

UD 8

DIAGRAMAS DE ESTRUCTURA

ENTORNOS DE DESARROLLO 20/21
CFGS DAW

PRÁCTICA 02 [NO EVALUABLE]

DIAGRAMAS DE CLASES (II)

SOLUCIONADA

Revisado por:

Sergio Badal

Autores:

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

Fecha:

13/02/21

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.

PRÁCTICA 02: DIAGRAMAS DE CLASES (II)

Contenidos:

1. Justificación de la práctica
2. Decálogo de recomendaciones
3. EJERCICIOS:
 1. VIAJE
 2. VIDEOJUEGO
 3. ASOCIACIÓN
 4. TORNEO
4. Bibliografía

UD 08: DIAGRAMAS DE ESTRUCTURA

Temporalización:

Del Lun 15-feb al Vie 19-feb (1 semana)

Revisado por:

Sergio Badal

Autores:

Cristina Álvarez

Fco. Javier Valero Garzón

M.ª Carmen Safont

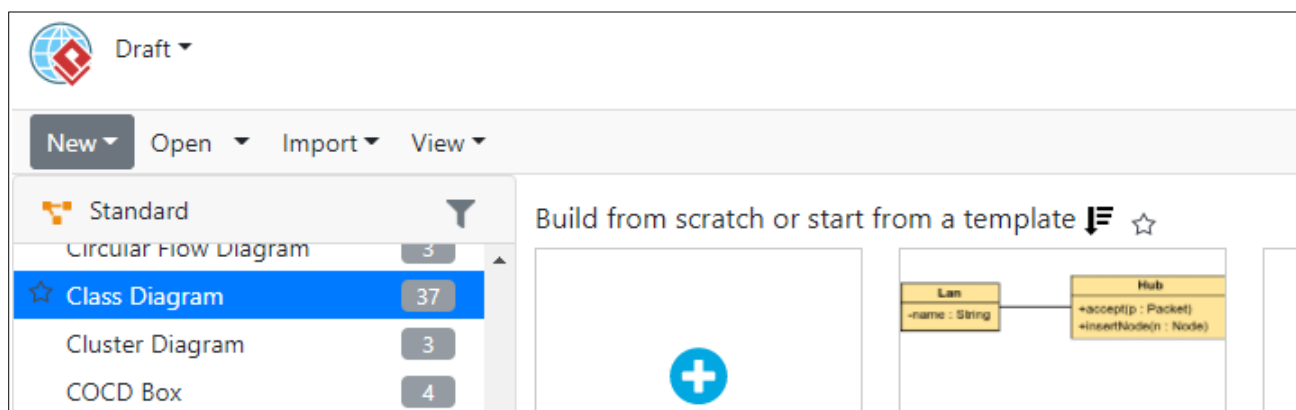
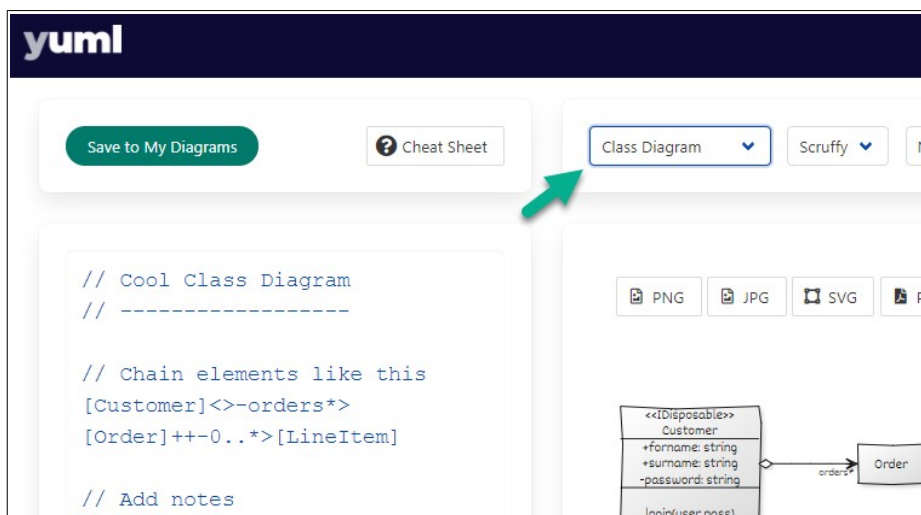


