

Hoja 2 de Ejercicios. Soluciones.

Orientación a Objetos

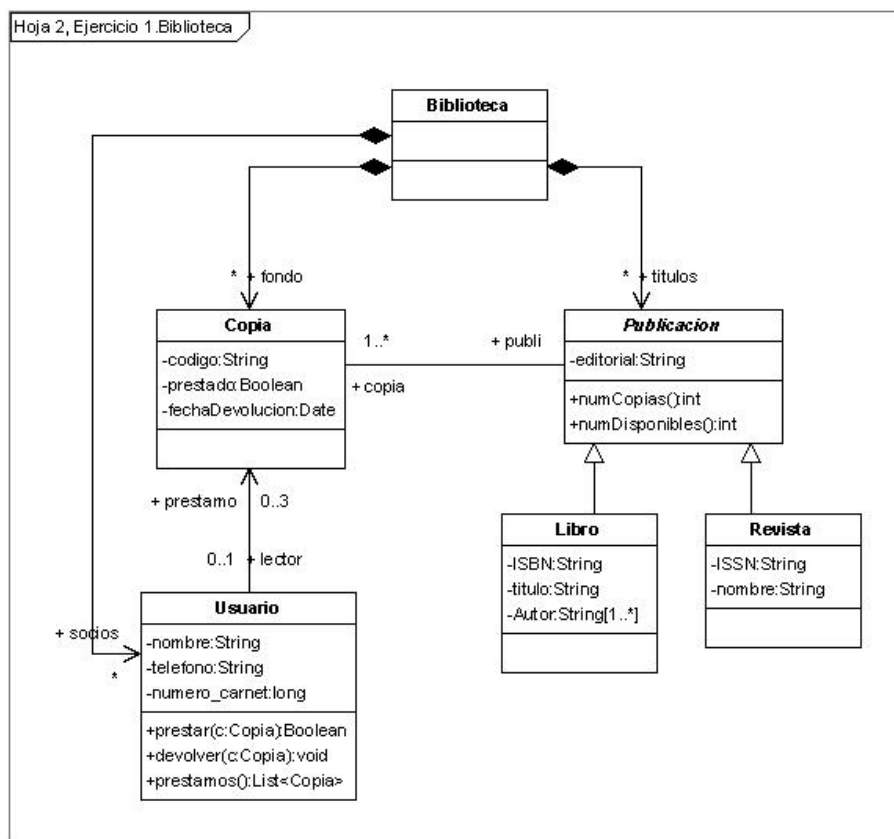
NOTA: En la mayoría de los casos, estas soluciones son esquemáticas, que el estudiante debe completar de acuerdo al enunciado

Inicio: Semana del 13 de Febrero.

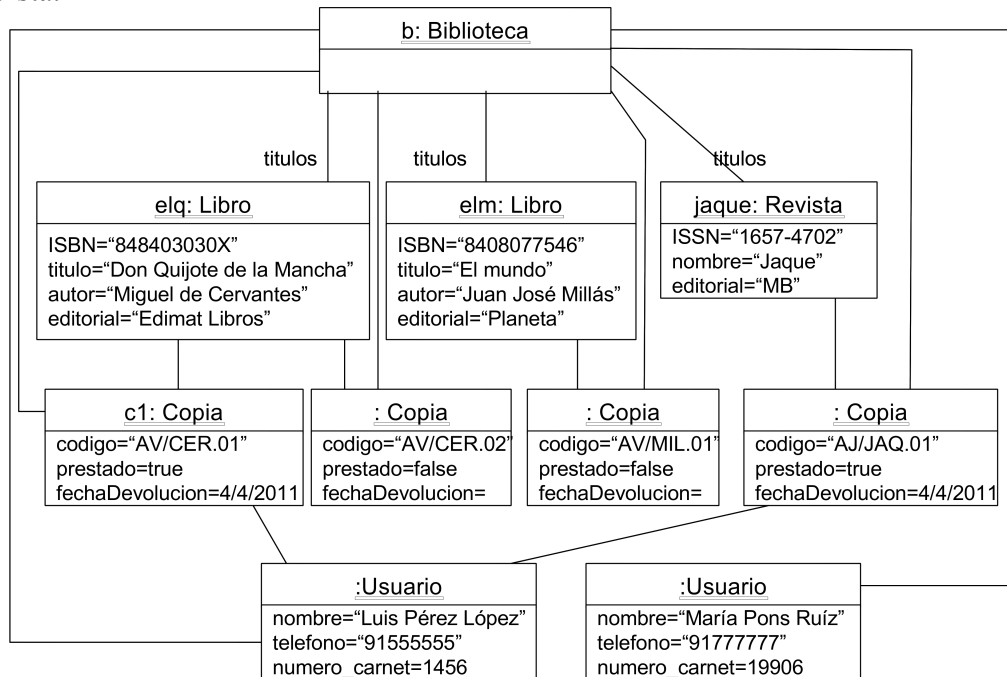
Duración: 2 semanas.

