGRAMAS DE CLASES EN VISUAL PARADIGM

Revisado por:

Sergio Badal

Autores:

Cristina Álvarez, Fco. Javier Valero Garzón, M.ª Carmen Safont, Paco Aldarias

Fecha:

04/02/21

Licencia Creative Commons



Reconocimiento - NoComercial - CompartirIgual (by-nc-sa): No se permite un uso comercial de la obra original ni de las posibles obras derivadas, la distribución de las cuales se debe hacer con una licencia igual a la que regula la obra original.

1. JUSTIFICACIÓN DE LA PRÁCTICA

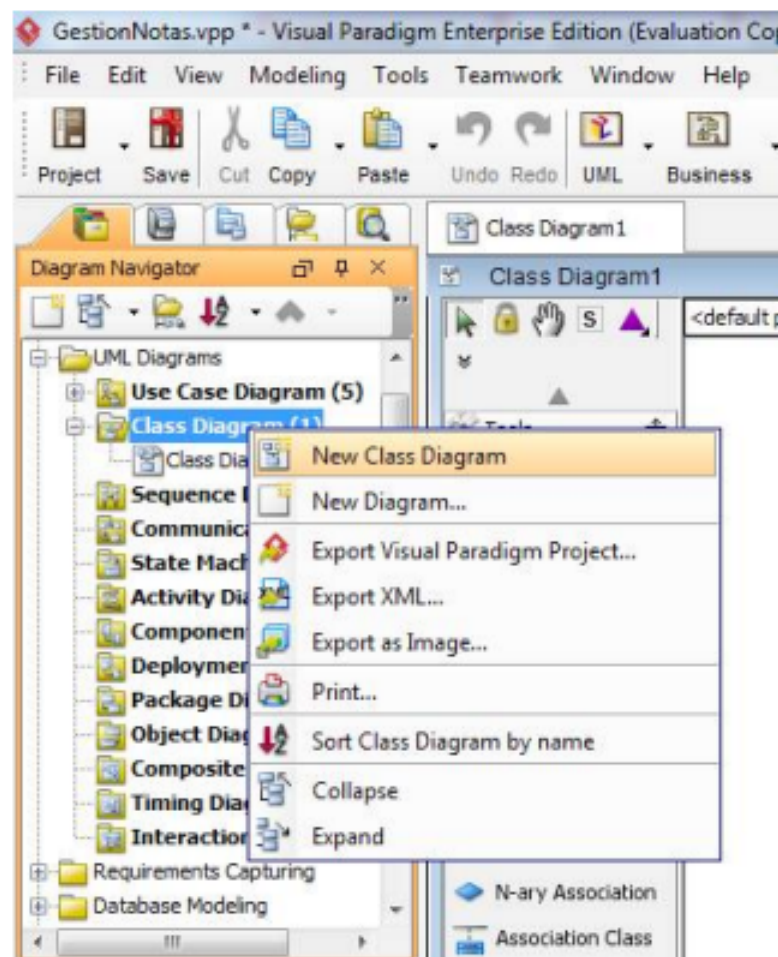
El diagrama de clases es vital para la definición del sistema. Las clases representan entidades de forma estática. Una clase puede contener atributos, propiedades y métodos.

En esta práctica vamos a enseñarte a cómo realizar diagramas de clases con Visual Paradigm en su versión de escritorio, que ya instalamos el tema anterior, **aunque puedes usar la versión online o cualquier otro editor.**

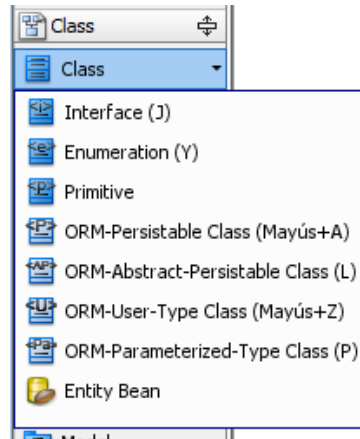
2. CREANDO NUESTRO PRIMER DIAGRAMA DE CLASES

Vamos a ver los pasos para crear un diagrama de clases simple usando el ejemplo de Alumno de los apuntes De teoría.

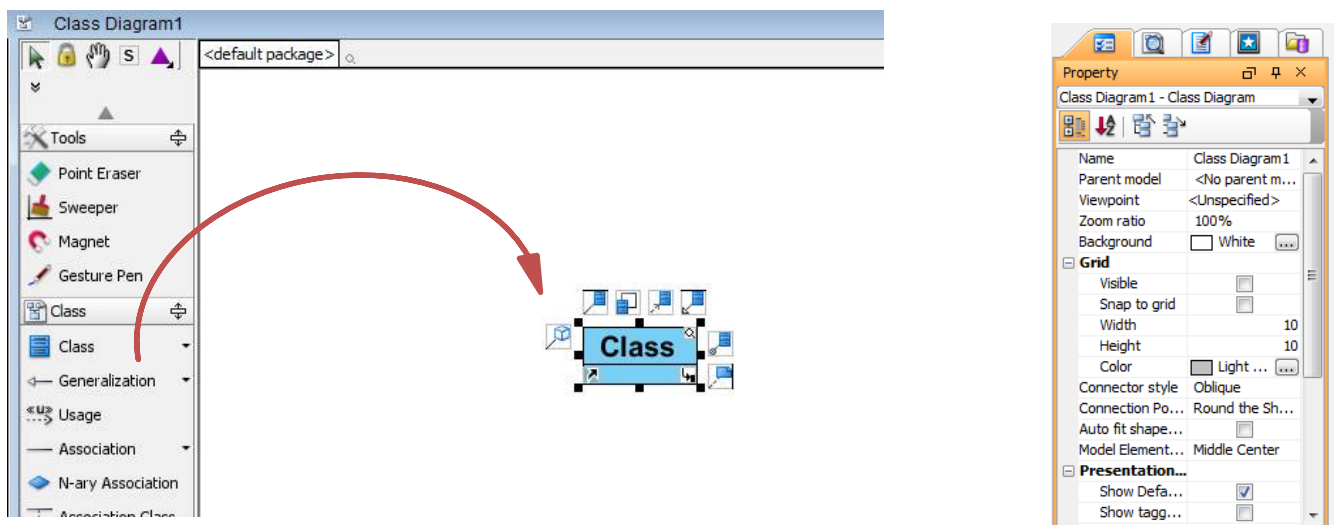
0. En el panel de diagramas, sobre Class Diagram haz botón derecho > New use case diagram. Veremos el área de trabajo.