PRÁCTICA NO EVALUABLE

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	<i>no</i>	<i>no</i>	<i>no</i>	<i>no</i>
#	protected	<i>sí</i>	<i>sí</i>	<i>sí</i>	<i>no</i>
+	public	<i>sí</i>	<i>sí</i>	<i>sí</i>	<i>sí</i>
~	<i>package</i>	<i>sí</i>	<i>sí</i>	<i>no</i>	<i>no</i>

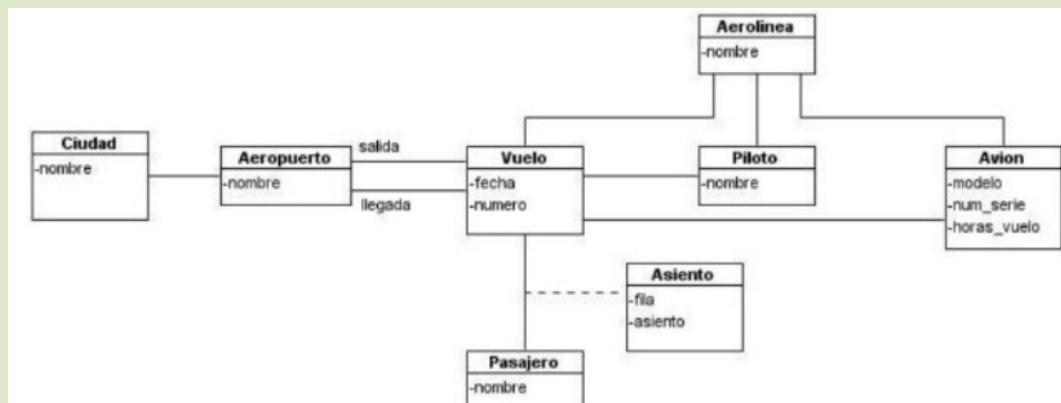
3. EJERCICIOS

PTCA 2. EJERCICIO 1: VIAJE

ENUNCIADO

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

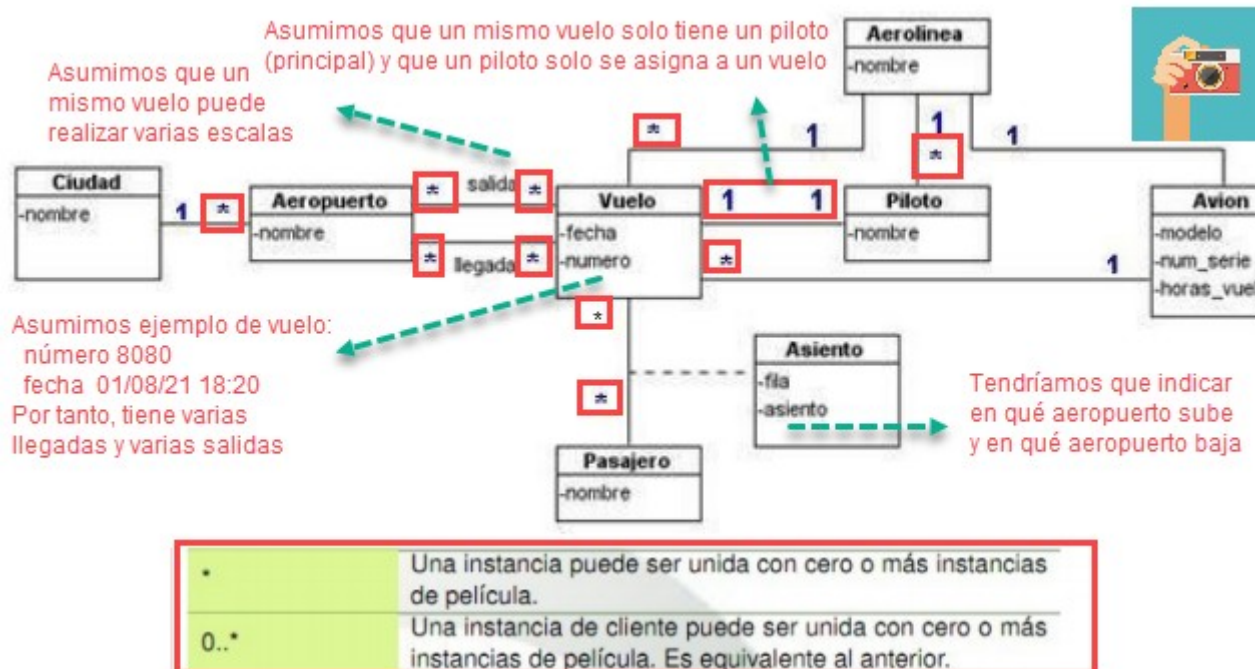
- Incluye, al menos, una instancia para cada clase.
- Asume que viajas con todos los miembros del equipo.
- El capitán Wilson fue el piloto del viaje de ida y el capitán López del de vuelta.
- Los números de asiento que os tocaron en ambos viajes fueron distintos, pero casualmente viajasteis en el mismo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



