CENTRE ESPECÍFIC
D´EDUCACIÓ A DISTÀNCIA DE
LA COMUNITAT VALENCIANA

UD 8

DIAGRAMAS DE ESTRUCTURA

PRÁCTICA 02 [NO EVALUABLE]

DIAGRAMAS DE CLASES (II)

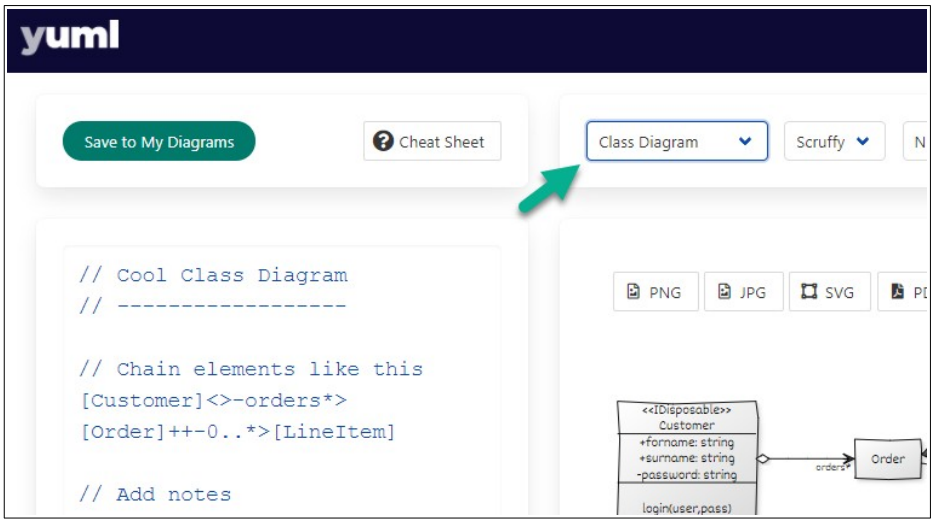
Revisado por:
Sergio Badal
Autores:
Cristina Álvarez, Fco. Javier Valero Garzón, M.ª Carmen Safont, Paco Aldarias
Fecha:
13/02/21

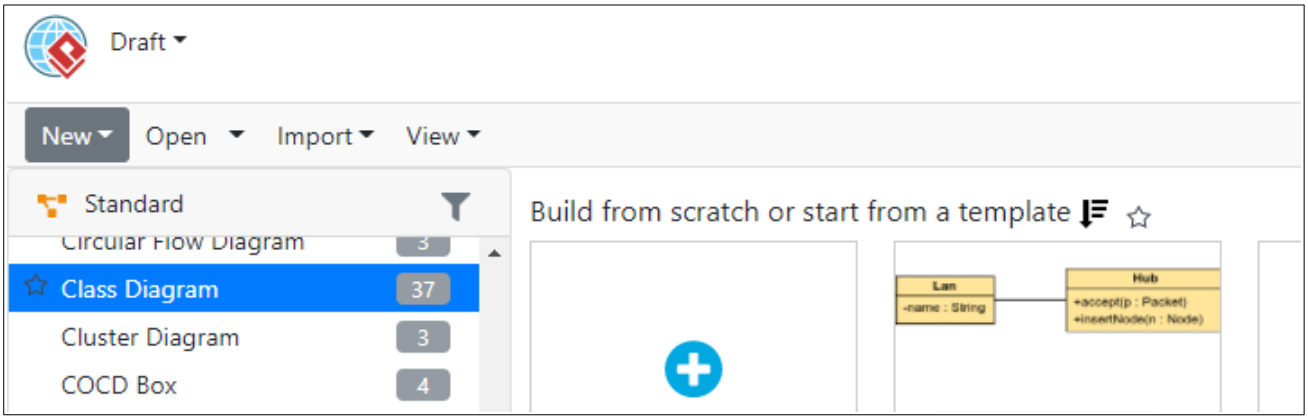
PRÁCTICA 02: DIAGRAMAS DE CLASES (II)	UD 08: DIAGRAMAS DE ESTRUCTURA
Contenidos: 1. Justificación de la práctica 2. Decálogo de recomendaciones 3. EJERCICIOS: 1. VIAJE 2. VIDEOJUEGO 3. ASOCIACIÓN 4. TORNEO 4. Bibliografía	
PRÁCTICA NO EVALUABLE	

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 realizar diagramas de clases en Visual Paradigm, YUML y DRAW.IO, **aunque puedes usar cualquier otro editor**.





