

Tutorial Enterprise Architect 7.5 – Versión en español



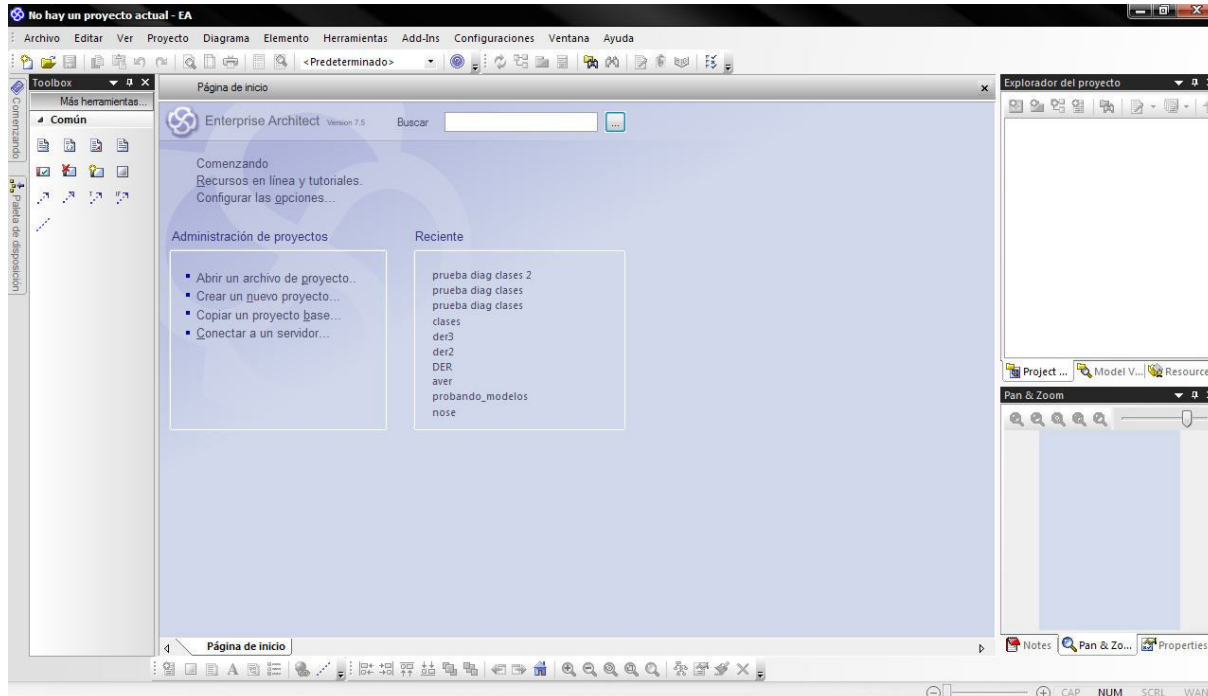
Índice

Aspectos básicos – Vistazo general	3
Diagrama de Clases	4
<i>Atributos</i>	6
<i>Métodos</i>	10
<i>Relaciones</i>	11
Configuración de Opciones de Diagramas	14
Clases de control	14
Configuración de página	16
Generación de código	17

Aspectos básicos

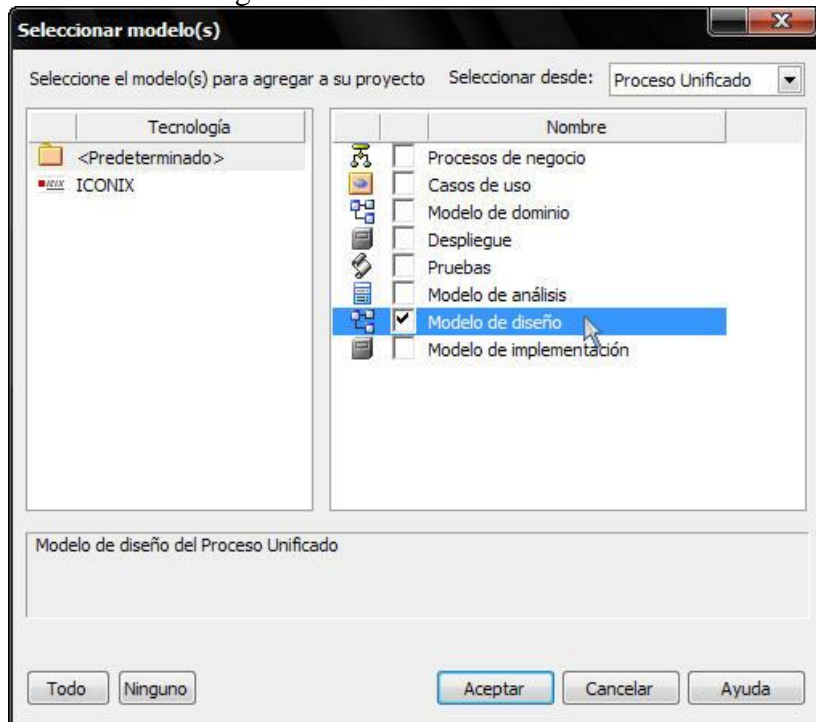
Pantalla inicial

Esta es la pantalla que se nos abre al iniciar EA por primera vez:



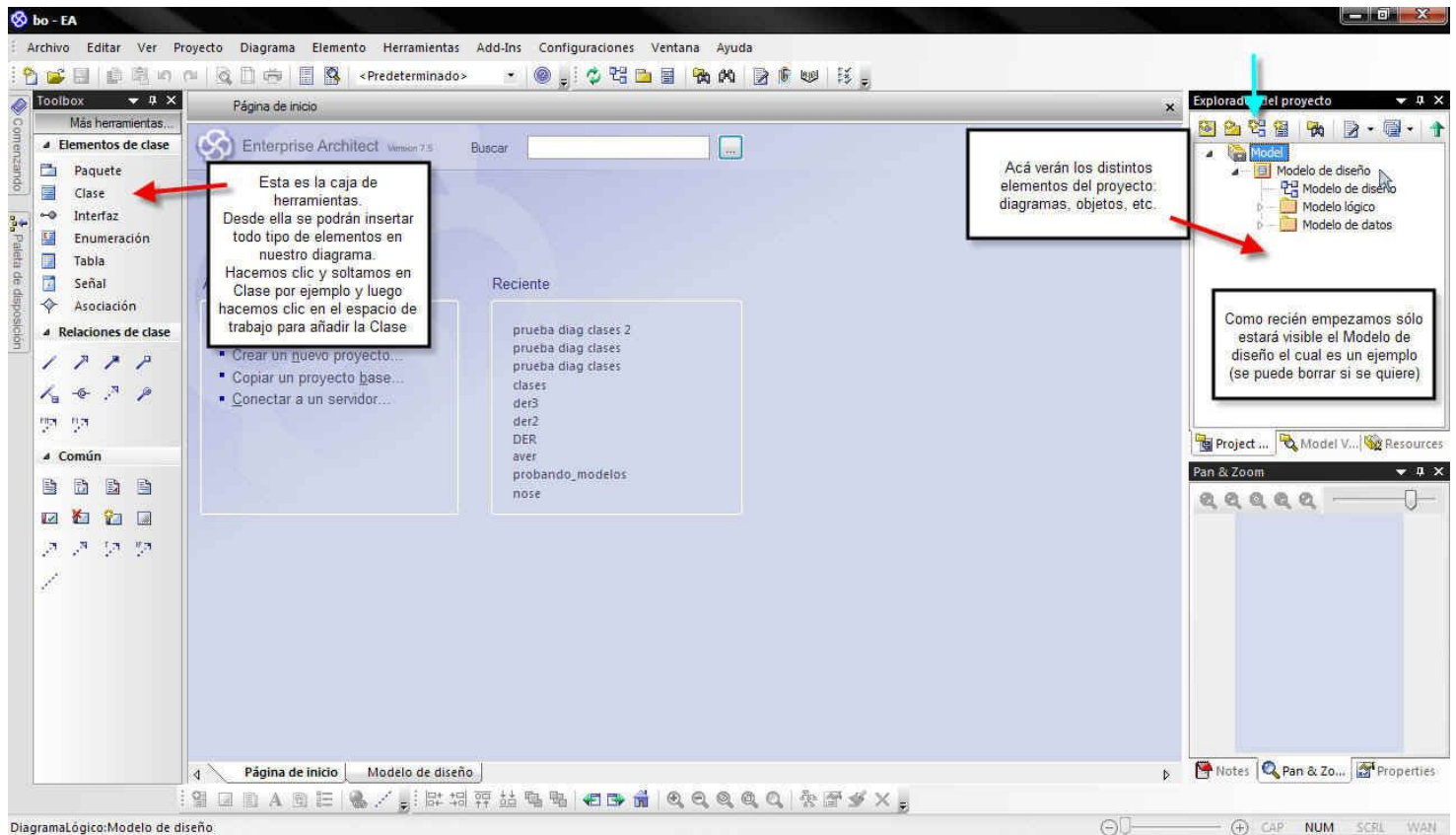
Lo primero que haremos será crear un proyecto nuevo y luego veremos en detalle la interfaz. Para crear un nuevo proyecto vamos al primer ícono, el de nuevo proyecto.