- 1) Construye el diagrama de clases de la siguiente aplicación para una biblioteca. La biblioteca cuenta con dos tipos de publicaciones: revistas y libros. Ambos son editados por alguna editorial. Mientras que los libros tienen un ISBN, un título y (uno o más) autores, las revistas tienen un ISSN y un nombre. La biblioteca tiene una o más copias de las publicaciones que formar su fondo bibliográfico. Las copias están clasificadas de acuerdo a un código. Los usuarios de la biblioteca pueden coger un máximo de 3 publicaciones a la vez.

Añade métodos a las clases adecuadas para: (i) que los usuarios puedan devolver o coger publicaciones, así como que se pueda acceder a las publicaciones que tienen prestadas, (ii) que dada una publicación, sepamos el número de copias que dispone la biblioteca, y el número de ellas que no están prestadas.

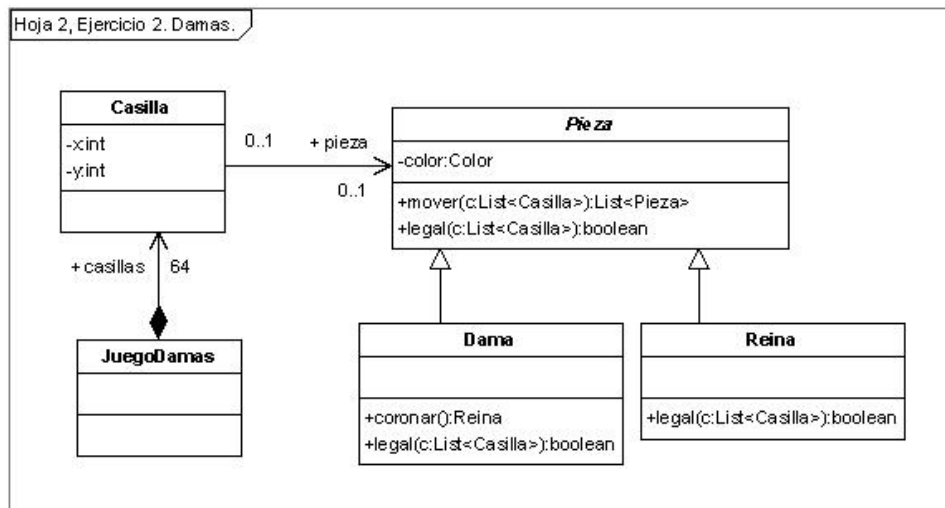


- 2) Basándote en el diagrama de clases del ejercicio 1, construye un diagrama de objetos que contenga al menos 2 usuarios, 2 libros, 1 revista. Uno de los usuarios tiene prestado un libro y una revista.