2. DECÁLOGO DE RECOMENDACIONES

[clases]

1. Nombra las clases con sustantivos en singular, UpperCamelCase y en cursiva si son abstractas.
2. No incluyas las clases que no representen una entidad del sistema como “main”, “test”

[atributos/campos/propiedades]

3. Nombra los atributos con sustantivos lowerCamelCase y en cursiva si son abstractos.
4. Los atributos de una clase suelen ser privados.
5. Los tipos de datos de los atributos suelen ser opcionales (diseño o implementación).

[métodos/operaciones/funciones]

6. Nombra los métodos con verbos lowerCamelCase y solo en cursiva si son abstractos.
7. No incluyas setters, getters, constructores ni destructores, salvo si te los piden.
8. Los métodos suelen ser públicos y los parámetros opcionales (diseño o implementación).

[relaciones/asociaciones]

9. Marca las asociaciones con un rombo relleno (composición) o vacío (agregación).
10. Etiqueta las asociaciones solo cuando sea necesario, con una, dos o hasta tres etiquetas.

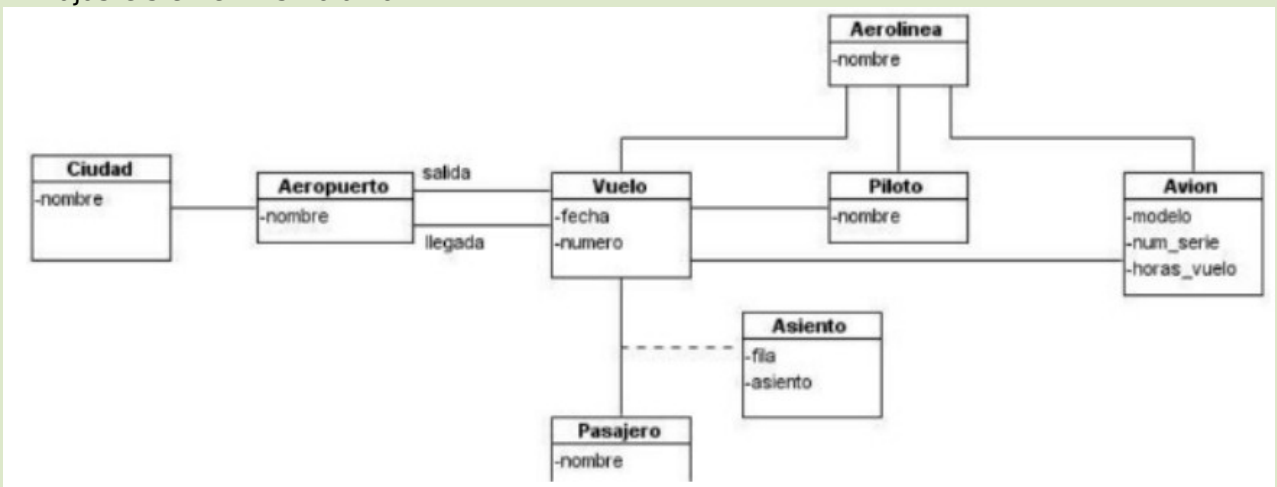
		Mismo paquete		Otro paquete	
		Subclase	Otra	Subclase	Otra
-	private	*no*	*no*	*no*	*no*
#	protected	*sí*	*sí*	*sí*	*no*
+	public	*sí*	*sí*	*sí*	*sí*
~	*package*	*sí*	*sí*	*no*	*no*

3. EJERCICIOS

PTCA 2. EJERCICIO 1: VIAJE	ENUNCIADO

Completa con los adornos SOBRE LAS RELACIONES necesarios el diagrama de un viaje a París:

- Incluye, al menos, una instancia para cada clase.
- Asume que viajas con todos los miembros del equipo.
- El capitán Wilson fue el piloto del viaje de ida y el capitán López del de vuelta.
- Los números de asiento que os tocaron en ambos viajes fueron distintos, pero casualmente viajasteis en el mismo avión.