Se nos abrirá la siguiente ventana:



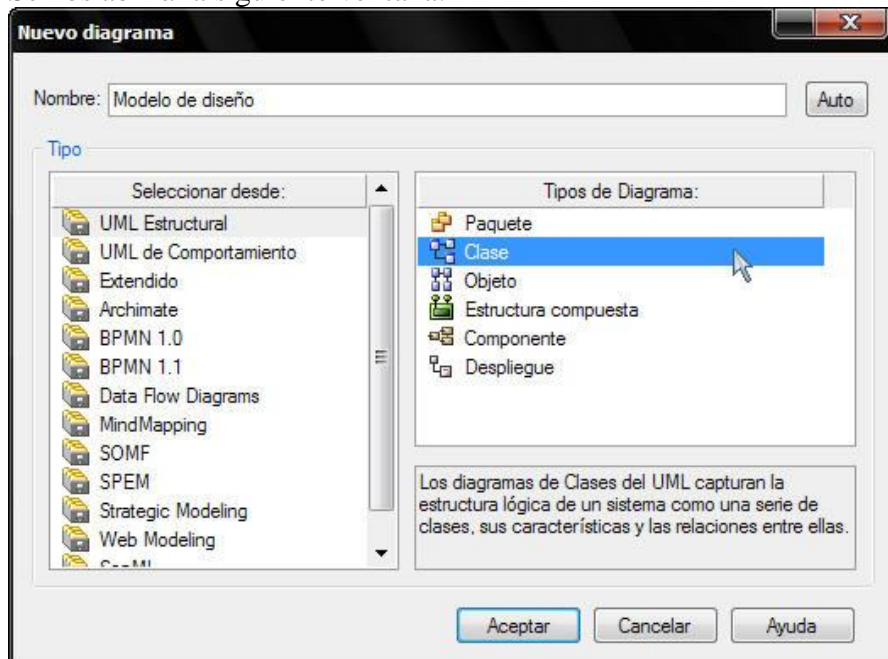
Acá elegimos
Modelo de diseño como
vemos en la imagen

Ahora EA se verá de la siguiente manera:



Ahora crearemos un nuevo diagrama, en este caso un **diagrama de Clase**. Para ello seleccionamos con el Mouse donde dice Modelo de diseño como se ve en la imagen anterior y pulsamos en el botón de nuevo diagrama (indicado con flecha celeste)

Se nos abrirá la siguiente ventana:



Seleccionamos la categoría UML Estructural y luego Clase en Tipo de diagrama

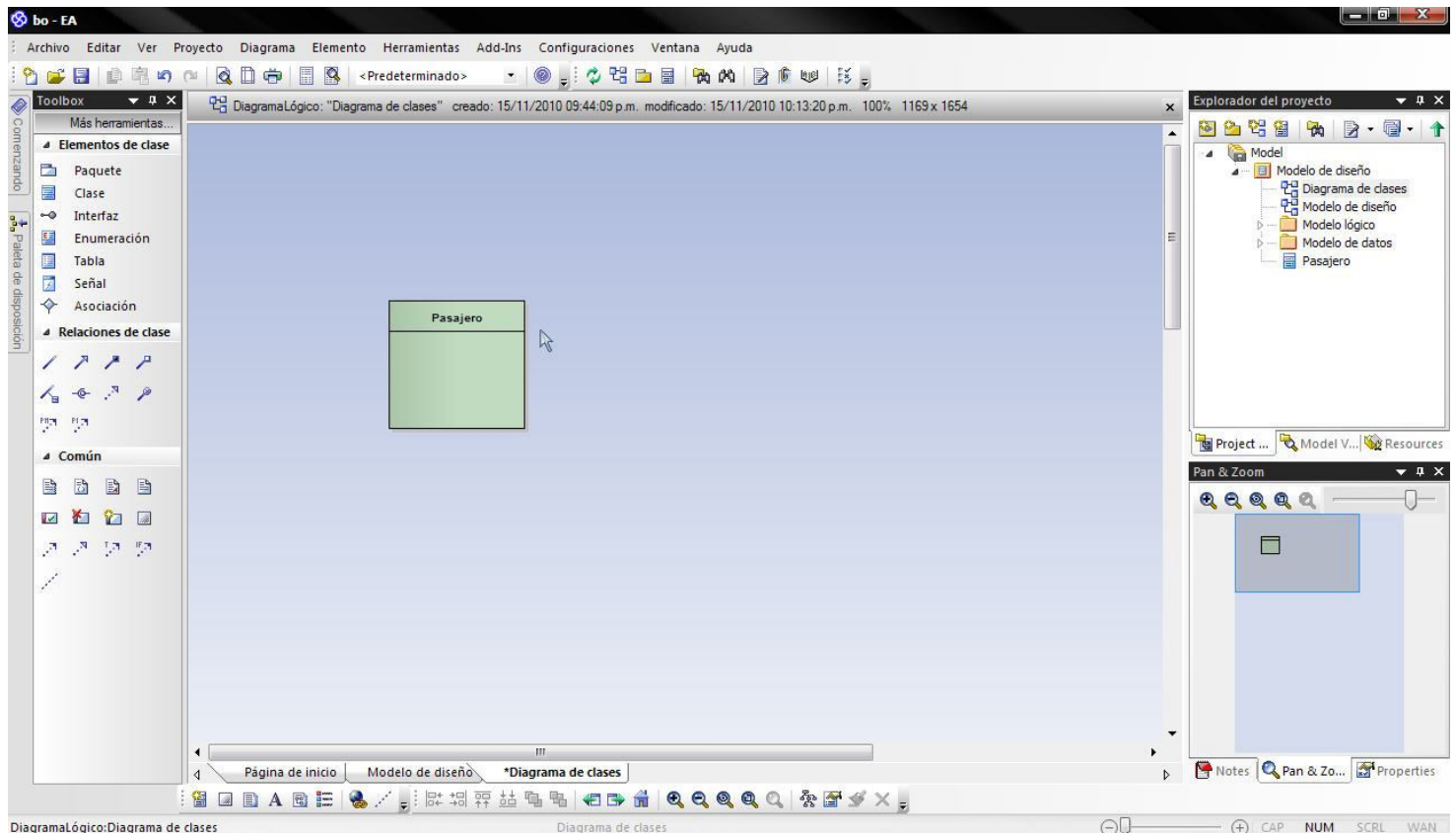
En Explorador de proyecto (arriba a la derecha) se habrá agregado un nuevo elemento “Modelo de diseño” que les recomiendo renombrar a “Diagrama de clases” pulsando F2 en el mismo.

