



## OBJETIVOS

- Diseño de tipos: gestión de restricciones mediante excepciones y métodos *checkers* privados.
- Creación de clases de tipo Excepción.
- Test de métodos con captura de excepciones.

## IMPLEMENTACIÓN DE LAS RESTRICCIONES SOBRE LOS TIPOS

Añada a cada clase el código necesario para tener en cuenta las restricciones que pesan sobre algunas de sus propiedades, y que se describen a continuación. Si necesita utilizar excepciones en el código de las clases, implemente dichas **excepciones**.

El código numérico de una **asignatura** consta de 7 dígitos (por ejemplo, “0000230”). El número de créditos es siempre mayor que cero y su curso debe tener en cuenta que los grados constan de 4 años. Si en el código necesita informar de determinados errores, implemente y utilice una excepción de nombre `ExcepcionAsignaturaNoValida`.

Las **becas** se identifican mediante un código alfanumérico único, formado por 3 letras seguidas de 4 dígitos. La duración mínima de una beca es de 1 mes, y la cuantía mínima es de 1500.0 euros (recuerde que la cuantía mínima podría variar en futuras versiones de la aplicación). Si en el código necesita informar de determinados errores, implemente y utilice una excepción de nombre `ExcepcionBecaNoValida`.

El DNI de una **persona** está formado por 8 dígitos y una letra, que se calcula mediante una fórmula conocida<sup>1</sup>. Además, su email debe contener el carácter ‘@’, o bien ser la cadena vacía si la persona aún no ha comunicado ese dato. Si en el código necesita informar de determinados errores, implemente y utilice una excepción de nombre `ExcepcionPersonaNoValida`.

El **espacio** tiene una única restricción, y es que su capacidad debe ser mayor que 0. Si en el código necesita informar de determinados errores, implemente y utilice una excepción de nombre `ExcepcionEspacioNoValido`.

Las **notas** tienen un valor numérico comprendido entre 0 y 10, ambos inclusive. Por otro lado, una nota sólo puede tener mención de honor si su valor numérico es igual o superior a 9. Si en el código necesita informar de determinados errores, implemente y utilice una excepción de nombre `ExcepcionNotaNoValida`.

Una **tutoría** sólo puede darse de lunes a viernes y debe tener una duración de al menos 15 minutos. Si en el código necesita informar de determinados errores, implemente y utilice una excepción de nombre `ExcepcionTutoriaNoValida`.

<sup>1</sup> Véase [http://es.wikipedia.org/wiki/Número\\_de\\_identificación\\_fiscal](http://es.wikipedia.org/wiki/Número_de_identificación_fiscal)



## **REALIZACIÓN DE TEST DE CAPTURA DE EXCEPCIONES**

---

Implemente un test para cada tipo en el que se comprueben sus restricciones. Debe probar tanto los constructores como los métodos que modifiquen propiedades con restricciones. En el caso de los constructores, intente construir objetos con valores incorrectos en las propiedades que tengan restricciones, y compruebe que se lanzan las excepciones correspondientes. En el caso de los métodos modificadores, construya objetos con valores correctos y posteriormente intente asignar valores incorrectos a las propiedades con restricciones, comprobando que se lanzan las excepciones correspondientes.

## **NOTA SOBRE LOS CONSTRUCTORES**

---

En el boletín T2 ha creado métodos constructores para todos los tipos. Para que el trabajo práctico pueda ser evaluado correctamente en futuros test, es muy importante que el orden en el que aparecen los parámetros en los constructores sea el adecuado. Revise todos los constructores y haga que el orden de los parámetros en los constructores coincida con el orden de aparición de las propiedades correspondientes en la descripción del tipo publicada en el boletín T1.