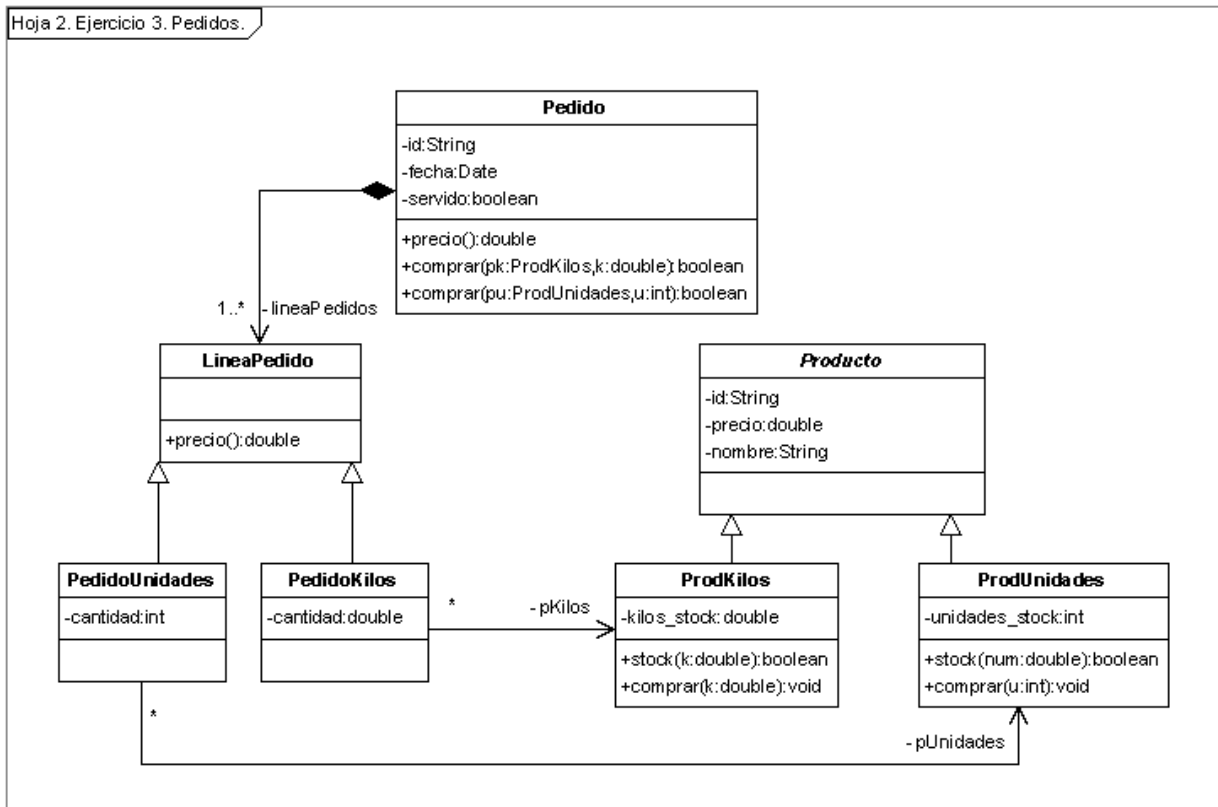
3) Construye el diagrama de clases que modele el juego de las damas. Este es un juego para dos jugadores que se desarrolla en un tablero cuadrado de 8x8 casillas. Las piezas tienen un color (negras o rojas) y pueden ser de dos tipos: damas o reinas. Mientras que las damas se mueven en diagonal una casilla, las reinas pueden moverse varias casillas a la vez (siempre en diagonal, y también hacia atrás). Al llegar al extremo del tablero, una dama se promociona en reina. Un movimiento también puede capturar una pieza al adversario, y es posible encadenar varias capturas en un sólo movimiento de una pieza.

Añade métodos a las clases adecuadas para: (i) mover una pieza, (ii) comprobar si un movimiento es legal, (iii) promocionar una dama.