Ahora ya podremos crear nuestro diagrama añadiendo los elementos necesarios desde la caja de herramientas.

Si añadimos una clase por ejemplo, después de hacer clic en el escenario nos aparecerá esto:

Lo único interesante aquí es el nombre de la clase, el Lenguaje en que estamos programando, por defecto Java (útil más adelante para la generación de código) y el Estereotipo (el cual dejaremos vacío para clases de entidad y lo cambiaremos si por ejemplo necesitamos crear clases de control) El estereotipo para clases de control se denomina justamente, control.

Así se ve nuestra clase después de creada:



Si hacemos doble clic en la clase, NO veremos la pantalla de definición de atributos sino que iremos a la pantalla anterior, donde definimos el nombre y demás propiedades de la clase.

Para definir los atributos y operaciones de la clase la forma más rápida es apretar F9 y F10 respectivamente (atributos, comportamiento), también lo podremos hacer a través del extenso menú contextual que se abre al pulsar con el botón secundario del mouse en la clase.

Al apretar F9 para definir **atributos**, saldrá esta pantalla:

Pasajero Atributos

General | Detalle | Restricciones | Valores etiquetados

Nombre:

Tipo: ... ☐ Derivado ☐ Propiedad

Alcance: ☐ Restricción ☐ Estático

Estereotipo:

Contenedor:

Alias:

Inicial:

Notas:

B I U A $\frac{1}{3}$ $\frac{1}{5}$ x^2 x_2

Atributos

Nombre	Tipo	Valor inicial
--------	------	---------------

Aquí definiremos los distintos atributos de nuestra clase.

Nota importante: *luego de crear un atributo y pulsar Enter es necesario apretar el botón Nuevo (o el atajo de teclado Alt+u) y luego seguir definiendo los otros atributos ya que de lo contrario estaremos sobrescribiendo siempre el mismo atributo!!*

En la lista desplegable de tipos, el tipo String no figura por lo que es necesario escribirlo manualmente.

Después de agregar algunos atributos:

Pasajero Atributos: nombre

General | Detalle | Restricciones | Valores etiquetados

Nombre:

Tipo: String ☐ Derivado ☒ Propiedad

Alcance: Private ☐ Restricción ☐ Estático

Esteriotipo:

Contenedor: Not Specified

Alias:

Inicial:

Notas:

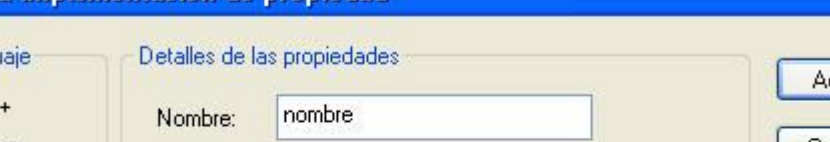
B
I
U

x^2
 x_2

Atributos

Nombre	Tipo	Valor inicial
apellido	String	
edad	int	
nombre	String	

Si seleccionamos un atributo y hacemos clic en Propiedad estaremos creando los getters y setters para ese atributo!



Crear la implementación de propiedad

Lenguaje

- ☐ C++
- ☒ Java
- ☐ Visual Basic
- ☐ C#
- ☐ Delphi
- ☐ VB Net
- ☐ PHP

Detalles de las propiedades

Nombre:

Getter:

Setter:

Alcance de Get: ☒ Lectura

Alcance de Set: ☒ Escritura

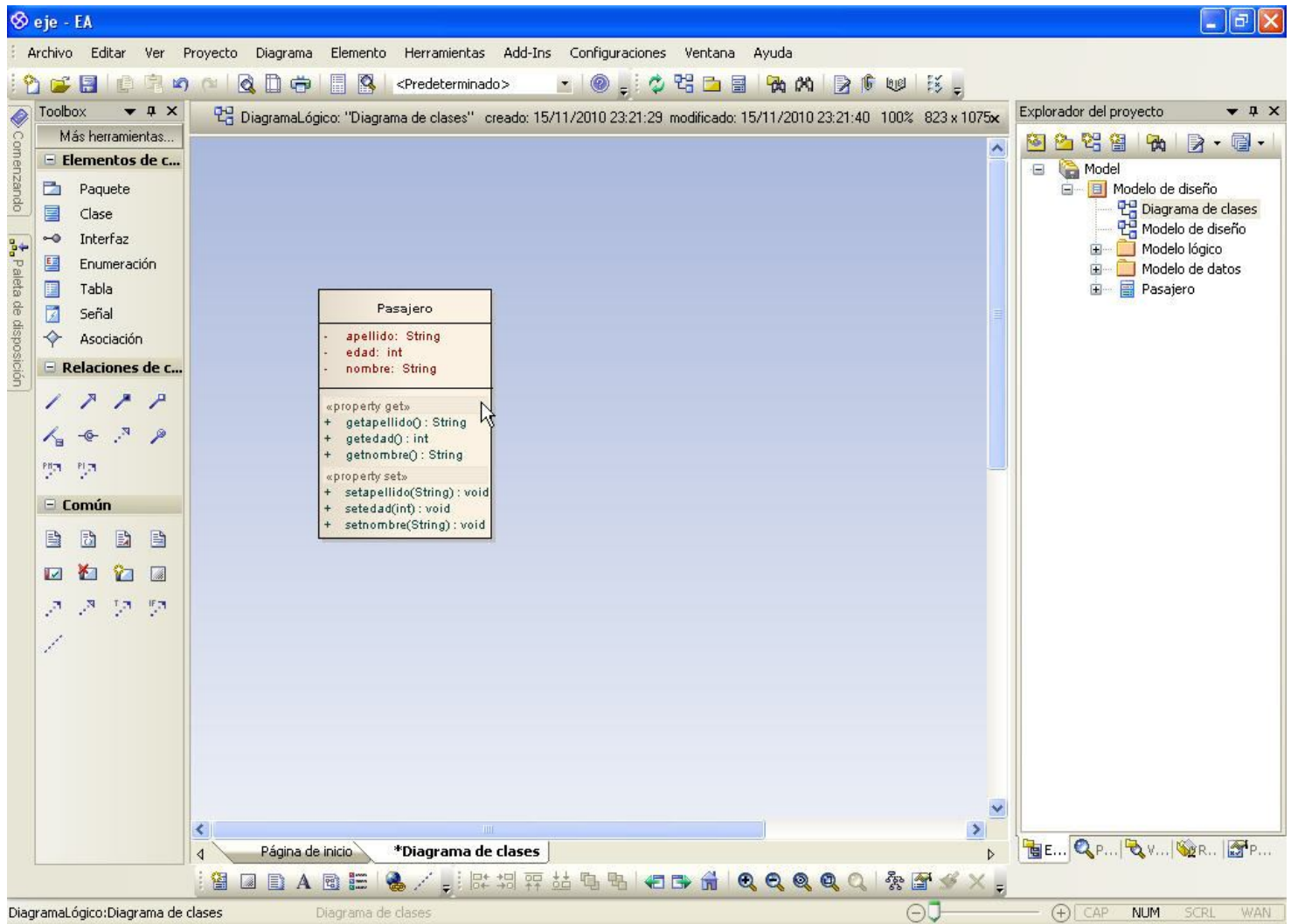
☐ Abstracto

Aceptar

Cancelar

Ayuda

Así quedará la clase con sus atributos:



Vemos que arriba de los getters aparece una leyenda <<property get>> y arriba de los setters <<property set>> indicando que son métodos gets y sets.