SOLUCIÓN en la siguiente página ...

PTCA 2. EJERCICIO 1: VIAJE

POSIBLE SOLUCIÓN



MUY IMPORTANTE:

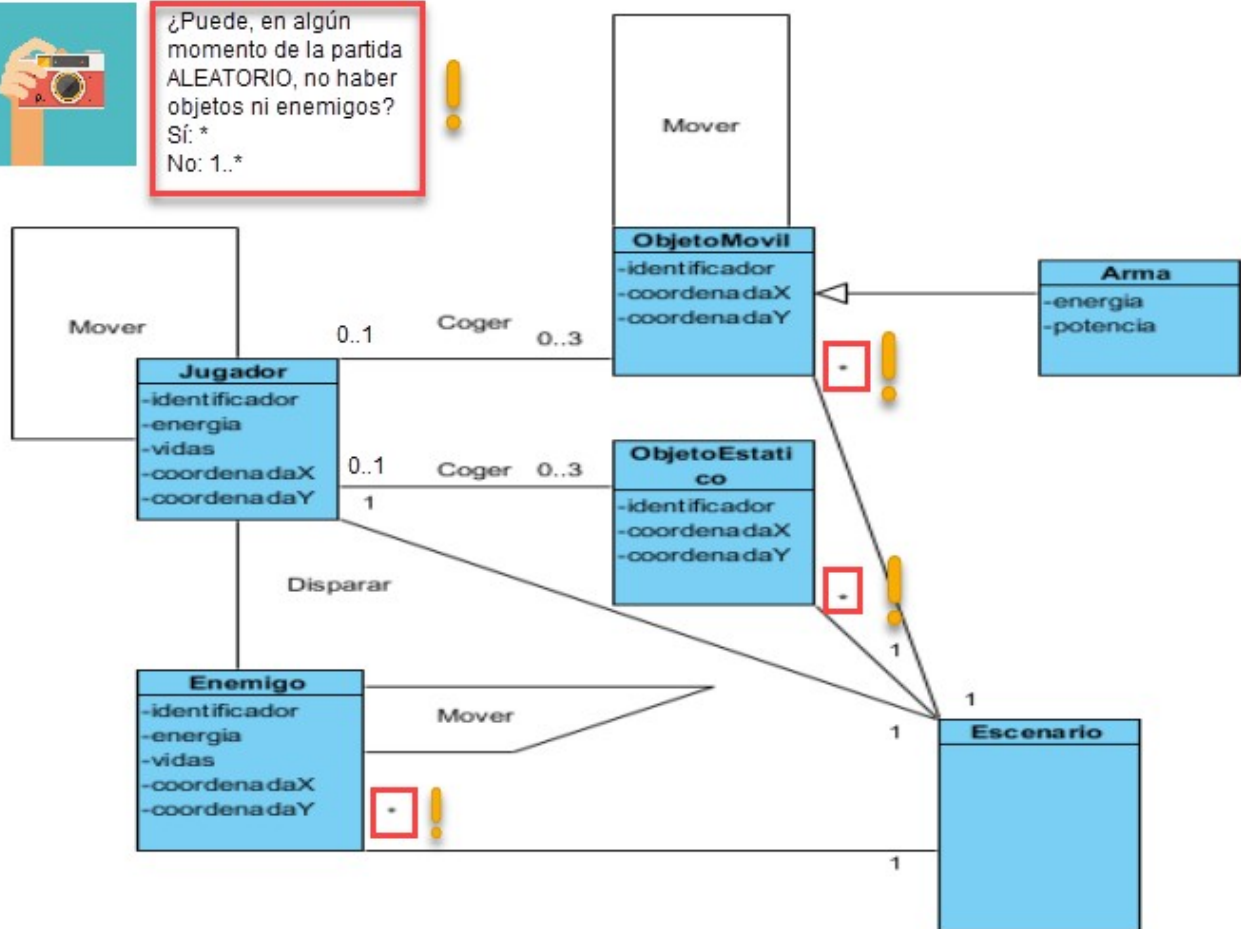
- Recuerda que, cualquier elemento del diagrama que no esté en el enunciado o que pueda causar confusión **a los desarrolladores**, debe ser justificado por escrito.
 - Nuestra propuesta contiene 4 aclaraciones. ¿Te parecen necesarias? ¿pondrías más? ¿pondrías menos?
- ¿Por qué nos referimos “a los desarrolladores”?
 - Cuidado, ¡estamos en fase de DISEÑO! Rara vez el cliente entra en fase de diseño, sobre todo en metodologías tradicionales que es donde más se usan diagramas como este.
- La relación * significa cero o más instancias** por lo que, si consideras que una ciudad no debe estar en el sistema si no tiene aeropuertos, puedes incluir **Ciudad (1)---(1..*) Aeropuerto** y sería válido. Lo mismo con todas las relaciones *.
- Es crucial que entiendas que un diagrama estático (de datos) es una **FOTOGRAFÍA DEL SISTEMA EN FUNCIONAMIENTO EN UN INSTANTE ALEATORIO**. Esto es aplicable también a Entidad-Relación, de base de datos, y a cualquier diagrama que represente datos o entidades.
 - Pero... si la relación entre piloto y vuelo es de 1 a 1 ¿me dejará el sistema crear un vuelo sin un piloto y viceversa? Por supuesto, porque **estos diagramas estáticos muestran una foto del sistema ¡cuando el sistema está 100% operativo!**