SOLUCIÓN en documento de soluciones

PTCA 2. EJERCICIO 2: VIDEOJUEGO	ENUNCIADO

Con los siguientes requisitos para crear un videojuego:

- En el videojuego (en cada partida o escenario) aparecerá: 1 jugador, varios enemigos, varios objetos estáticos (inmóviles) y varios objetos dinámicos (móviles).
- Todos los elementos anteriores dispondrán de un identificador y unas coordenadas (X e Y) para ubicarlos.
- Jugadores y enemigos disponen de: identificador, nivel de energía y número de vidas.
- Los jugadores pueden coger hasta tres objetos móviles y tres estáticos.
- Jugadores y enemigos pueden dispararse entre sí y pueden moverse a otras coordenadas.
- Las armas son objetos móviles y constan de un nivel de energía y de un factor potenciador de la capacidad ofensiva de los personajes.

Realiza un diagrama de clases representando un escenario con 1 jugador y varios enemigos.

Tienes varias pistas más abajo, por si no sabes por dónde empezar.

TEXTO DE AYUDA:

Pista1:

Las relaciones entre clases pueden ser de tres tipos:

HERENCIA: Las características (métodos y/o atributos) de una clase (hija o subclase) las comparte con otra (padre o superclase).

ASOCIACIÓN: Cuando las clases interactúan entre sí mediante métodos o acciones como puede ser alumno ------ asignatura. En este caso, la instancia de la clase alumno llamará a un método de la clase alumno llamado matricular con una instancia de la clase asignatura como parámetro.

AGREGACIÓN/COMPOSICIÓN/: Una clase se compone de otra, de manera que la parte sin el todo puede o no puede existir.

Pista2: *Que un jugador o que un enemigo se pueda -mover- podría entenderse como una relación de asociación consigo mismo con la etiqueta -mover-. En código, sería algo así:*

```
x=2;y=3;
monstruoVerde = new Enemigo(x,y);
x=22;y=33;
monstruoVerdeNuevaPosicion= new Enemigo(x,y);
monstruoVerde.x=mover(monstruoVerdeNuevaPosicion);
```

Pista3: *La acción de -disparar- puede entenderse como una relación entre jugadores y enemigos. En código, sería algo así:*

```
personaje = new Personaje(...);
monstruoVerde= new Enemigo(...);
...
personaje.disparar(monstruoVerde);
```

Pista4: *No hay agregaciones ni composiciones.*

SOLUCIÓN en documento de soluciones

PTCA 2. EJERCICIO 3: ASOCIACIÓN	ENUNCIADO

- Crear un proyecto UML llamado Asociación en el que se diseñe un diagrama de clases que modele el proceso de dar de alta a cada una de las personas que se apuntan a una asociación.
- De cada persona interesa saber sus datos básicos: NIF, nombre completo y fecha de nacimiento. Cuando cada nuevo socio se da de alta, se le asigna un código de asociado alfanumérico y se anota la fecha de alta.
- La clase Fecha se modela con tres campos (día, mes y año) de tipo entero. La clase Nif se modela con un campo de tipo entero llamado dni y un campo de tipo carácter llamado letra.

SOLUCIÓN en documento de soluciones

PTCA 2. EJERCICIO 4: TORNEO	ENUNCIADO

Tenemos una aplicación “Torneo” que manejar los datos de los encuentros de un torneo de tenis de mesa en la modalidad de sorteo y eliminatoria.

Del torneo interesa conocer la fecha del torneo, los encuentros celebrados y el ganador. De cada jugador, que debe de conocer perfectamente las reglas, interesa saber el número de federado de la federación de la que es miembro.

De cada persona interesa saber sus datos básicos: NIF, nombre completo y fecha de nacimiento. La clase Fecha se modela con tres campos (día, mes y año) de tipo entero. La clase Nif se modela con un campo de tipo entero llamado dni y un campo de tipo carácter llamado letra.

De cada encuentro interesa conocer los oponentes, el ganador y el resultado final del marcador de cada una de las tres partidas que se juegan a 21 puntos

SOLUCIÓN en documento de soluciones

4. BIBLIOGRAFÍA Y ENLACES

- Aldarias, F. (2012): “Entornos de desarrollo”, CEEDCV
- Casado, C. (2012):Entornos de desarrollo, RA-MA, Madrid
- Ramos, A.; Ramos, MJ (2014):Entornos de desarrollo, Garceta, Madrid
- José A. Pacheco Ondoño Con el mazo dando https://joanpaon.wordpress.com/
- Visual Paradigm,www.visual-paradigm.com