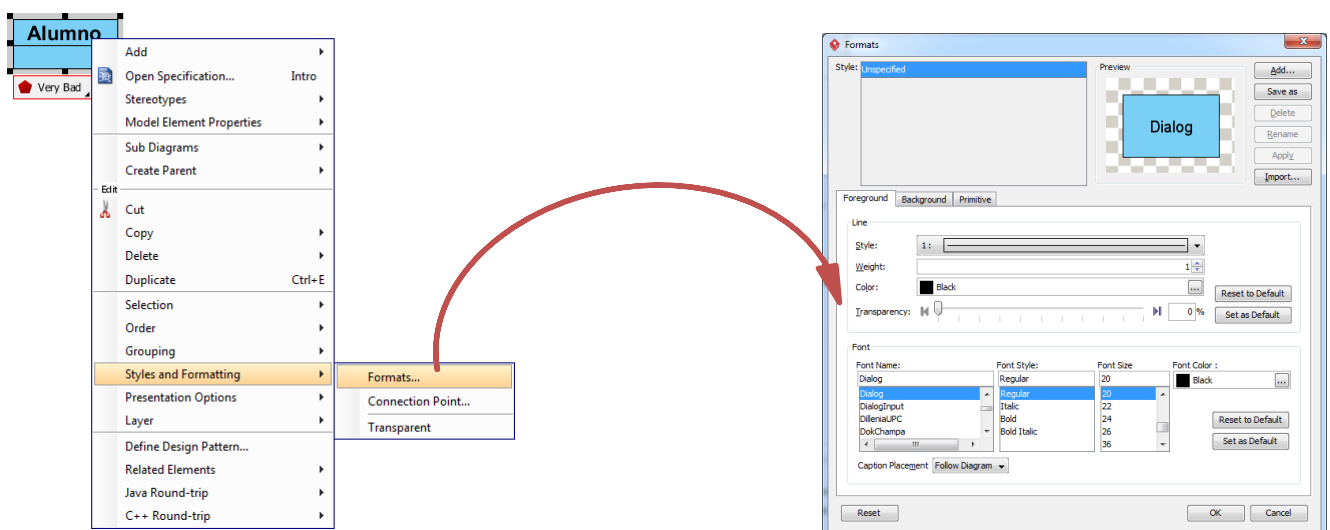
1. En el panel izquierdo tenemos todos los elementos necesarios para hacer nuestro diagrama, agrupados por tipo de elemento. Por ejemplo, si haces clic en “Class” verás que aparecen los distintos tipos



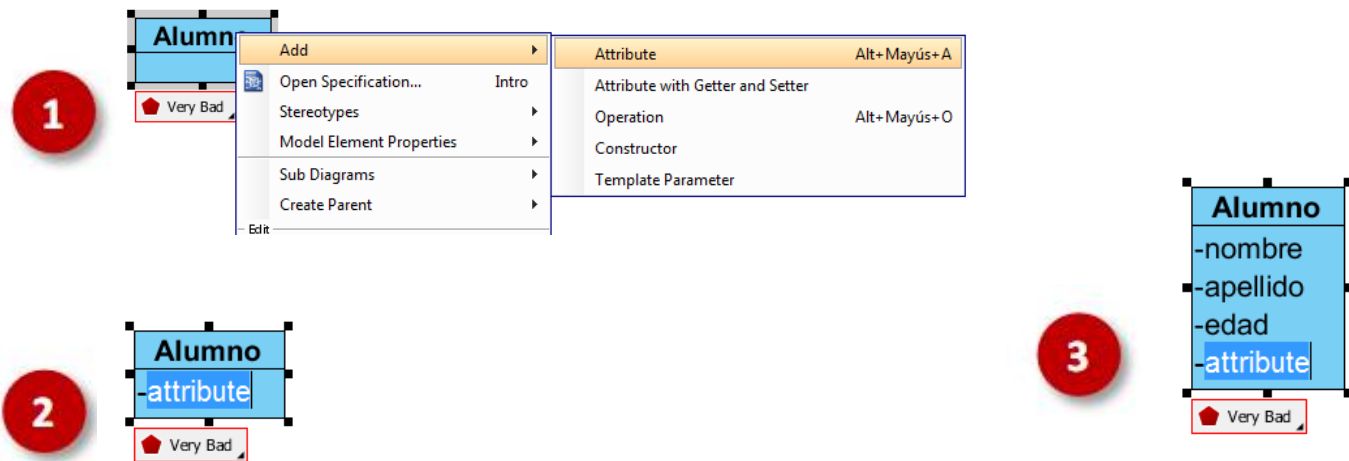
Arrastramos la primera al área de dibujo. Cámbiale el nombre por Alumno. Recuerda que luego podremos moverlo a la zona que queramos. Además, en el panel de propiedades vemos las características de la figura:



