



## OBJETIVO

Este documento contiene material de apoyo para el boletín T3 del trabajo práctico. En concreto, el documento incluye las clases y los tests que se trabajarán en el aula. **Sólo se muestran los métodos nuevos o modificados con respecto a las clases implementadas en el boletín anterior.** El profesor los explicará en la sesión de laboratorio y responderá a sus dudas al respecto. El alumno, por su parte, debe añadir el código a su proyecto de curso, y realizar por su cuenta el resto de ejercicios del boletín.

## CLASES

Clase AsignaturaImpl

```
public AsignaturaImpl(String nombre, String codigo, Double creditos,
    TipoAsignatura tipo, Integer curso) {
    checkCodigo(codigo);
    checkCreditos(creditos);
    checkCurso(curso);

    this.nombre = nombre;
    this.codigo = codigo;
    this.creditos = creditos;
    this.tipo = tipo;
    this.curso = curso;
}

private void checkCodigo(String codigo) {
    Boolean esCorrecto = codigo.length() == 7 &&
        Character.isDigit(codigo.charAt(0)) &&
        Character.isDigit(codigo.charAt(1)) &&
        Character.isDigit(codigo.charAt(2)) &&
        Character.isDigit(codigo.charAt(3)) &&
        Character.isDigit(codigo.charAt(4)) &&
        Character.isDigit(codigo.charAt(5)) &&
        Character.isDigit(codigo.charAt(6));
    if(!esCorrecto) {
        throw new ExcepcionAsignaturaNoValida(
            "El código debe estar formado por 7 dígitos.");
    }
}

private void checkCreditos(Double creditos){
    if (creditos <= 0) {
        throw new ExcepcionAsignaturaNoValida(
            "Los créditos deben ser un número positivo distinto de 0.");
    }
}

private void checkCurso(Integer curso){
    if (curso < 1 || curso > 4) {
        throw new ExcepcionAsignaturaNoValida(
            "El curso debe estar comprendido entre 1 y 4.");
    }
}
```

## Clase BecaImpl

```
public BecaImpl(String codigo, TipoBeca tipo) {
    checkCodigo(codigo);

    this.codigo = codigo;
    this.tipo = tipo;
    this.cuantiaTotal = CUANTIA_MINIMA;
    this.duracion = 1;
}

public BecaImpl(String codigo, Double cuantiaTotal, Integer duracion, TipoBeca tipo) {
    checkCodigo(codigo);
    checkCuantiaTotal(cuantiaTotal);
    checkDuracion(duracion);

    this.codigo = codigo;
    this.tipo = tipo;
    this.cuantiaTotal = cuantiaTotal;
    this.duracion = duracion;
}

private void checkCodigo(String codigo) {
    boolean esCorrecto= codigo.length() == 7
        && Character.isLetter(codigo.charAt(0))
        && Character.isLetter(codigo.charAt(1))
        && Character.isLetter(codigo.charAt(2))
        && Character.isDigit(codigo.charAt(3))
        && Character.isDigit(codigo.charAt(4))
        && Character.isDigit(codigo.charAt(5))
        && Character.isDigit(codigo.charAt(6));

    if(!esCorrecto){
        throw new ExcepcionBecaNoValida(
            "El código debe estar formado por 3 letras y 4 dígitos.");
    }
}

private void checkCuantiaTotal(Double cuantiaTotal){
    if (cuantiaTotal < CUANTIA_MINIMA) {
        throw new ExcepcionBecaNoValida(
            "La cuantía total debe ser como mínimo " + CUANTIA_MINIMA
            + " euros.");
    }
}

private void checkDuracion(Integer duracion){
    if (duracion < 1) {
        throw new ExcepcionBecaNoValida(
            "La duración debe ser de al menos un mes.");
    }
}

public void setCuantiaTotal(Double cuantiaTotal) {
    checkCuantiaTotal(cuantiaTotal);
    this.cuantiaTotal = cuantiaTotal;
}

public void setDuracion(Integer duracion) {
    checkDuracion(duracion);
    this.duracion = duracion;
}
```

## Clase PersonaImpl

```
public PersonaImpl(String dni, String nombre, String apellidos,
    LocalDate fechaNacimiento, String email) {
    checkDni(dni);
    checkEmail(email);
    checkFechaNacimiento(fechaNacimiento);

    this.nombre = nombre;
    this.apellidos = apellidos;
    this.dni = dni;
    this.fechaNacimiento = fechaNacimiento;
    this.email = email;
}

public PersonaImpl(String dni, String nombre, String apellidos,
    LocalDate fechaNacimiento) {
    this(dni,nombre,apellidos,fechaNacimiento,"");
}

private void checkDni(String dni){
    boolean esDniCorrecto = checkDniTipoCaracteres(dni) && checkDniLetra(dni);

    if (!esDniCorrecto) {
        throw new ExcepcionPersonaNoValida(
            "DNI incorrecto.");
    }
}

private Boolean checkDniTipoCaracteres(String dni){
    return dni.length() == 9 && Character.isDigit(dni.charAt(0))
        && Character.isDigit(dni.charAt(1))
        && Character.isDigit(dni.charAt(2))
        && Character.isDigit(dni.charAt(3))
        && Character.isDigit(dni.charAt(4))
        && Character.isDigit(dni.charAt(5))
        && Character.isDigit(dni.charAt(6))
        && Character.isDigit(dni.charAt(7))
        && Character.isLetter(dni.charAt(8));
}

private Boolean checkDniLetra(String dni){
    String letras = "TRWAGMYFPDXBNJZSQVHLCKE";
    Integer numeroDni = new Integer(dni.substring(0, 8));
    return dni.charAt(8)==letras.charAt(numeroDni%23);
}

private void checkEmail(String email){
    if (!(email.isEmpty() || email.contains("@"))) {
        throw new ExcepcionPersonaNoValida(
            "El email debe contener el usuario, una arroba y el servidor.");
    }
}

private void checkFechaNacimiento(LocalDate fechaNacimiento) {
    if(!fechaNacimiento.isBefore(LocalDate.now())){
        throw new ExcepcionPersonaNoValida(
            "La fecha de nacimiento de una persona debe ser anterior " +
            "a la fecha actual del sistema.");
    }
}
```



(continúa...)

```
public void setDNI(String dni) {  
    checkDni(dni);  
    this.dni = dni;  
}  
  
public void setFechaNacimiento(LocalDate fecha) {  
    checkFechaNacimiento(fecha);  
    this.fechaNacimiento = fecha;  
}  
  
public void setEmail(String email) {  
    checkEmail(email);  
    this.email = email;  
}
```

## TESTS

---

Las clases `TestAsignatura`, `TestBeca` y `TestPersona` se proporcionan en ficheros independientes. En el caso de `TestPersona` faltan algunos casos de prueba por implementar, que serán resueltos en clase por su profesor.