

La clase `Persona` permite guardar información sobre un individuo (nombre, apellidos, dni, fecha de nacimiento, género), para lo que dispone de varios métodos y constructores.

A su vez, posee varios métodos, algunos de ellos sobrecargados:

- **`validarDatos`**: valida la información proporcionada y devuelve `true` si es correcta, `false` en caso contrario.
- **`bonoJovenDisponible`**: comprueba si la persona tiene derecho al bono joven, que está disponible para mayores de 18 y menores de 30.
- **`obtenerEdad`**: devuelve la edad de la persona para la fecha actual.
- **`obtenerEdadPara`**: recibe un año y calcula la edad de la persona a finales de ese año

Si la información introducida para una persona es incorrecta (no supera la validación de los datos), los constructores con parámetros lanzan la excepción correspondiente. En caso contrario, se genera un objeto que almacena la información introducida.

1. Basándote en esta especificación y en el código proporcionado, desarrolla los tests unitarios que proporcionen la máxima cobertura para estos cuatro métodos y los constructores con parámetros.
2. Esta clase posee múltiples ejemplos de malas prácticas que hemos visto en clase. Una vez realizado los tests, corrige estas malas prácticas y refactoriza el código donde es conveniente, garantizando que los tests siguen funcionando correctamente. Justifica las modificaciones en código.
3. Incluye tests parametrizados para el método **`obtenerEdadPara`**

⇒ Opcional 1: Incluye tests parametrizados por métodos (`@MethodSource`).

⇒ Opcional 2: Modifica los constructores y el método de validación para que se informe de los tipos de error cometidos al introducir datos incorrectos (p.ej.: El campo nombre está vacío. La fecha de nacimiento no puede ser mayor que la fecha actual. etc.).

** No se permite el uso de clase principal `main`.