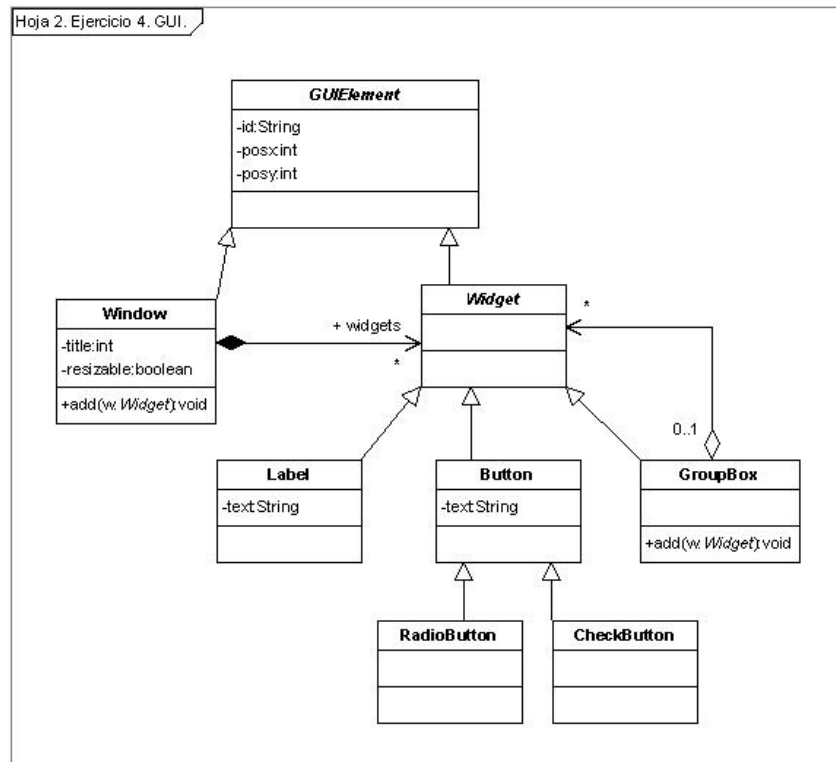
4) Construye el diagrama de clases para una aplicación de gestión de pedidos. Hay dos tipos de productos, los que se venden por unidades, y los que se venden por kilos. En cualquiera de los dos casos, los productos están descritos por un identificador, un nombre y un precio (por unidad o por kilo). Un pedido está formado por líneas de pedido, que contienen las unidades o kilos que se han pedido de un producto determinado. Los pedidos están descritos por un identificador, la fecha en que se efectuó el pedido y un atributo que indica si el pedido se ha servido o no.

Añade métodos a las clases adecuadas para: (i) calcular el precio de un pedido, (ii) añadir a un pedido la compra de ciertas unidades o kilos de un producto determinado. Este método debe comprobar si hay stock del producto que se quiere añadir, devolviendo si la operación se ha podido realizar o no.



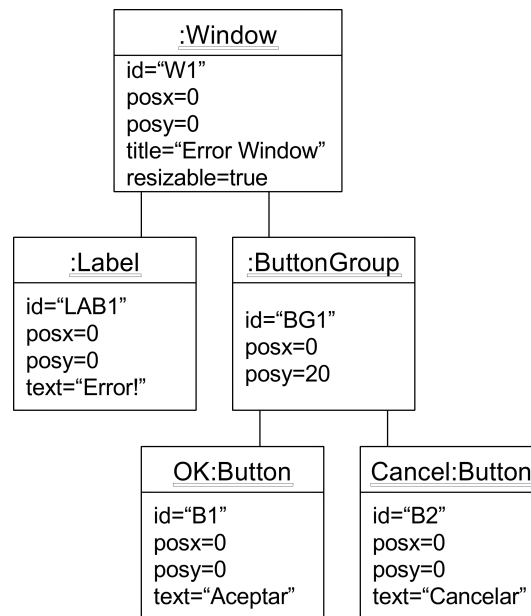
5) Construye el diagrama de clases para una aplicación que permita la definición de interfaces de usuario gráficas (GUI) sencillas. Todos los elementos de una interfaz contienen un identificador y una posición (coordenadas x e y). Las ventanas (Window) tienen un título y pueden ser o no redimensionables. Además, pueden contener cero o más elementos de tipo etiqueta (Label), botón (Button), o grupo (GroupBox). Las etiquetas y botones tienen un texto. Los botones además pueden ser normales, radio-botones (RadioButton) o de selección (CheckBox). Un GroupBox puede contener etiquetas, botones y otros GroupBoxes.

Añade métodos a las clases adecuadas para: (i) añadir un elemento a una ventana, y (ii) añadir un elemento a un GroupBox.