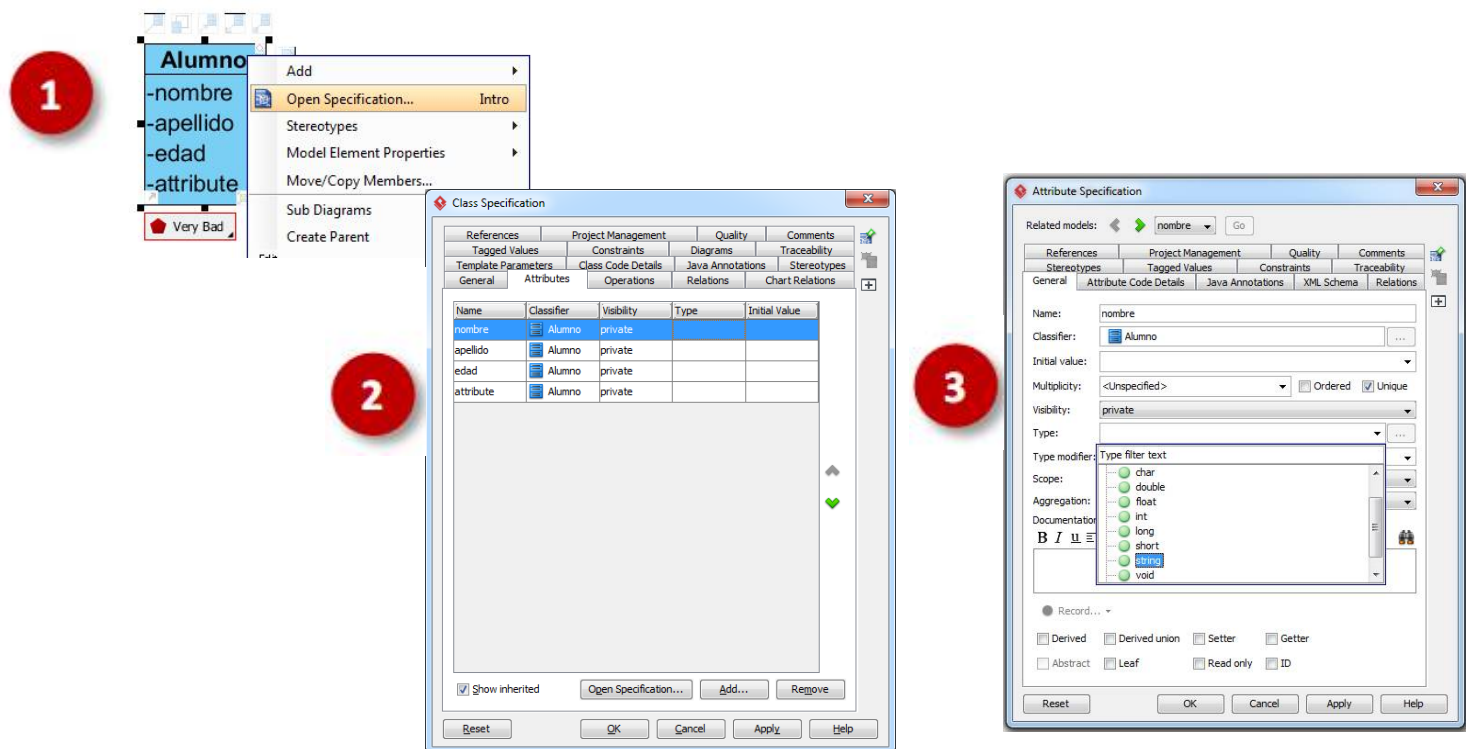
2. Vamos a editar la forma Pulsamos en Botón derecho > Styles and Formatting > Formats. Veremos la pantalla de edición del formato visual, para cambiarle el tipo de letra, color de fondo, etc. Prueba a cambiarle el tamaño de la letra.



3. Vamos a añadir un **atributo**. Para ello, seleccionar la clase a la que se lo queremos añadir (Alumno) y pulsa el botón derecho> Add-Attribute. Podremos ir escribiendo los atributos y añadiendo nuevos pulsando INTRO.



Vamos a especificar más nuestros atributos. Para ello, selecciona la clase y pulsa Botón Derecho > Open Specification. Veremos una pantalla donde podemos cambiar todo sobre la clase, y en la pestaña Attributes podremos poner los tipos de atributos. Puedes cambiar la información directamente en la tabla o editando el atributo. Déjalo como se ve en la figura:



Name	Classifier	Visibility	Type	Initial Value
nombre	Alumno	private	string	
apellido	Alumno	private	string	
edad	Alumno	private	int	
attribute	Alumno	private		

En la ventana del punto 3 (botón derecho sobre el atributo > Open Specification...>General) aparece la configuración por defecto del atributo.

Podemos darles un **valor por defecto** a los atributos (valor inicial para cuando el objeto sea instanciado). Para ello, en la ventana del punto 3 podemos ir a "Initial value" e introducir el que queramos.

También podemos marcar su navegabilidad, en el desplegable de la misma ventana así como su multiplicidad.