Si queremos que no aparezcan esas leyendas entonces iremos a Diagrama->Propiedades y allí en la pestaña Características desactivaremos la casilla de verificación “Mostrar los estereotipos”

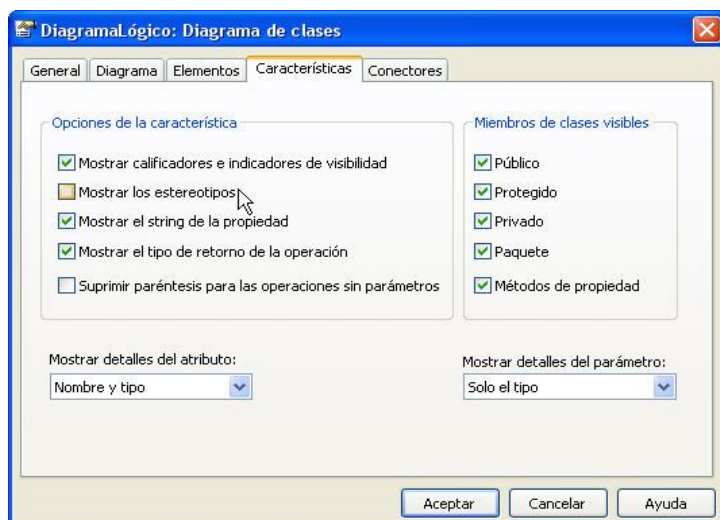
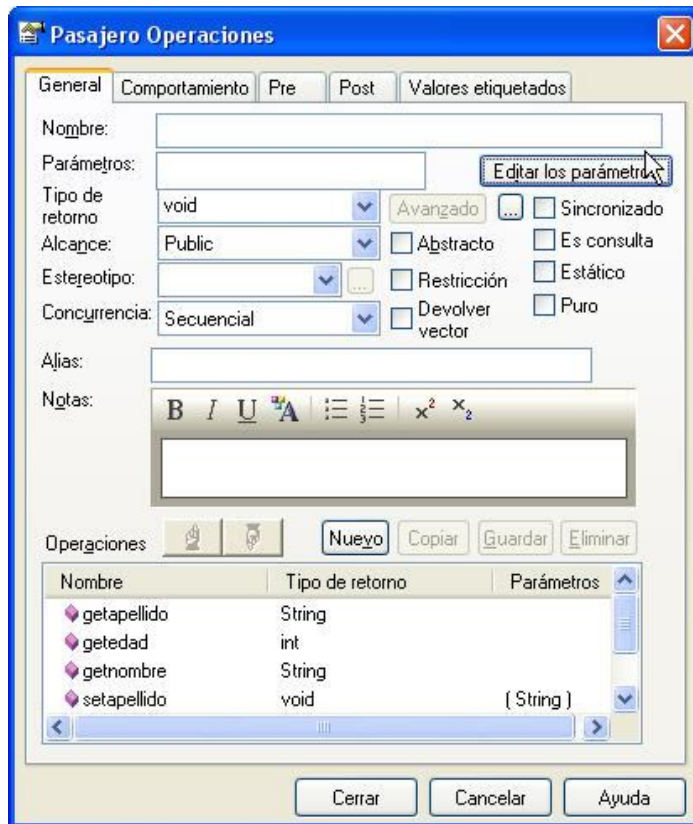


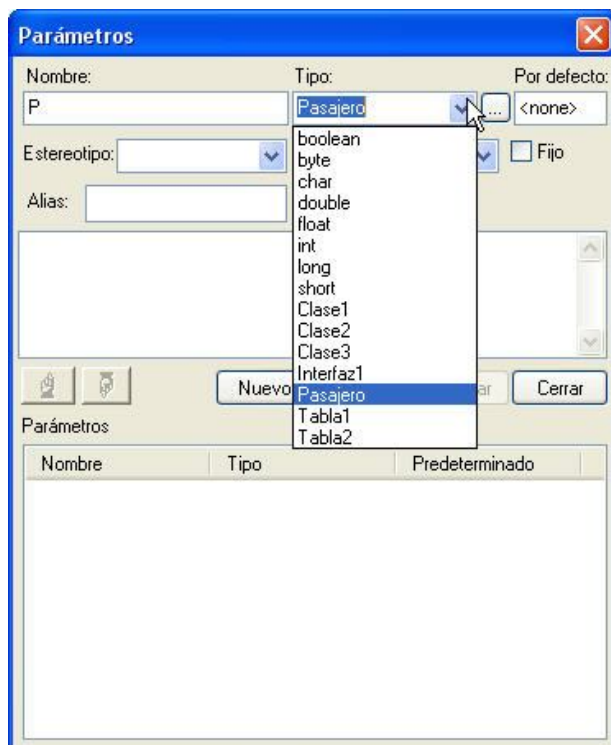
Diagrama de clases – Métodos

Para definir los métodos haremos clic en la clase y apretaremos F10:



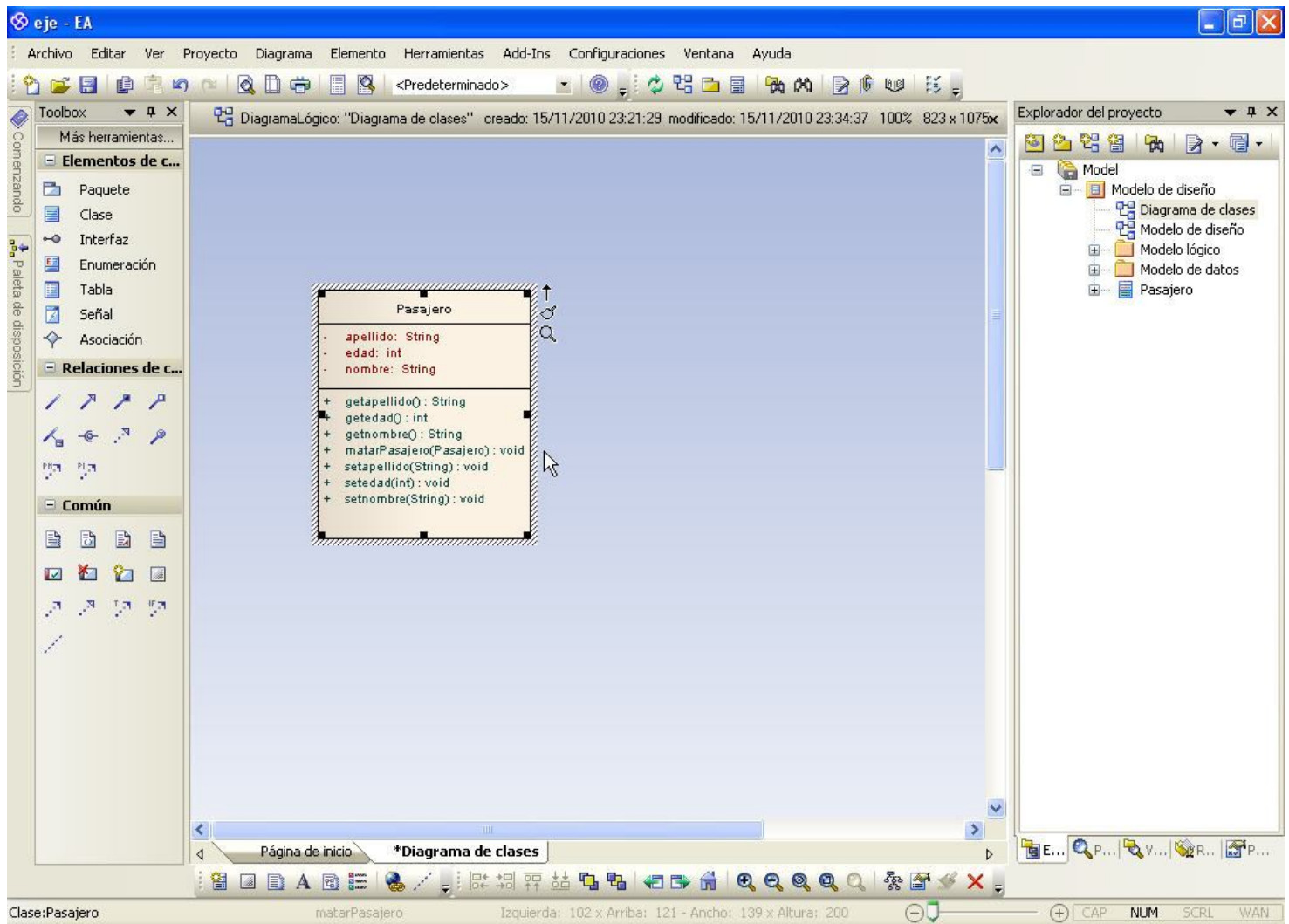
Lo único que nos interesa es el nombre, alcance, tipo de retorno y parámetros.