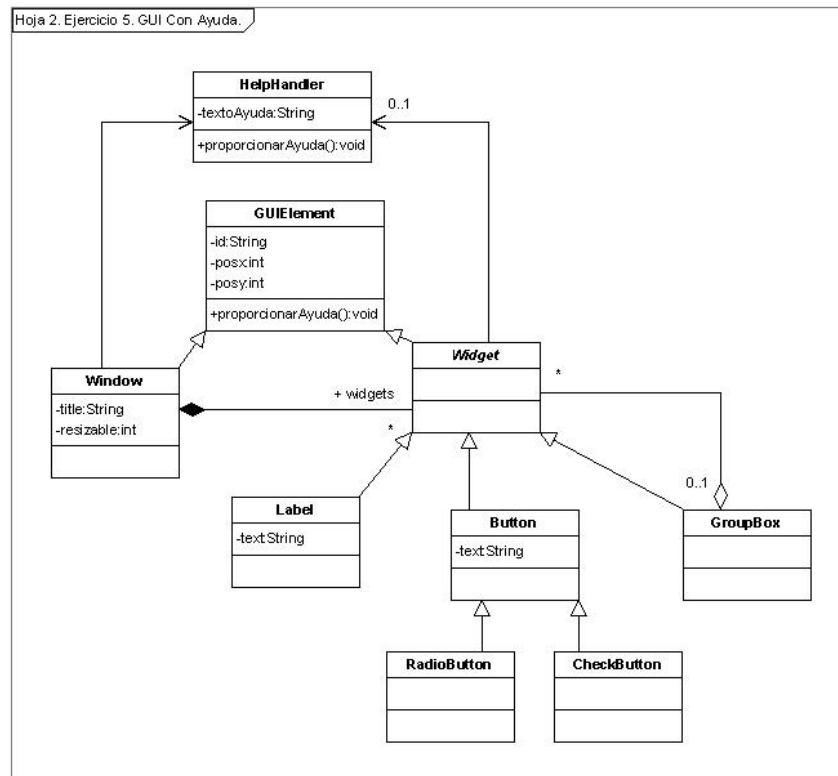


Nota: Tenemos el patrón de diseño “Composite”.

6) Basándote en el diagrama de clases del ejercicio 5, construye un diagrama de objetos que contenga una ventana con dos botones (Aceptar y Cancelar) dentro de un grupo y un mensaje.



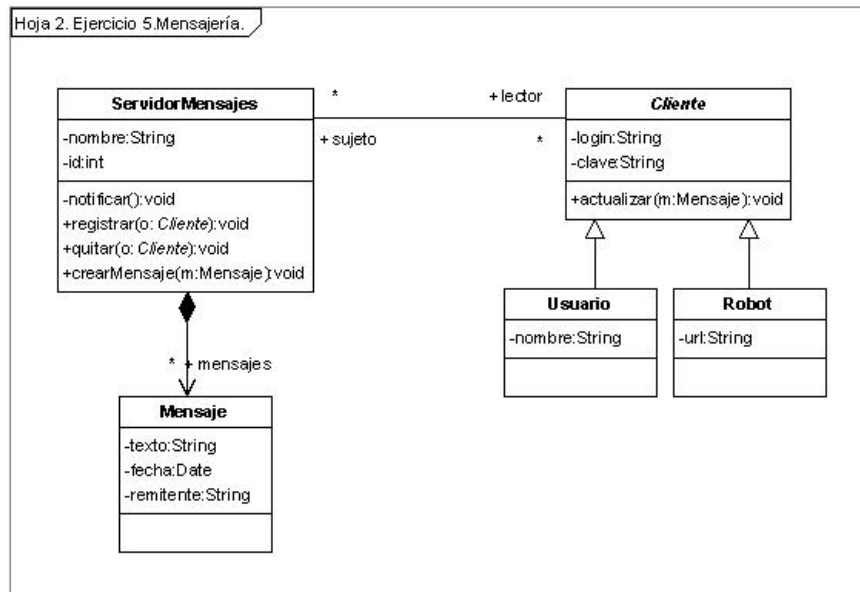
7) Modifica el diagrama de clases del ejercicio 5 para añadir un sistema de ayuda contextual, que cuando se activa, da ayuda sobre el elemento específico que tiene el foco en ese momento. Si un elemento de la interfaz no tiene asignada ninguna ayuda, entonces se debe presentar la ayuda del contenedor del elemento, si este existe. La ventana, al ser el elemento de más alto nivel, siempre tiene asignado un elemento de ayuda contextual.