PTCA 2. EJERCICIO 2: VIDEOJUEGO	ENUNCIADO
<p>Con los siguientes requisitos para crear un videojuego:</p> <ul style="list-style-type: none">• 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. <p>Realiza un diagrama de clases representando un escenario con 1 jugador y varios enemigos.</p>	
SOLUCIÓN en la siguiente página ...	

PTCA 2. EJERCICIO 2: VIDEOJUEGO

POSIBLE
SOLUCIÓN

¿Puede, en algún momento de la partida ALEATORIO, no haber objetos ni enemigos?
Sí: *
No: 1..*



MULTIPLICIDAD	SIGNIFICADO
1	Una instancia de cliente debe ser unida con exactamente una instancia de película no más y no menos.
*	Una instancia puede ser unida con cero o más instancias de película.
0..*	Una instancia de cliente puede ser unida con cero o más instancias de película. Es equivalente al anterior.
0..1	Una instancia de cliente debe ser unida con cero o una instancia de película. Esto se llama multiplicidad opcional.
1..*	Una instancia de Cliente debe ser unida con al menos una o más instancias de película.
5..9	Una instancia de cliente puede ser unida con al menos cinco instancias de película pero no más de 9.
3,5,7	Una instancia de cliente puede ser unido a 3, o 5 o 7 instancias de película.