Al hacer clic en **Editar los parámetros** podremos definir el tipo del parámetro.



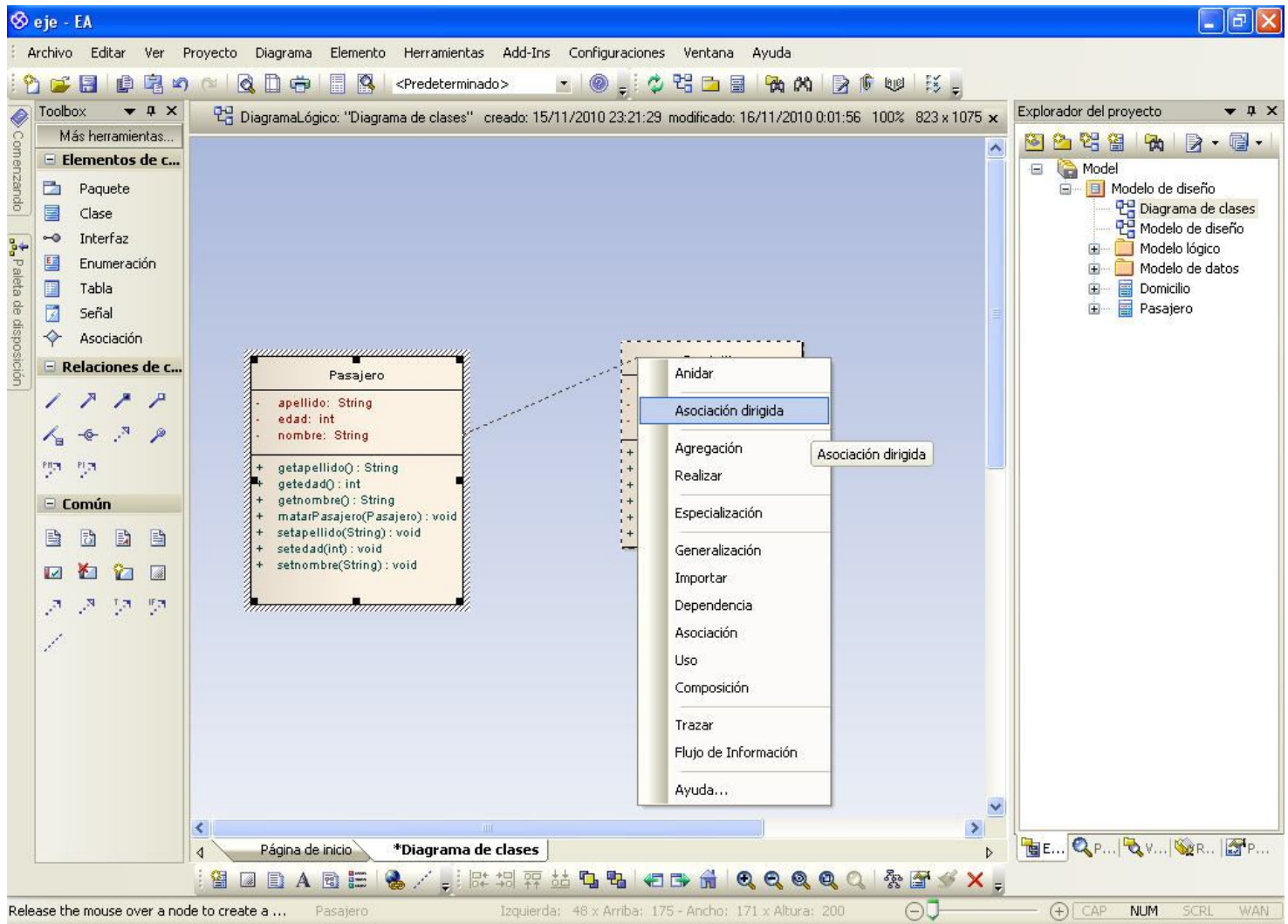
En Tipo aparecerán tanto los tipos primitivos como las clases que tengamos definidas en el diagrama.

Clase con atributos y métodos definidos



Relaciones entre clases

Para relacionar dos clases entre sí lo haremos pulsando sobre la flecha que aparece al hacer clic en la clase origen y arrastraremos la misma hasta la clase destino:



Ahí elegimos el tipo de relación que queremos realizar entre las clases.

Cardinalidad de las relaciones

Para cambiar la cardinalidad y nombre de la relación haremos doble clic sobre la misma:

The 'Asociación Propiedades' dialog box is shown with the 'General' tab selected. It contains the following fields and options:

- Origen:** Pasajero
- Destino:** Domicilio
- Nombre:** (empty text box)
- Alias:** (empty text box)
- Dirección:** Origen -> Destino (dropdown menu)
- Estilo:** Personalizado (dropdown menu)
- Estereotipo:** (empty dropdown menu with a button to open the list)
- Notas:** A text area with a rich text editor toolbar (Bold, Italic, Underline, Bulleted List, Numbered List, Indent, Outdent, Link, Unlink, Undo, Redo).
- Buttons:** Aceptar, Cancelar, Ayuda.