Nota: Es parecido al patrón de diseño “chain of responsibility”. Hay que poder navegar por los enlaces de composición, que han de ser bidireccionales.

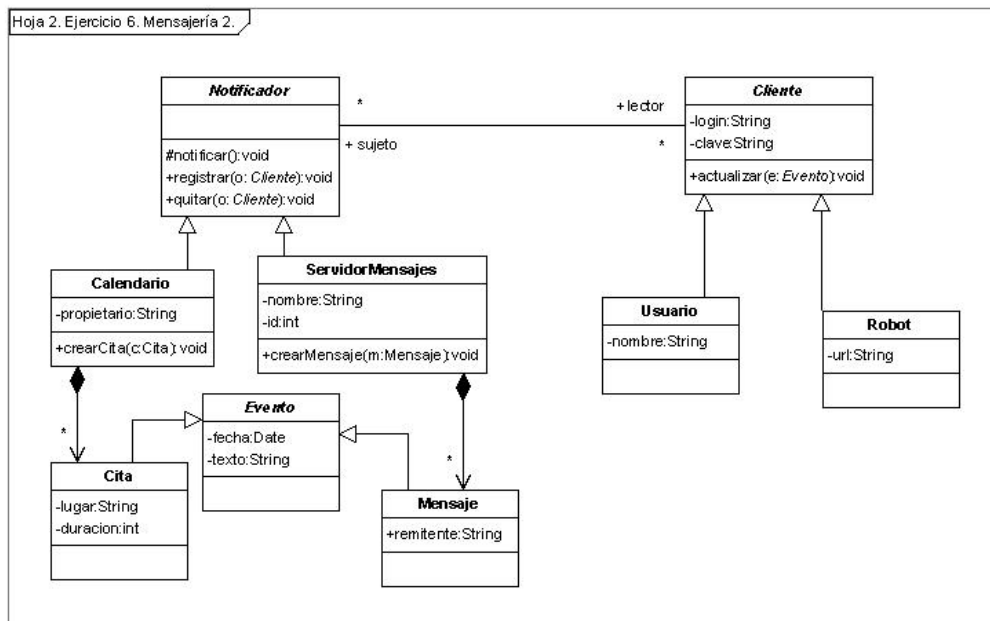
8) Construye el diagrama de clases de una aplicación de mensajería. Los mensajes están formados por un texto, un remitente y una fecha. Los mensajes se añaden a un servidor de mensajería, que está descrito por un nombre y un identificador. Distintos clientes pueden registrarse en un servidor de mensajes, para leer su contenido. Hay dos tipos de clientes: usuarios (personas) y robots (aplicaciones). Los primeros tienen un nombre y los segundos una URL donde muestran los mensajes de manera automática. Ambos tipos de cliente tienen un login y una clave. Ambos tipos de cliente tienen un login y una clave.

Añade métodos a las clases adecuadas para: (i) Que los clientes puedan registrarse y borrarse de un servidor, (ii) que se puedan crear mensajes en un servidor, y que el servidor avise a todos los clientes registrados de la llegada de un nuevo mensaje.



