



OBJETIVO

Este documento contiene material de apoyo para el boletín T5 del trabajo práctico. En concreto, el documento incluye las interfaces y clases que se trabajarán en el aula. El profesor las 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.

INTERFACES

Interfaz Alumno

```
package fp.grados.tipos;

import java.util.Set;

public interface Alumno extends Persona {
    Set<Asignatura> getAsignaturas();
    // TODO: Integer getCurso();
    void matriculaAsignatura(Asignatura a);
    void eliminaAsignatura(Asignatura a);
    Boolean estaMatriculadoEn(Asignatura a);
}
```

Interfaz Becario

```
package fp.grados.tipos;

import java.time.LocalDate;

public interface Becario extends Alumno {
    Beca getBeca();
    LocalDate getFechaComienzo();
    void setFechaComienzo(LocalDate fecha);
    LocalDate getFechaFin();
}
```

CLASES

Clase AlumnoImpl

```
package fp.grados.tipos;

import java.time.LocalDate;
import java.util.HashSet;
import java.util.Set;

import fp.grados.excepciones.ExcepcionAlumnoNoValido;
import fp.grados.excepciones.ExcepcionAlumnoOperacionNoPermitida;

public class AlumnoImpl extends PersonaImpl implements Alumno {
    private Set<Asignatura> asignaturas;

    public AlumnoImpl(String dni, String nombre, String apellidos,
        LocalDate fechaNacimiento, String email) {
        super(dni, nombre, apellidos, fechaNacimiento, email);
        checkEmailUniversidad(email);
        this.asignaturas = new HashSet<Asignatura>();
    }

    private void checkEmailUniversidad(String email) {
        if (!email.endsWith("@alum.us.es")) {
            throw new ExcepcionAlumnoNoValido(
                "El email de un alumno debe terminar en @alum.us.es.");
        }
    }

    // Redefinimos setEmail() para añadir restricción nueva
    public void setEmail(String email) {
        checkEmailUniversidad(email);
        super.setEmail(email);
    }

    public Set<Asignatura> getAsignaturas() {
        return new HashSet<Asignatura>(asignaturas);
    }

    public void matriculaAsignatura(Asignatura a) {
        if (estaMatriculadoEn(a)) {
            throw new ExcepcionAlumnoOperacionNoPermitida(
                "El alumno ya está matriculado en esta asignatura.");
        }
        asignaturas.add(a);
    }

    public void eliminaAsignatura(Asignatura a) {
        if (!estaMatriculadoEn(a)) {
            throw new ExcepcionAlumnoOperacionNoPermitida(
                "El alumno no está matriculado en esta asignatura.");
        }
        asignaturas.remove(a);
    }

    public Boolean estaMatriculadoEn(Asignatura a) {
        return asignaturas.contains(a);
    }

    public String toString() {
        // TODO: usar propiedad curso cuando se implemente.
        return "(?) " + super.toString();
    }
}
```

Clase BecarioImpl

```
package fp.grados.tipos;

import java.time.LocalDate;

import fp.grados.excepciones.ExcepcionBecarioNoValido;

public class BecarioImpl extends AlumnoImpl implements Becario {
    private Beca beca;
    private LocalDate fechaComienzo;

    public BecarioImpl(String dni, String nombre, String apellidos,
        LocalDate fechaNacimiento, String email, Beca beca,
        LocalDate fechaComienzo) {
        super(dni, nombre, apellidos, fechaNacimiento, email);

        checkFechaComienzo(fechaComienzo);

        this.beca = beca;
        this.fechaComienzo = fechaComienzo;
    }

    public BecarioImpl(String dni, String nombre, String apellidos,
        LocalDate fechaNacimiento, String email, String codigo,
        Double cuantiaTotal, Integer duracion, TipoBeca tipo,
        LocalDate fechaComienzo) {
        super(dni, nombre, apellidos, fechaNacimiento, email);

        checkFechaComienzo(fechaComienzo);

        this.beca = new BecaImpl(codigo, cuantiaTotal, duracion, tipo);
        this.fechaComienzo = fechaComienzo;
    }

    private void checkFechaComienzo(LocalDate fechaComienzo) {
        if (!fechaComienzo.isAfter(LocalDate.now())) {
            throw new ExcepcionBecarioNoValido(
                "La fecha de comienzo de la beca debe ser posterior a la actual.");
        }
    }

    public Beca getBeca() {
        return beca;
    }

    public LocalDate getFechaComienzo() {
        return fechaComienzo;
    }

    public void setFechaComienzo(LocalDate fecha) {
        checkFechaComienzo(fecha);
        fechaComienzo = fecha;
    }

    public LocalDate getFechaFin() {
        return getFechaComienzo().plusMonths(getBeca().getDuracion());
    }

    // Redefinimos setEmail para no permitir usar esta operación (la propiedad
    // pasa a ser sólo consultable para el becario)
    public void setEmail(String email) {
        throw new UnsupportedOperationException(
            "No se permite cambiar el email para un becario.");
    }

    public String toString() {
        return super.toString() + " " + getBeca().toString();
    }
}
```



TESTS

Las clases `TestAlumno` y `TestBecario` se proporcionan en ficheros independientes.