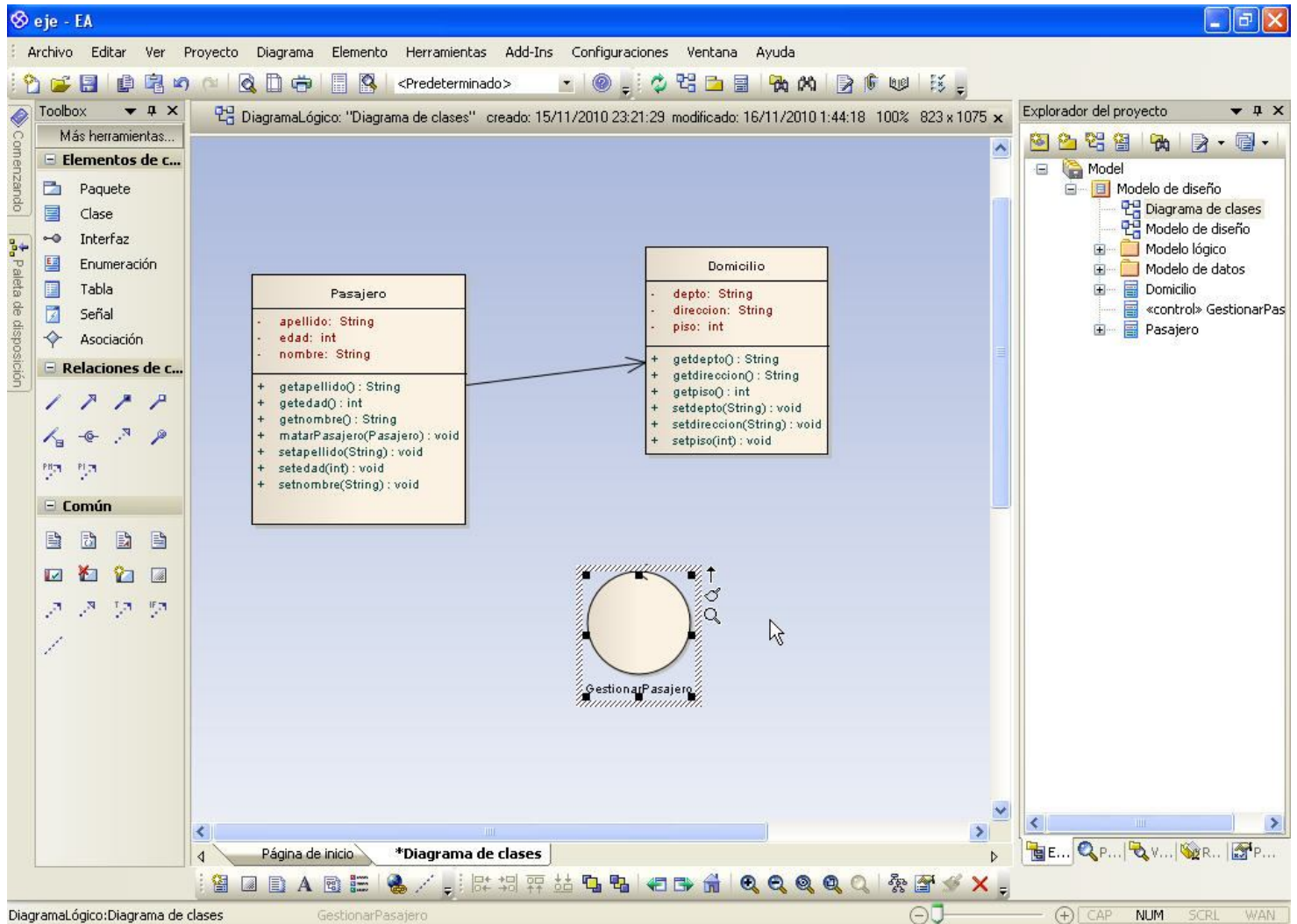
En las pestañas Rol de origen y destino podremos definir la multiplicidad de la relación:

The 'Asociación Propiedades' dialog box is shown with the 'Rol de origen' tab selected. It contains the following fields and options:

- Pasajero:** (dropdown menu showing 'Pasajero')
- Alias:** (empty text box)
- Notas del rol:** (empty text area)
- Derivado:** ☐
- Unión derivada:** ☐
- Apropiado:** ☐
- Contenedor:** (empty dropdown menu)
- Acceso:** Public (dropdown menu)
- Agregación:** ninguno (dropdown menu)
- Alcance del destino:** instancia (dropdown menu)
- Navegabilidad:** No navegable (dropdown menu)
- Modificable:** ninguno (dropdown menu)
- Multiplicidad:** A dropdown menu is open, showing options: *, 0, 0..*, 0..1, R 1, 1.., C 1..*. The '*' option is currently selected.
- Estereotipo:** (empty dropdown menu with a button to open the list)
- Tipo de miembro:** (empty text box)
- Buttons:** Aceptar, Cancelar, Ayuda.

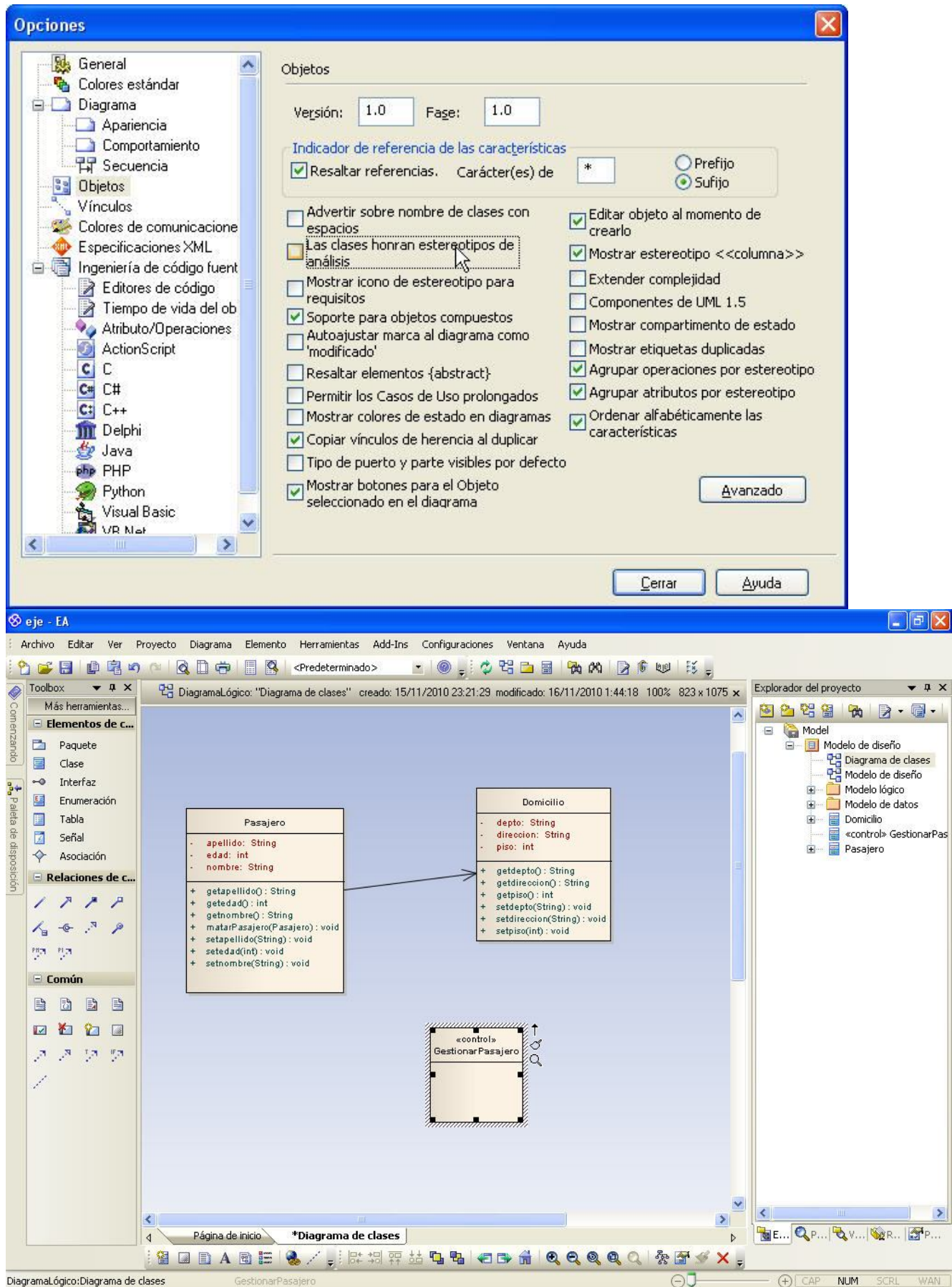
Configurando opciones del diagrama

Al crear una clase de control vemos que EA la representa de la siguiente manera:

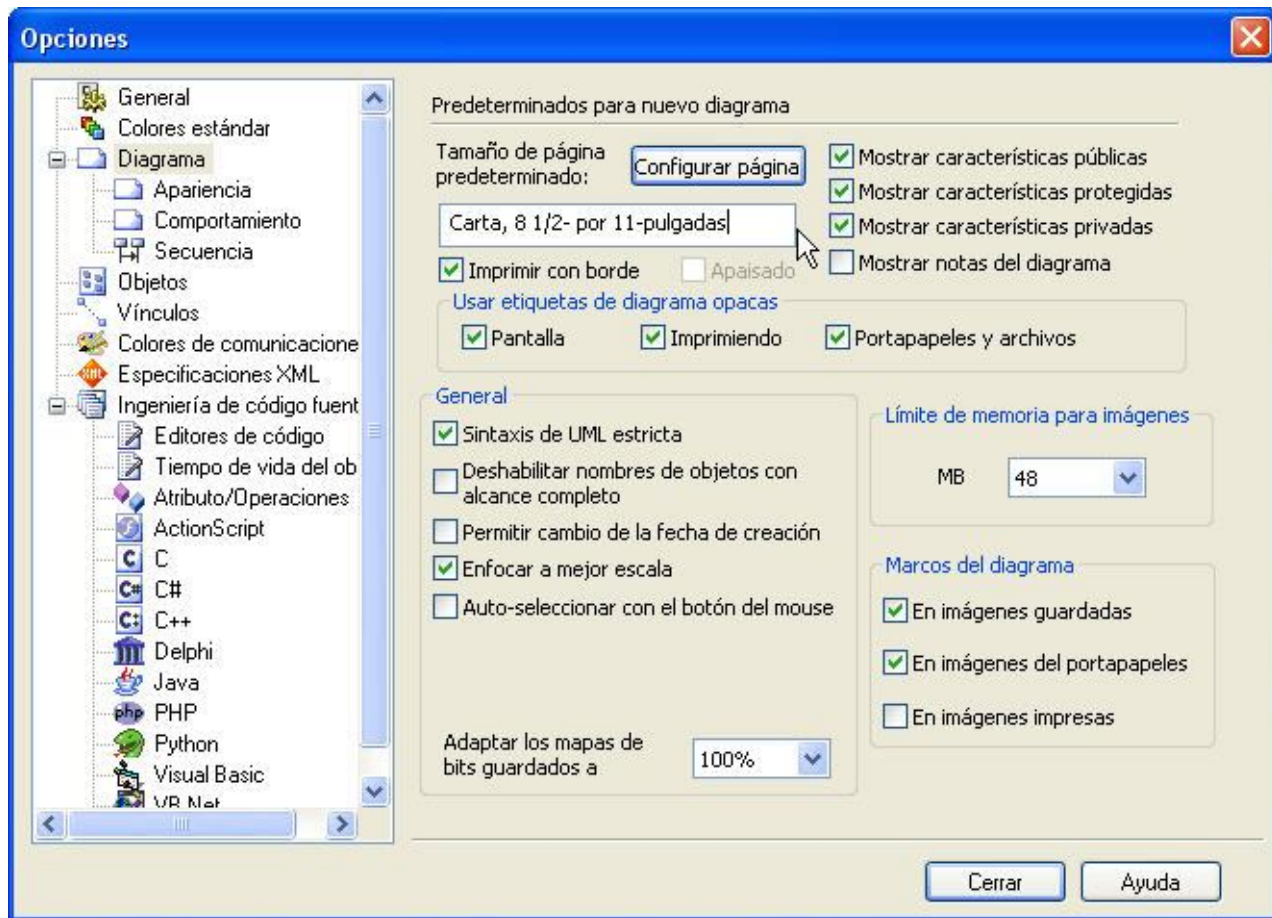


Es decir, como una clase de control del modelo de análisis.

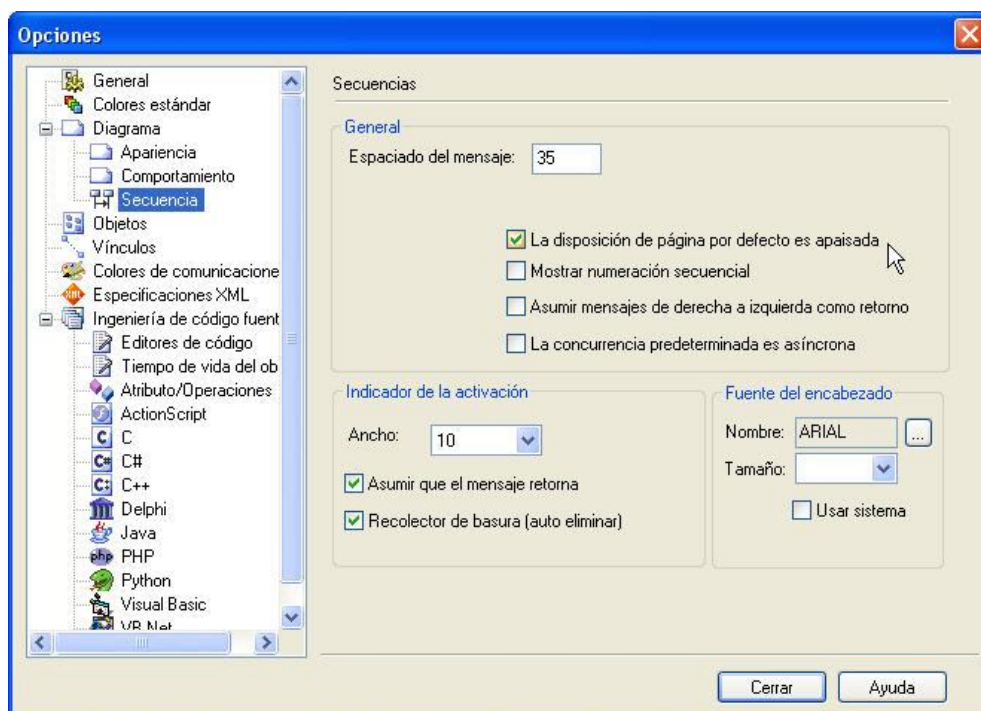
Si nosotros queremos que se vea como una clase de diseño tenemos que ir a Herramientas->Opciones->Objetos y destildar la casilla que dice “Las clases honran estereotipos de análisis”:



Configurando tamaño predeterminado de página



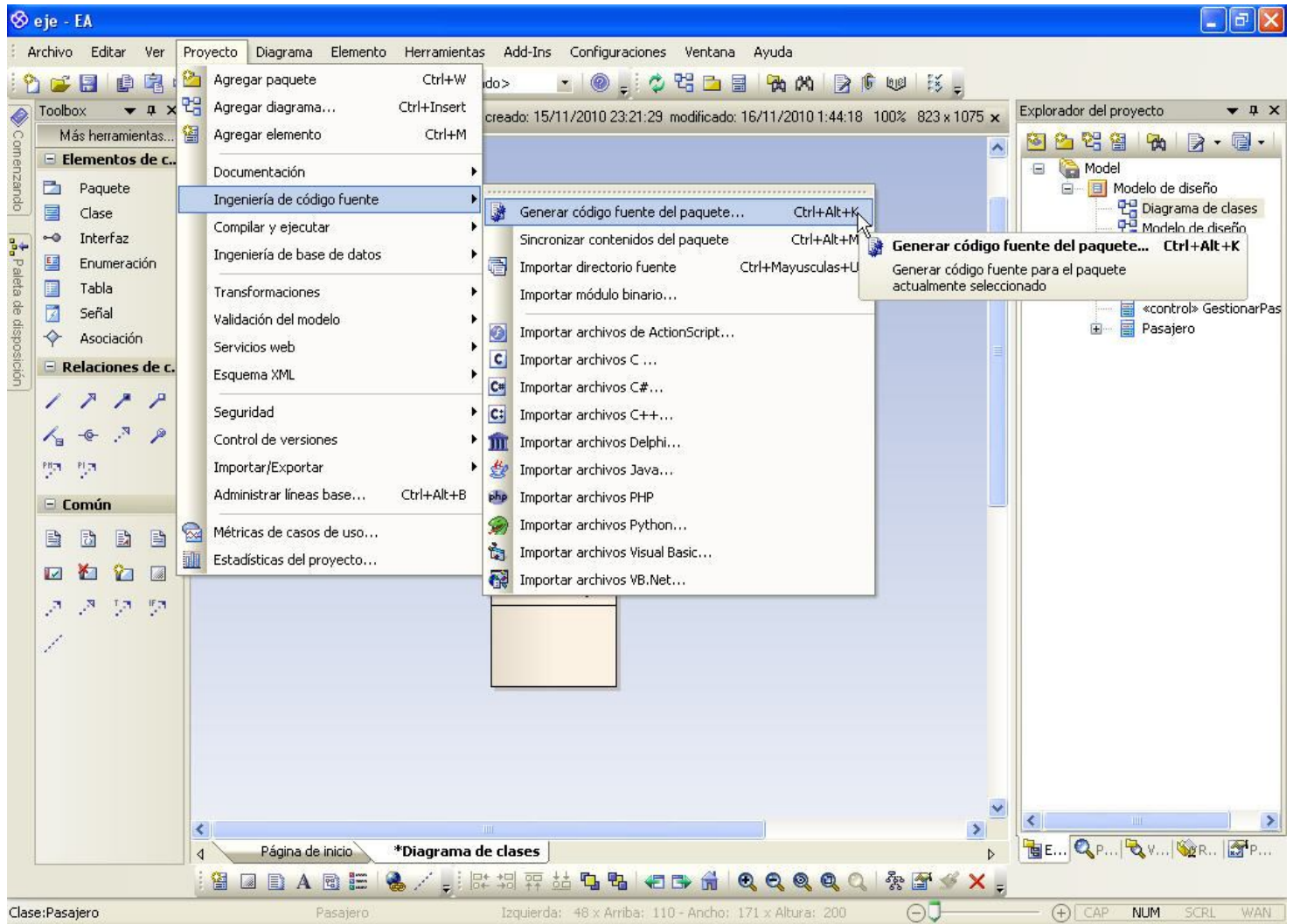
Configurando orientación Diagrama de Secuencia



Generación de código

Si bien al principio el Architect es complicado de manejar, una de las razones por las que lo estoy utilizando es la de generación de código que ofrece el programa.

Se puede generar el código (Java, C, C++ entre otros) de una clase en particular o de todo el diagrama. Para ellos debemos dirigirnos a:



Saldrá la siguiente ventana:



Aquí pondremos la opción Auto generar archivos y le diremos donde quiere que guarde los archivos generados.

Luego le daremos a generar y ya tenemos listo el cód. fuente a partir del diagrama ☺

Ejemplo (clase Pasajero):

```
/**
 * @author Administrador
 * @version 1.0
 * @created 16-nov-2010 2:05:54
 */
public class Pasajero {

    private String apellido;
    private int edad;
    private String nombre;
    public Domicilio m_Domicilio;

    public Pasajero(){

    }

    public void finalize() throws Throwable {

    }

    public String getapellido(){
        return apellido;
    }

    public int getedad(){
```

```
        return edad;
    }

    public String getnombre(){
        return nombre;
    }

    /**
     *
     * @param P
     */
    public void matarPasajero(Pasajero P){

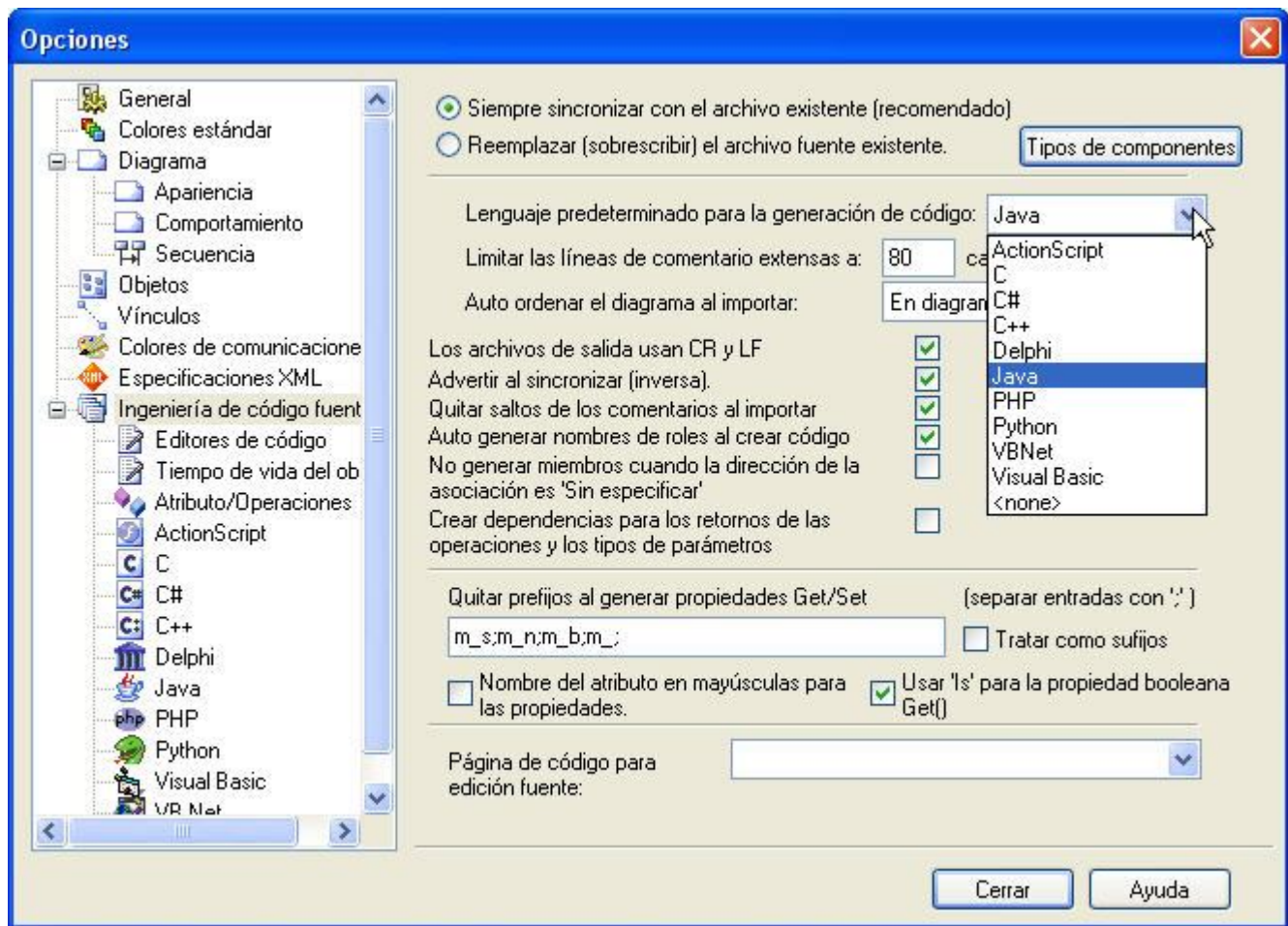
    }

    /**
     *
     * @param newVal
     */
    public void setapellido(String newVal){
        apellido = newVal;
    }

    /**
     *
     * @param newVal
     */
    public void setedad(int newVal){
        edad = newVal;
    }

    /**
     *
     * @param newVal
     */
    public void setnombre(String newVal){
        nombre = newVal;
    }
}
```

Por último para cambiar el lenguaje de generación de código (por si necesitamos el código en un lenguaje distinto a Java) nos dirigimos a:



Hasta acá llega el tuto básico de Enterprise Architect 7.5, espero que por lo menos los ayude a empezar a utilizar el programa y les sea de utilidad.

Hasta luego!!!

MARCOS P ---- www.facebook.com/nmcomp