4. Ahora vamos a añadirle las operaciones que tiene la clase. Se hace de forma similar al paso anterior, sólo que ahora las operaciones pueden tener parámetros de un tipo y devolver información de otro tipo:

The screenshots show the following steps:

- Class Specification:** A window showing the class 'Alumno' with attributes 'pNombre: String' and 'pEdad: int'. The 'Operations' tab is selected.
- Parameter Specification:** A window for specifying the parameters of the 'setNombre' operation. It shows 'Name: pNombre', 'Operation: setNombre', 'Type: String', and 'Direction: inout'.
- Operation Specification:** A window showing the operation 'setNombre' in a table with columns 'Name', 'Type', 'Default Value', and 'Direction'.

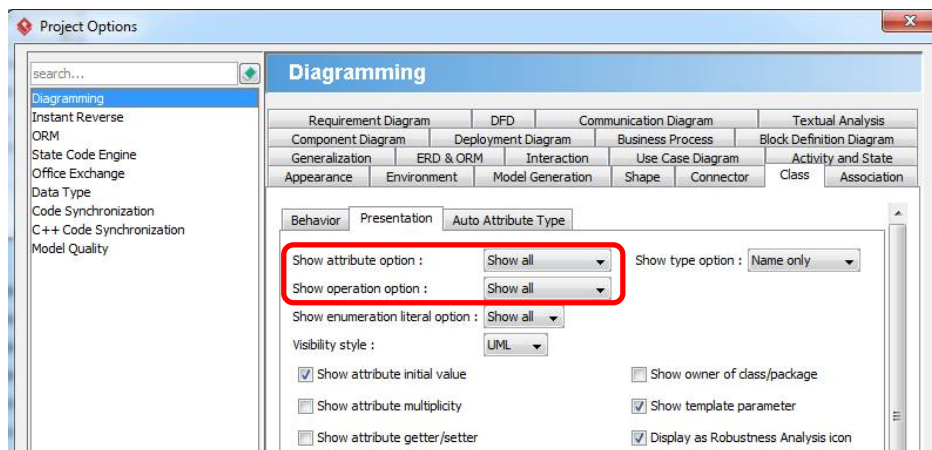
Red arrows indicate the flow from the 'Class Specification' window to the 'Parameter Specification' window, and then to the 'Operation Specification' window.

Name	Type	Default Value	Direction
pNombre	String		inout

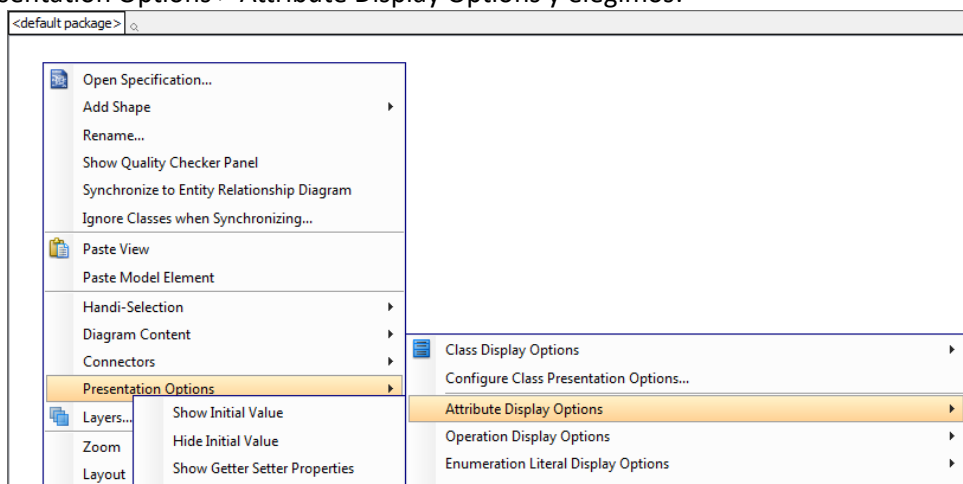
Alumno

- nombre : string
- apellido : string
- edad : int
- attribute
- +setNombre(pNombre : String)
- +setEdad(pEdad : int)
- +toString() : string

5. **Visibilidad.** Como habrás visto, en UML se crean las clases como un rectángulo dividido en tres partes: nombre de la clase, atributos y métodos. La visibilidad de los elementos de la tabla se muestra con los símbolos que ya conocemos ("+" público, "-" privado, "#" protegido). Podemos editar la visibilidad en la ventana de propiedades de cada elemento o en general (para el entorno de trabajo, para el diagrama o para la clase). Si queremos editar la visibilidad para todo el workspace, vamos al menú **Tools > Options > Diagramming > Class > Presentation**:

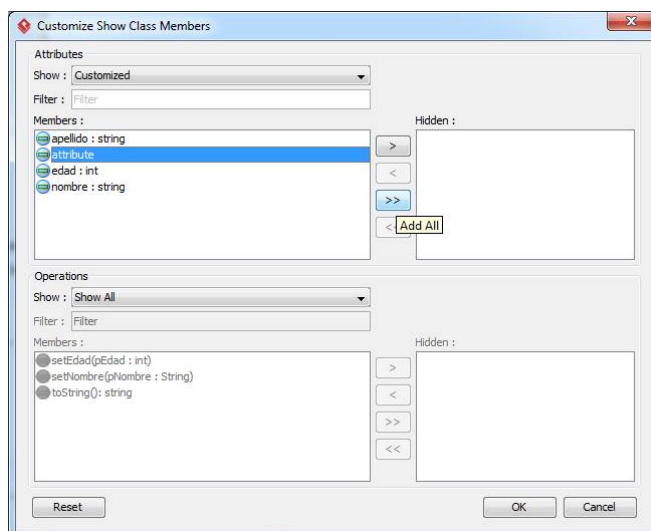


Si queremos marcar la visibilidad sólo para este diagrama, hacemos botón derecho una zona blanca del diagrama > Presentation Options > Attribute Display Options y elegimos:



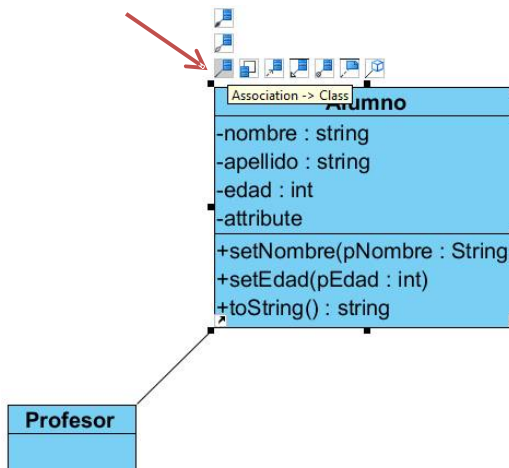
Si nos interesa marcar la visibilidad por clases, hacemos botón derecho sobre ella > Presentation Options > Attributes y elegimos.

Si queremos actuar sobre los atributos de una clase hacemos botón derecho en la clase > Presentations Options > Attributes > Customized y del desplegable de la ventana que aparece elegimos Customized y ahora sólo tenemos que seleccionar qué atributos queremos mostrar u ocultar, pudiendo pasarlos en bloque al otro lado:

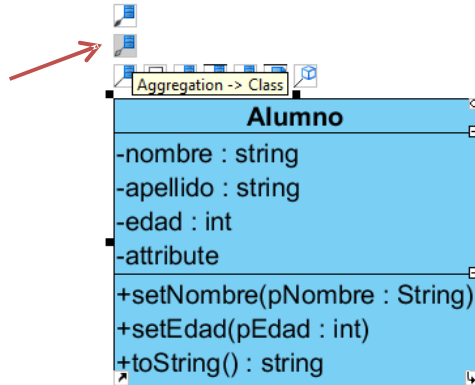


Finalmente, podemos ir atributo a atributo entrando en su pantalla de edición y eligiendo la visibilidad en el campo correspondiente.

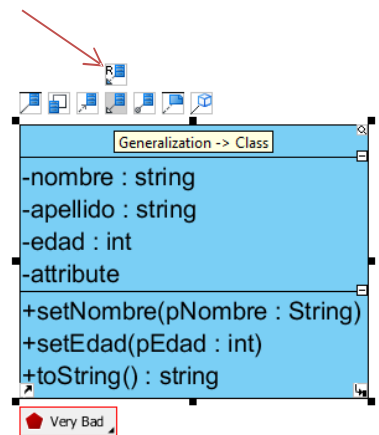
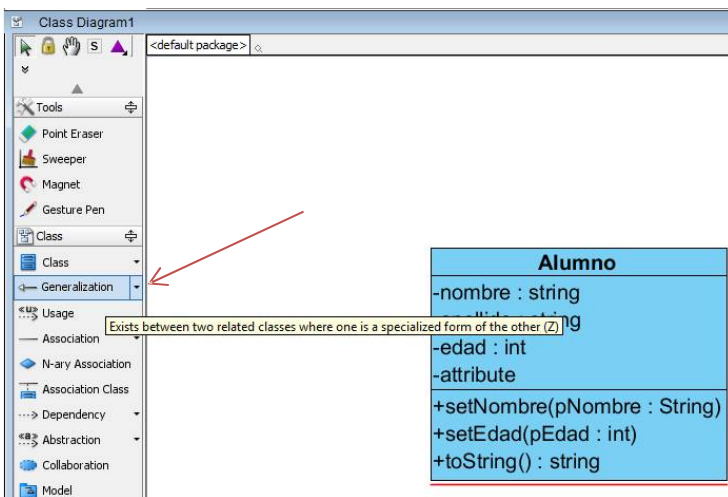
6. Vamos a crear una **asociación**. Elige el icono adecuado y arrastra a otra zona de trabajo. Aparecerá una clase vacía. También podemos estirar hasta una clase existente para conectarnos con ella.



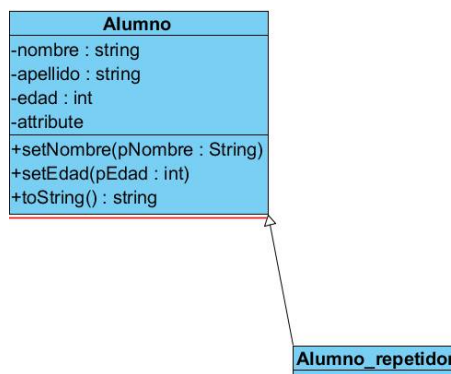
7. Para una **agregación** elegiríamos el icono superior al seleccionado:



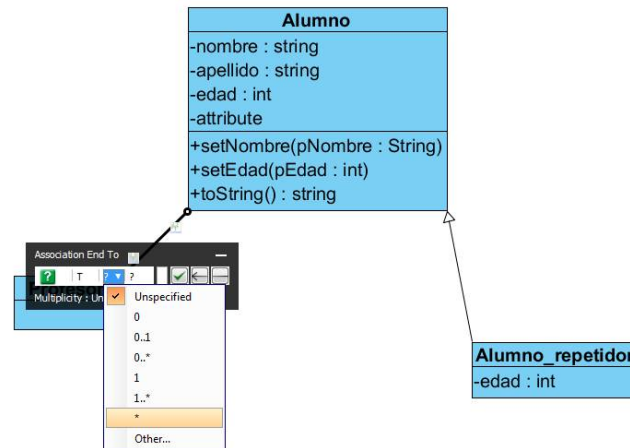
8. Vamos a crear una **generalización**. Para ello, al igual que en el DCU podemos usar la barra de herramientas lateral o desde las opciones que aparecen en la misma clase. Se puede crear una generalización a una clase que ya exista o a una nueva.



