

**Realiza un proyecto con tu nombre y 1º apellido (NombrePrimerApellido).**

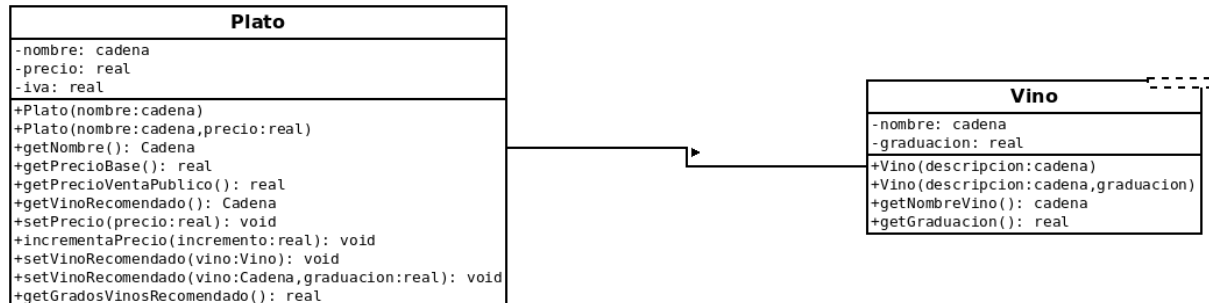
**Exporta el proyecto en formato zip y súbelo a la plataforma.**

**⇒ Recuerda comentar y documentar tu código.**

Criterio	Peso del criterio	Nota en la prueba 1
2.a) Se han identificado los fundamentos de la programación orientada a objetos.	0,1	0,91%
2.b) Se han escrito programas simples.	0,1	0,91%
2.c) Se han instanciado objetos a partir de clases predefinidas.	0,1	0,91%
2.d) Se han utilizado métodos y propiedades de los objetos.	0,1	0,91%
2.f) Se han utilizado parámetros en la llamada a métodos.	0,1	0,91%
2.g) Se han incorporado y utilizado librerías de objetos.	0,1	0,91%
2.h) Se han utilizado constructores.	0,1	0,91%
2.i) Se ha utilizado el entorno integrado de desarrollo en la creación y compilación de programas simples	0,1	0,91%
4.a) Se ha reconocido la sintaxis, estructura y componentes típicos de una clase.	0,1	0,91%
		8,18%
4.b) Se han definido clases.	0,5	4,55%
4.c) Se han definido propiedades y métodos.	0,5	4,55%
4.d) Se han creado constructores.	0,5	4,55%
4.e) Se han desarrollado programas que instancian y utilizan objetos de las clases creadas anteriormente.	0,5	4,55%
4.f) Se han utilizado mecanismos para controlar la visibilidad de las clases y de sus miembros.	0,5	4,55%
4.h) Se han creado y utilizado métodos estáticos.	0,5	4,55%
2.e) Se han escrito llamadas a métodos estáticos.	0,5	4,55%
		31,82%
6a) Se han escrito programas que utilicen arrays	3,6	32,73%
		32,73%
3.d) Se ha escrito código utilizando control de excepciones.	2	18,18%
		18,18%
7.h) Se ha comentado y documentado el código.	1	9,09%
		9,09%

## Orientación a objeto

Nos piden realizar una aplicación para gestionar la carta de un restaurante. Se ha diseñado el siguiente UML.



Se debe tener en cuenta que:

- El iva de los platos será común para todos los platos y no podrá modificarse. En principio será el 21%, pero deja preparado el programa para que sea fácil de modificar.
- Si se intenta crear un plato con precio igual o menor a 0 euros se deberá lanzar una excepción.
- Si se intenta crear un vino con una graduación menor que 0, se deberá lanzar una excepción.
- Un plato será igual al otro siempre y cuando el nombre sea igual al otro.
- Un vino será igual al otro siempre que tengan el mismo nombre.
- Crear los métodos toString en cada una de las clases, aunque no se disponga en el UML. El toString del plato deberá mostrar el nombre del plato, el precio sin iva, el precio venta público, y el nombre y la graduación del vino recomendado. Si no tiene vino recomendado deberá poner "Sin recomendaciones para el vino".
- Se deberá poder consultar el porcentaje de Iva que se aplica con el formato Plato.getIva();

Además se deberá crear un programa principal que pueda almacenar los platos del restaurante y muestre el siguiente menú:

1. Dar de alta un plato (Se deberá solicitar el nombre y precio del plato). Si el plato ya existe se deberá mostrar un mensaje de error. Si el plato no se puede crear se debe informar al respecto.
2. Modificar el precio de un plato (Se deberá pedir el nombre del plato, si el plato no existe informar del error, en caso contrario pedir el nuevo precio y actualizar el precio.)
3. Asignar un vino a un plato. Se preguntará el nombre del plato, el nombre del vino y la graduación. Informar si hay error.
4. Mostrar información de un plato. (Se pedirá el nombre del plato, error si no existe).
5. Mostrar información de todos los platos.
6. Salir.