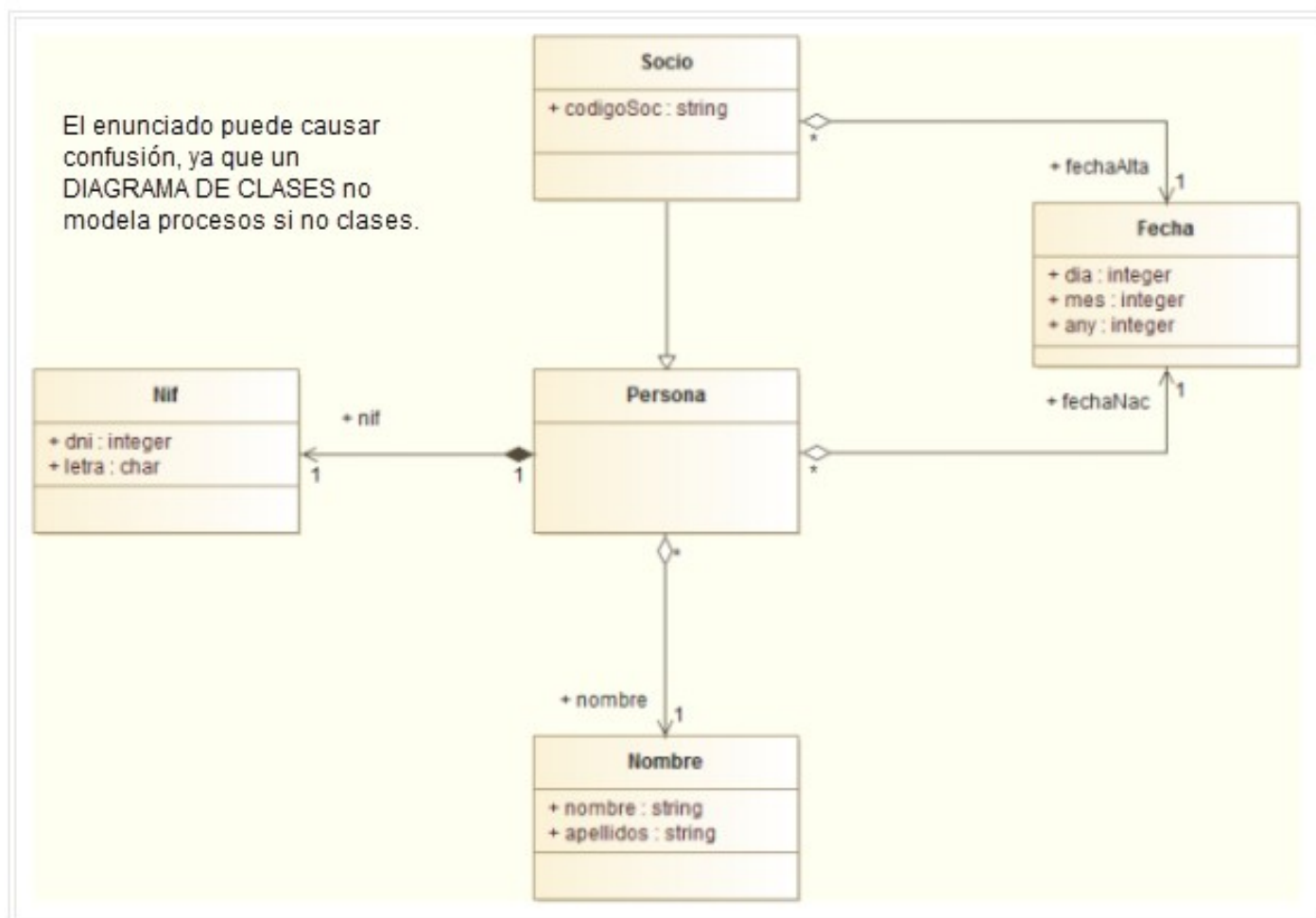
PTCA 2. EJERCICIO 3: ASOCIACIÓN

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 DETALLADA:

<https://joanpaon.wordpress.com/2013/07/01/uml-diagrama-de-clases-ejercicio-1/>



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.