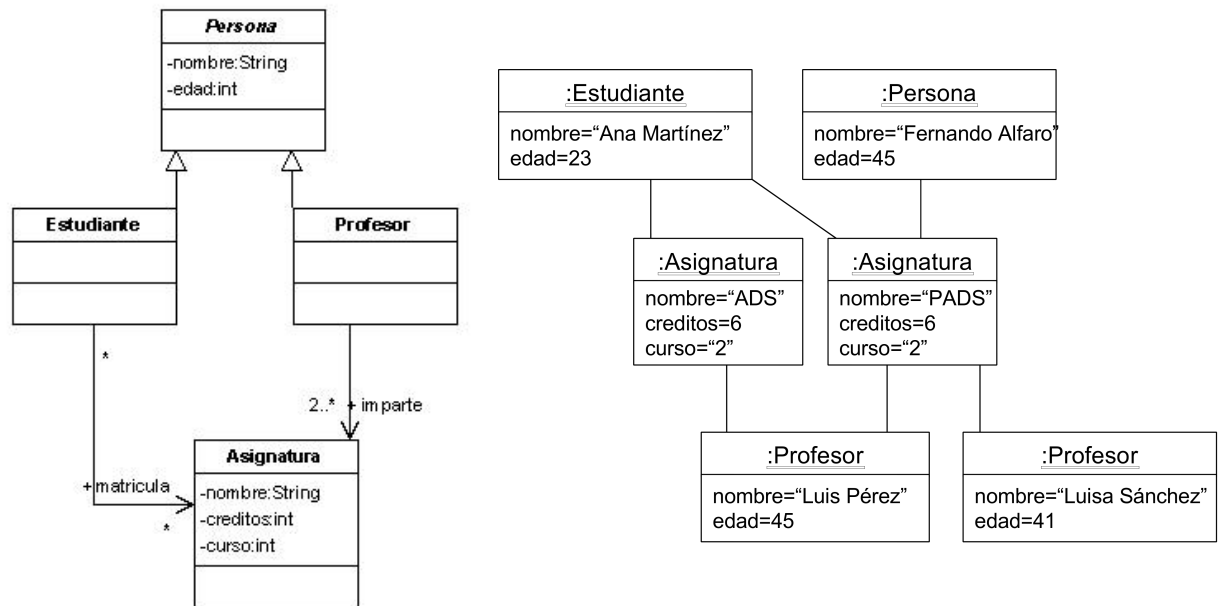
Nota: Esta solución es parecida al patrón de diseño Observer.

9) Modifica el diseño anterior para que se considere además un servicio de calendario, donde se pueden añadir citas, que están formadas por una fecha, lugar, duración y un texto descriptivo. Se desea que los clientes puedan acceder de manera uniforme tanto al servicio de calendario como de mensajes.



Nota: Es un patrón Observer.

10) Dado el siguiente diagrama de clases de la izquierda, indica los errores (si los hubiera) del diagrama de objetos de la derecha.



Persona es abstracta, y además no podemos conectarlo con asignatua.

Curso es entero.

La cardinalidad mínima de “imparte” es 2.

11) Se desea diseñar una aplicación para la edición de documentos. Los documentos pueden ser de dos tipos: de texto plano o gráficos. Para cualquier tipo de documento se ha de guardar el nombre de fichero, el idioma en que está escrito, así como el autor. Un documento gráfico está compuesto de páginas, que tienen una cabecera y un pie (que pueden ser distintos para cada página). Las páginas a su vez pueden contener imágenes, sonidos, fragmentos de texto, o bien tener incrustados documentos (de cualquiera de los dos tipos). Una imagen está descrita por el nombre de fichero, su tipo (JPEG, GIF, PNG o BMP), tamaño y posición. Un sonido está descrito por el nombre de fichero y su tipo (WAV, MP3 o WMA), y un fragmento de texto por un nombre de fichero y el texto que contiene. Un documento de texto plano no contiene páginas, sino que está compuesto directamente por fragmentos de texto.

Se pide:

- Construye el diagrama de clases UML que refleje el diseño.
- Añade los métodos necesarios a las clases correspondientes para implementar los siguientes requisitos:
 - Cuando una página se abre en el editor, ha de mostrar todos los componentes que contiene (imágenes, sonidos, fragmentos de texto y documentos).
 - Por defecto, los documentos dentro de una página se muestran como un icono, reflejando simplemente el nombre del fichero. Los documentos han de poderse desplegar, y se mostraría entonces su contenido.
- Debe ser posible justificar los fragmentos de texto individuales, así como todos los fragmentos de la misma página.

