

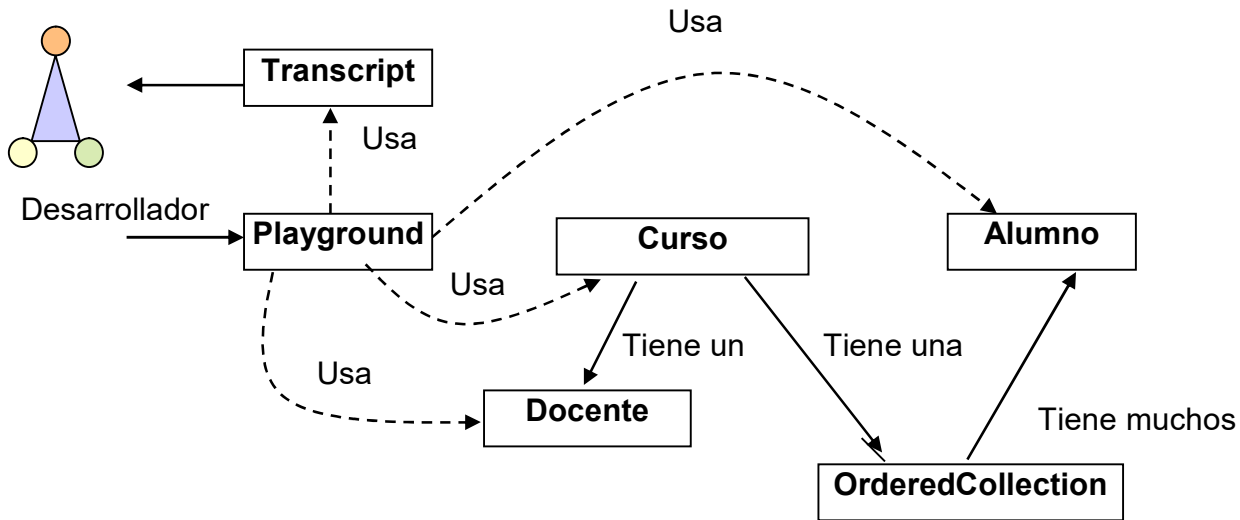
Caso de estudio – Clase 3

Utilización de clase simple y contenedora

Manejo de objetos, mensajes, bloques y colecciones

Enunciado

Se desea implementar, en forma general, la operatoria básica de gestión de alumnos de un curso, para ello se deberá implementar lo siguiente:



Parte 1

1. Crear la clase Docente.
2. Crear la clase Curso.
3. Crear la instancia Curso e invocar a sus métodos.

Parte 2

4. Crear la clase Alumno.
5. Adecuar la clase Curso para los nuevos requerimientos.
6. Crear la instancia Curso e invocar a sus métodos.

Resolución:

1. Diseño de clases

Parte 1

Docente: Atributos: legajo, apellido, cargo.
Métodos: Inicializador, metodos de acceso y modificadores, asString.

Curso: Atributos: codigo, turno y docente.
Métodos: Inicializador, metodos de acceso y modificadores, asString.

Parte 2

- Alumno: Atributos: legajo, nombre, nota1, nota2.
Métodos: Inicializador, metodos de acceso y modificadores y el método de control que retorne el promedio de nota del alumno.
- Curso: Atributos: codigo, turno, docente y una colección de alumnos.
Métodos: Inicializador, metodos de acceso y modificadores, método add que agrega un alumno a la colección de alumnos, asString y los siguientes métodos de control:
1. listadoAlumnos: retorna el listado de alumnos.
 2. cantidadInscriptos: retorna la cantidad de alumnos inscriptos en la colección.
 3. cantidadRegulares: retorna la cantidad de alumnos que tienen aprobadas sus dos notas.
 4. promedioTotal: retorna el promedio general del curso.
- Puntos adicionales**
5. primerAlumnoAprobado: retorna el primer alumno que encuentra con promedio ≥ 4 .
 6. coleccionAlumnosPromocionados: retorna una colección con los alumnos que tienen promedio > 8 .
 7. listadoOrdenado: retorna el listado de alumnos ordenados por legajo.

Implementacion en Smalltalk

Implementar Clase Docente, Alumno y Curso

Pruebas simples

Parte 1

```
|unDocente unCurso|

"creo objeto Docente para asignárselo a objeto curso"
unDocente := Docente new initialize.
unDocente legajo:12; apellido:'Pérez'; cargo:'JTP'.

"creo una instancia de Curso"
unCurso := Curso new initialize.
unCurso codigo:'2K3'; turno: 'Mañana'; docente:unDocente.

Transcript show: String cr, 'El curso: ', unCurso asString.
Transcript show: String cr, 'El docente del curso:', unCurso docente
asString.
```

```
Transcript show: String cr, 'El apellido del docente del curso:', unCurso
docente apellido asString.
```

Parte 2

```
|alu1 alu2 alu3 unDocente unCurso aux aux1 res|

"creo un objeto Docente para asignárselo al objeto curso"
unDocente := Docente new initialize.
unDocente legajo:12; apellido:'Pérez'; cargo:'JTP'.

"creo un objeto Curso"
unCurso:=Curso new initialize.
unCurso codigo:'2K3'; turno:'Mañana'; docente:unDocente.

"creo objetos Alumnos para asignárselos al curso"
alu1:=Alumno new initialize.
alu1 legajo:1;nombre: 'Lopez Maria'; nota1: 6; nota2: 8.

alu2:=Alumno new initialize.
alu2 legajo:3;nombre: 'Perez Juan'; nota1: 9; nota2: 10.

alu3:=Alumno new initialize.
alu3 legajo:2;nombre: 'Torrez Marcos'; nota1: 2; nota2: 2.

"Asigno los alumnos al objeto curso"
unCurso add: alu1.
unCurso add: alu2.
unCurso add: alu3.

Transcript show: 'Los datos del curso: ', unCurso asString ; cr.
Transcript show: (String cr, String cr).
Transcript show: String cr, 'Listado de alumnos: ', unCurso
listadoAlumnos ; cr.
Transcript show: String cr, 'Alumnos inscriptos: ', unCurso
cantidadInscriptos asString; cr.
Transcript show: String cr, 'Alumnos regulares: ', unCurso
cantidadRegulares asString;cr.
Transcript show: String cr, 'El promedio es: ', unCurso promedioTotal
asString;cr.
```

Pruebas para los puntos adicionales

```
aux:=unCurso primerAlumnoAprobado.
(aux~=nil)ifTrue:[Transcript show: String cr, ' Los datos del primer
alumno aprobado son ', aux asString.]
                    ifFalse:[Transcript show: String cr, 'No hay ningun alumno
aprobado'].
res:=unCurso coleccionAlumnosPromocionados.
```

```
"Recorro la colección retornada por el método"  
aux1:=String cr, ' Alumnos con promedio mayor a 8: '.  
res do[:x| aux1:=aux1, x asString].
```

```
Transcript show: aux1.  
Transcript show: String cr, 'Listado de alumnos ordenado: ', unCurso  
listadoOrdenado; cr.
```