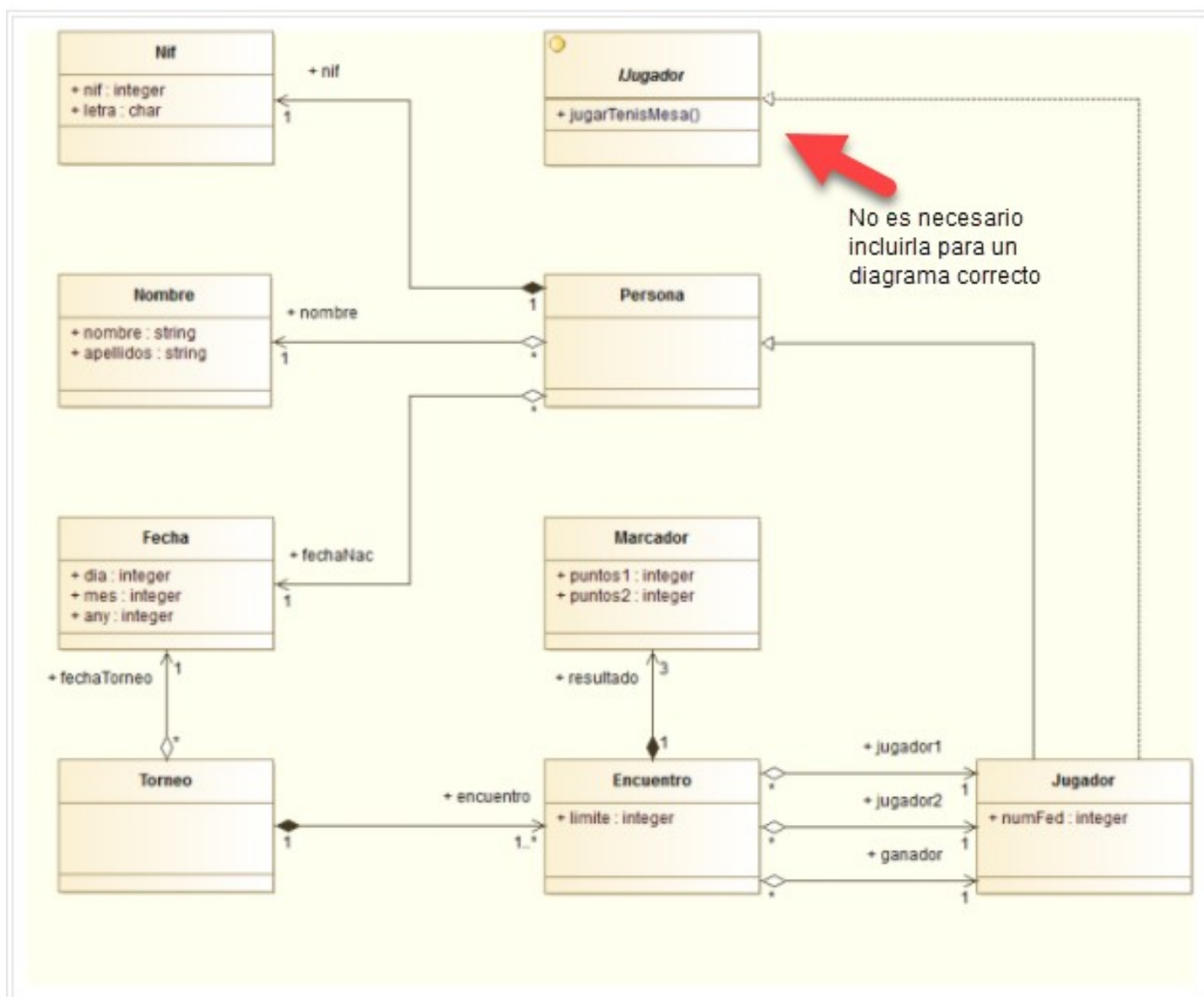
Del torneo interesa conocer la fecha del torneo, los encuentros celebrados y el ganador. De cada jugador, que debe de conocer perfectamente las reglas, interesa saber el número de federado de la federación de la que es miembro.

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 DETALLADA:

<https://joanpaon.wordpress.com/2013/07/24/uml-diagrama-de-clases-ejercicio-2/>



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