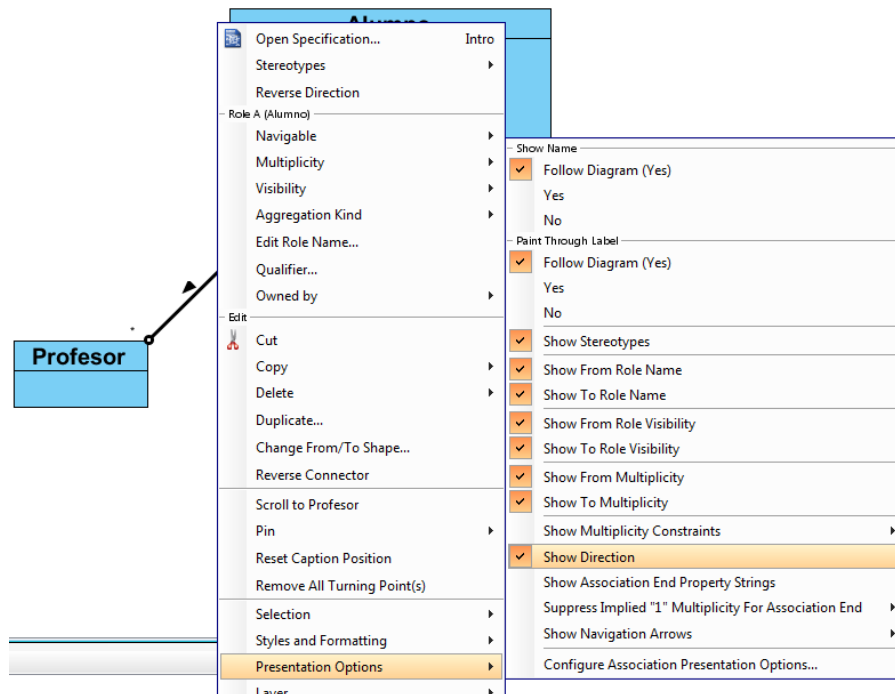
Creamos un Alumno repetidor:



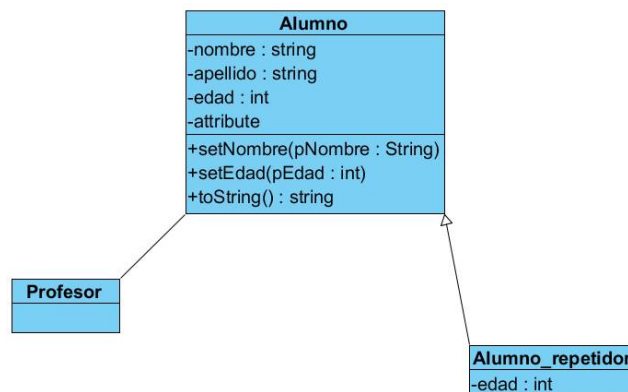
9. Una vez creadas las relaciones podemos editarlas haciendo clic en ella o yendo a propiedades. Vamos a ponerle la **multiplicidad**, para ello, selecciona el extremo de la asociación (profesor), haz botón derecho > Multiplicity y elige del listado o simplemente doble click y despliega. Lo haremos para ambos extremos de la relación.



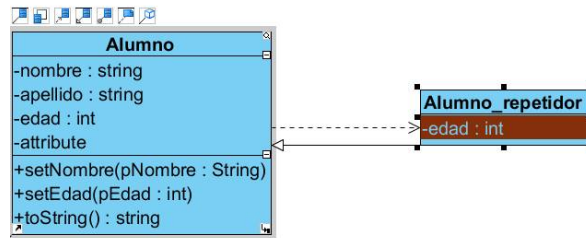
10. Ahora pongamos la **dirección** de la asociación. Para ello, de nuevo doble click o botón derecho sobre la relación > Presentation Options > Show Direction:



11. Se pueden **reordenar** los elementos de la tabla. Para ello, simplemente pincha en él y arrástralo a donde quieras. También se pueden **copiar** elementos de una tabla a otra, arrastrándolos mientras pulsas CTRL. Aparecerá un símbolo + en nuestro puntero del ratón indicando que vamos a copiar:



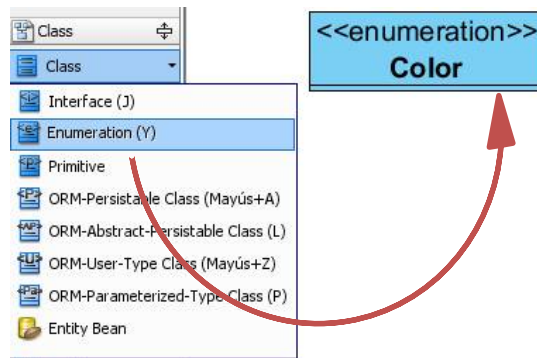
12. Podemos relacionar atributos y operaciones de diferentes clases. Para ello, selecciona el tipo de relación a usar en la barra de herramientas, haz clic en el elemento a relacionar y arrastra hasta el otro elemento a relacionar:



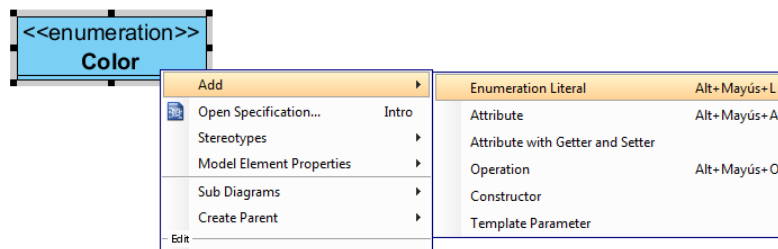
13. Vamos a crear el tipo **enumeración**. Es un tipo de datos especial que consta de una serie de valores predefinidos (literales de enumeración), como:

- Color (rojo, verde, azul, amarillo...)
- Orientación (norte, sur, este, oeste)
- Encendido (on, off)

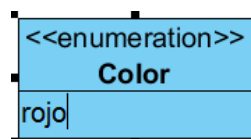
Para crearla, selecciónala de la barra de herramientas y arrástrala a la zona de trabajo.



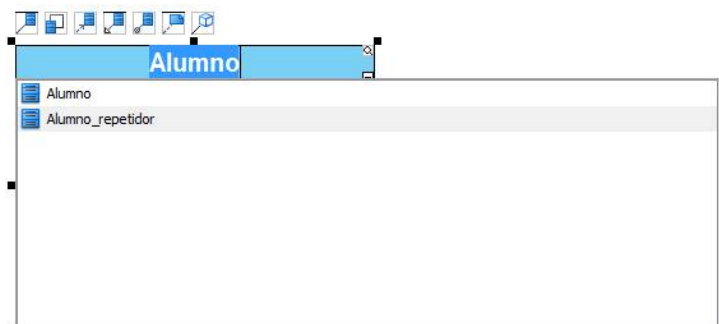
Para añadir un literal, haz botón derecho sobre ella > Add > Enumeration Literal:



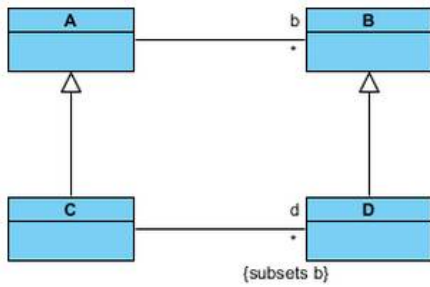
Ponle el nombre y confirma:



14. Ahora vamos a poner las **terminaciones** de las relaciones. Estos permiten crear múltiples vistas rápidamente del mismo modelo de clase. Cuando creamos una clase o la renombramos, aparecen en la lista el resto de clases y se puede filtrar escribiendo las primeras letras:



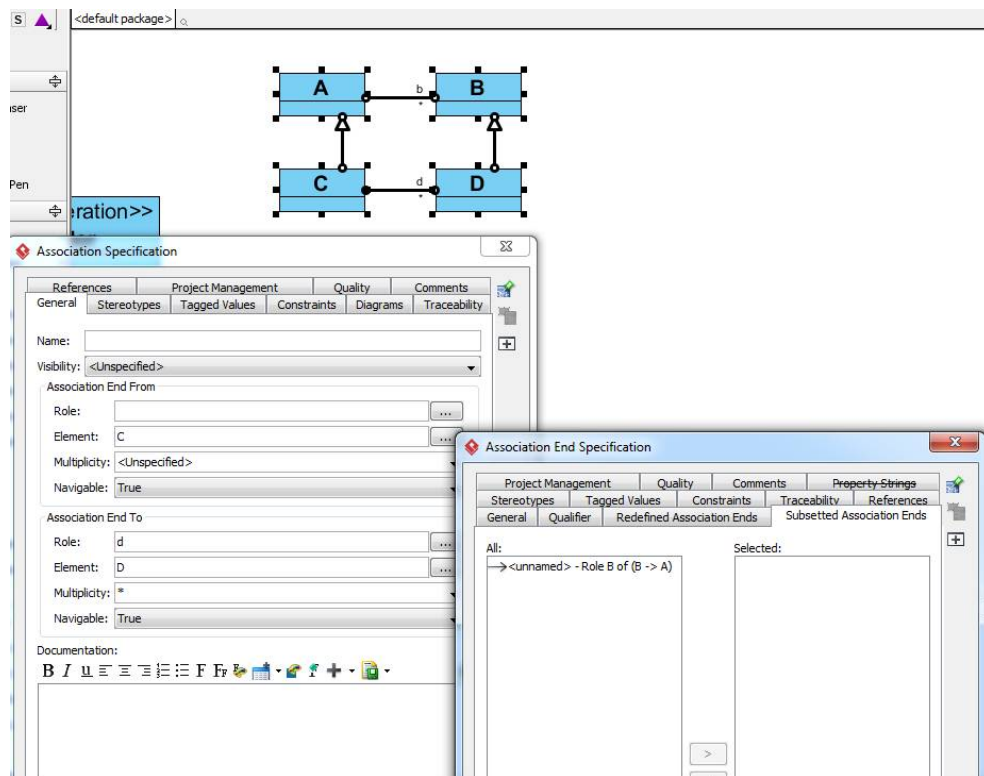
Podemos tener un **subconjunto de terminaciones de asociación** como se ve en la figura:



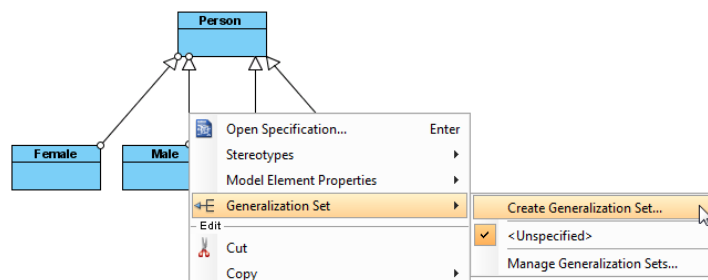
Donde el subconjunto sobre la “d” indica que la colección “d” (que es una instancia de la clase “C”) es un subconjunto de la colección “b”(instancia de la clase “A”).

Para definir un subconjunto en una terminación de asociación hacemos botón derecho en la terminación de la asociación > Open Specification... > General. Ahí elegimos la terminación que queremos definir como un subconjunto. Vamos a “Role” y hacemos clic en “...”. Se abre una ventana de “Association End Specification”, vamos a la pestaña “Subsetted

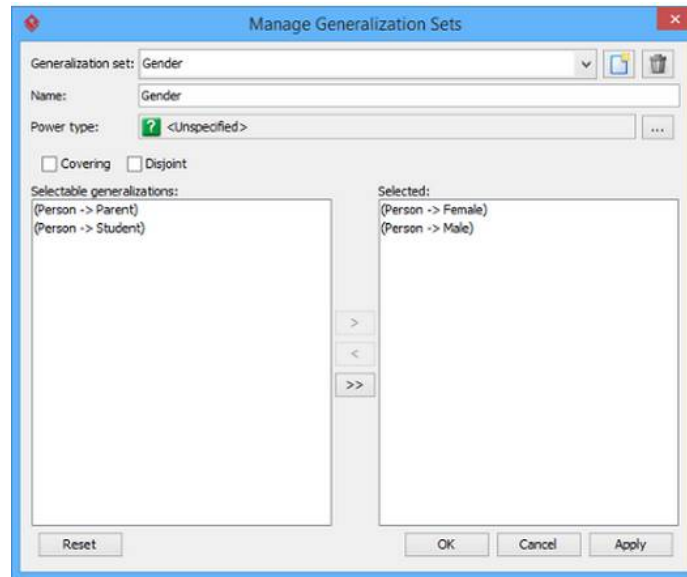
Association Tab”. Haz clic en el rol que quieres definir como subconjunto. Si no ves ninguno, revisa el patrón de tu diseño. Confirma todo y ahora vamos a hacer que sea visible. Para ello, haz clic de nuevo en la terminación, botón derecho > presentation options > show association and property strings.



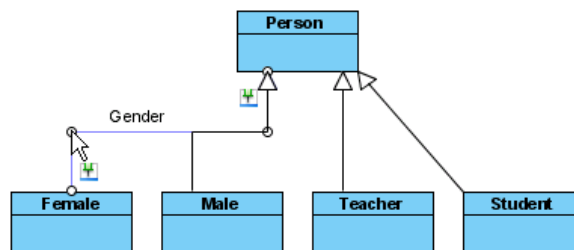
- Podemos tener un **kit de generalización**. Define un conjunto particular de las relaciones de generalización donde una relación a un clasificador general (o superclase) puede ser dividida usando subtipos específicos. Para definir un conjunto de generalizaciones selecciona todas las que quieras incluir y haz botón derecho > Generalization Set > Create Generalization Set..... Vamos a verlo con un ejemplo nuevo:



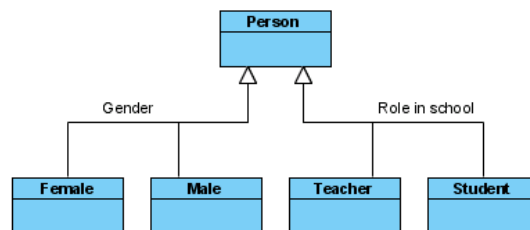
Ahora le ponemos nombre al kit de generalización en la ventana que se abre:



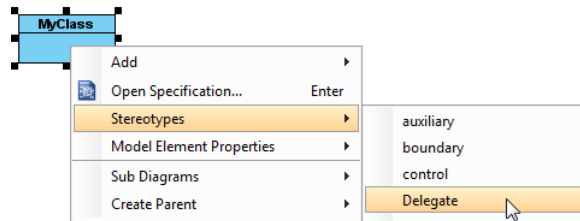
Y verás cómo las generalizaciones que habíamos seleccionado se habrán agrupado. Ajusta el conector para que quede bien:



Vamos a hacer lo mismo para las otras:



16. Si hemos programado en C# o en Visual Basic, se puede definir un método de delegación para clases. Para ello, botón derecho en la clase > Stereotypes > Delegate:



17. Visual Paradigm nos ofrece hacer conversiones automáticas entre diagramas y código Java. Para saber más sobre ello consulta la página oficial del programa, donde viene hasta con un código de ejemplo descargable con el que probar el funcionamiento:

- Conversión directa (de diagrama a código): http://www.visual-paradigm.com/support/documents/vpuserguide/276/381/7486_generateorup.html
- Conversión indirecta (de código a diagrama) http://www.visual-paradigm.com/support/documents/vpuserguide/276/381/7530_generateorup.html
- Para otro tipo de conversiones (otros lenguajes), consultar el panel izquierdo de esta página: http://www.visual-paradigm.com/support/documents/vpuserguide/276/330/27971_instantgener.html