



Desarrollo de Software Basado en Componentes y Servicios

“Ejercicios OCL y UML”

María Victoria Santiago Acalá

76656695H

Contenidos

<i>Ejercicio OCL</i>	3.
<i>Ejercicio UML</i>	4

Ejercicio OCL

1. Especificar completamente la operación makeReservation (...) con OCL; utilizando los operadores: exists, select y asSequence->first.

El OCL es el que se muestra en la siguiente sección:

OCL

```
IEstudianteMgt::makeReservation (in enrol: DetallesEnrolamiento, in est: IdEstudiante, out
IdEnrolamiento: int): boolean
pre:
-- El ID del alumno no debe de ser nulo
IdEstudiante != null
-- El acto de enrolamiento en un curso en una universidad no es nulo
and enrol != null
-- La universidad tiene que existir
and self.universidades->exists(uni: Universidad | uni.id = enrol.universidad)
post:

-- Ahora para obtener una universidad por medio de su id
Let universidad : Universidad = self.universidades->select(uni: Universidad | uni.id = enrol.universidad)-
>asSequence()->first()

in
-- Buscamos si tenemos curso con los detalles que queremos en la universidad
if universidad.cursos->exists(c : Facultad | c.curso.disponibilidad(enrol.fecha) and c.curso = enrol.curso)
then

    -- Ahora el numero de matriculados se tiene que incrementar en 1 y si hay ya un enrolamiento
    se devuelve la referencia de enrolamiento del alumno que está solicitandolo (IdEstudiante) en la fecha
    dicha (enrol.fecha), además va a realizar la asociación al curso (enrol.curso) que desea el alumno
    (IdEstudiante) y finalmente el resultado de lo que se hace es verdadero
    self.enrolamientos->size() = self.enrolamientos@pre->size()+1 and
    self.enrolamientos->includes(c: Enrolamiento | c.IdEnrolamiento = enrol.IdEnrolamiento and
c.fecha = enrol.fecha and c.claimed = true and c.estudiante.id = IdEstudiante and
c.curso.disponibilidad(enrol.fecha) = true and c.curso = enrol.curso) and
    result = true
else
    result = false

end if

////////Comentarios
DetallesEnrolamiento:
Universidad: idUniversidad

Se puede añadir al invariante asociada quedando asi:
if c.allocation->notEmpty() then
    c.claimed = true and
    c.curso.disponibilidad(c.fecha) = true
else
    c.claimed = false
endif
```

Ejercicio UML

2. Realizar los siguientes diagramas UML.
 - a) Factorizar los elementos comunes de los modelos de información de las interfaces IMakeReservation y ITakeUpReservation y ubicarlos en una nueva interfaz IReservationSystem de las que las 2 interfaces aludidas heredan.
 - b) Elaborar los diagramas de clases de las interfaces IReservationSystem y rehacer el de ImakeReservation.

Quedaría de la siguiente forma:

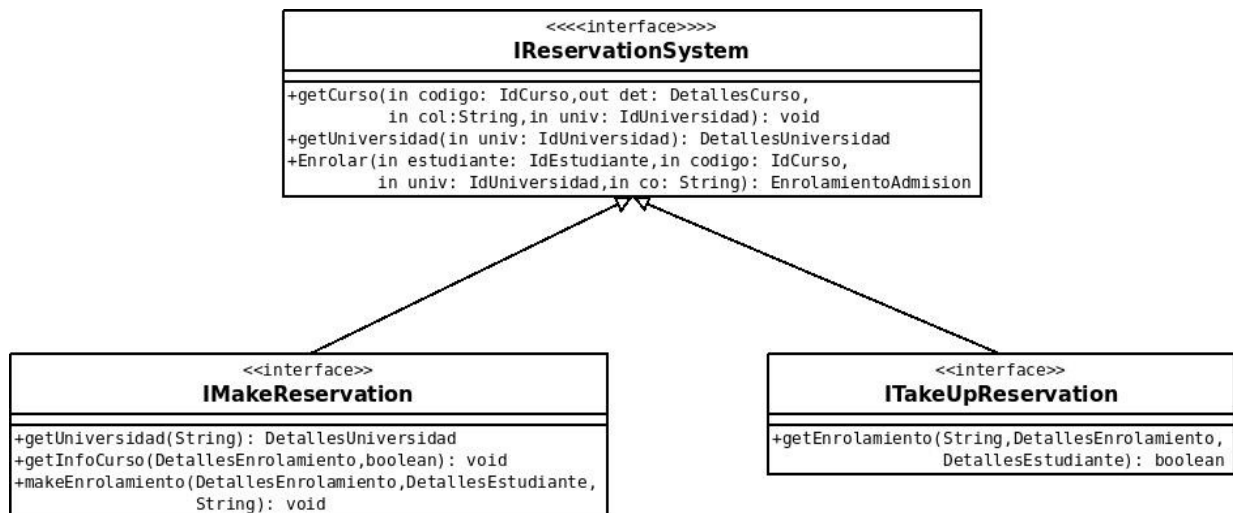


Diagrama 1

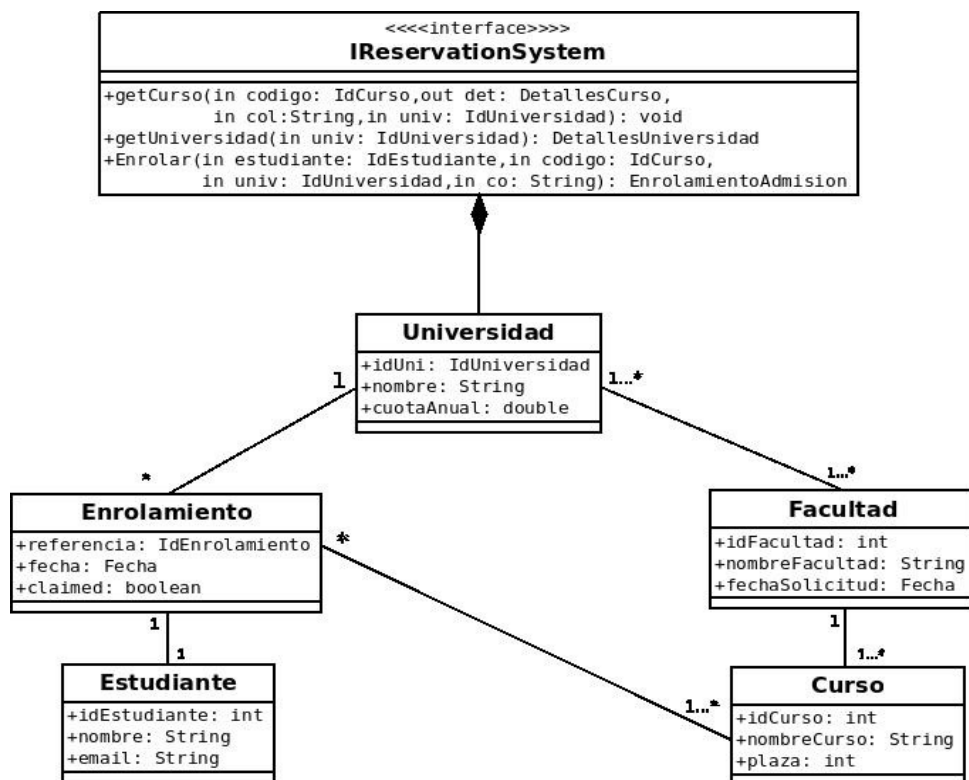


Diagrama 2

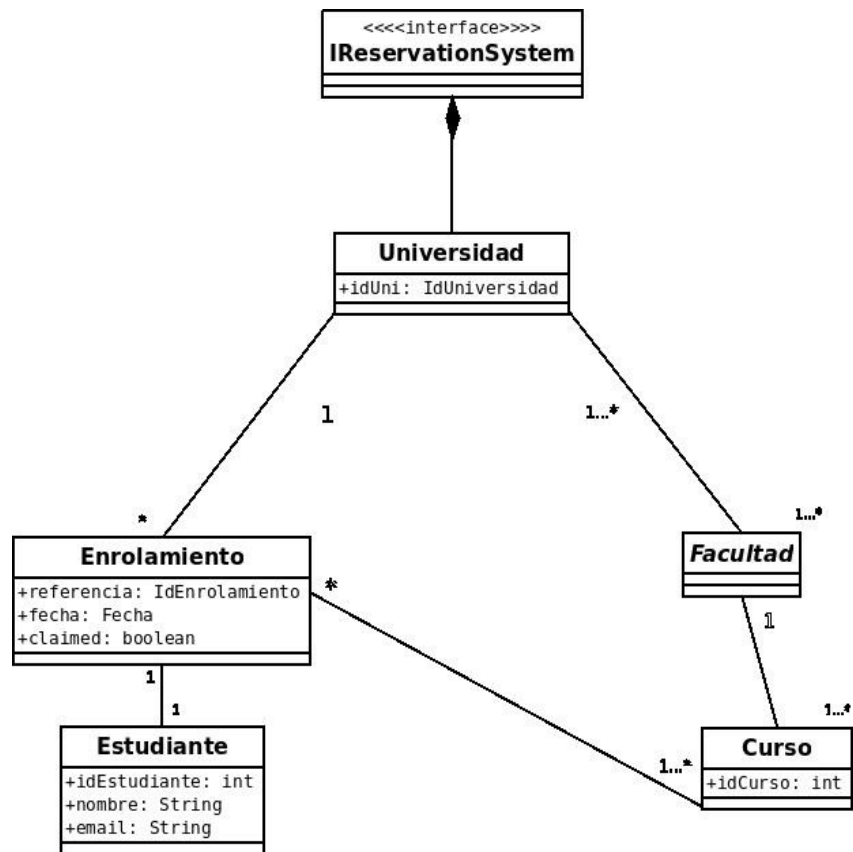


Diagrama 3

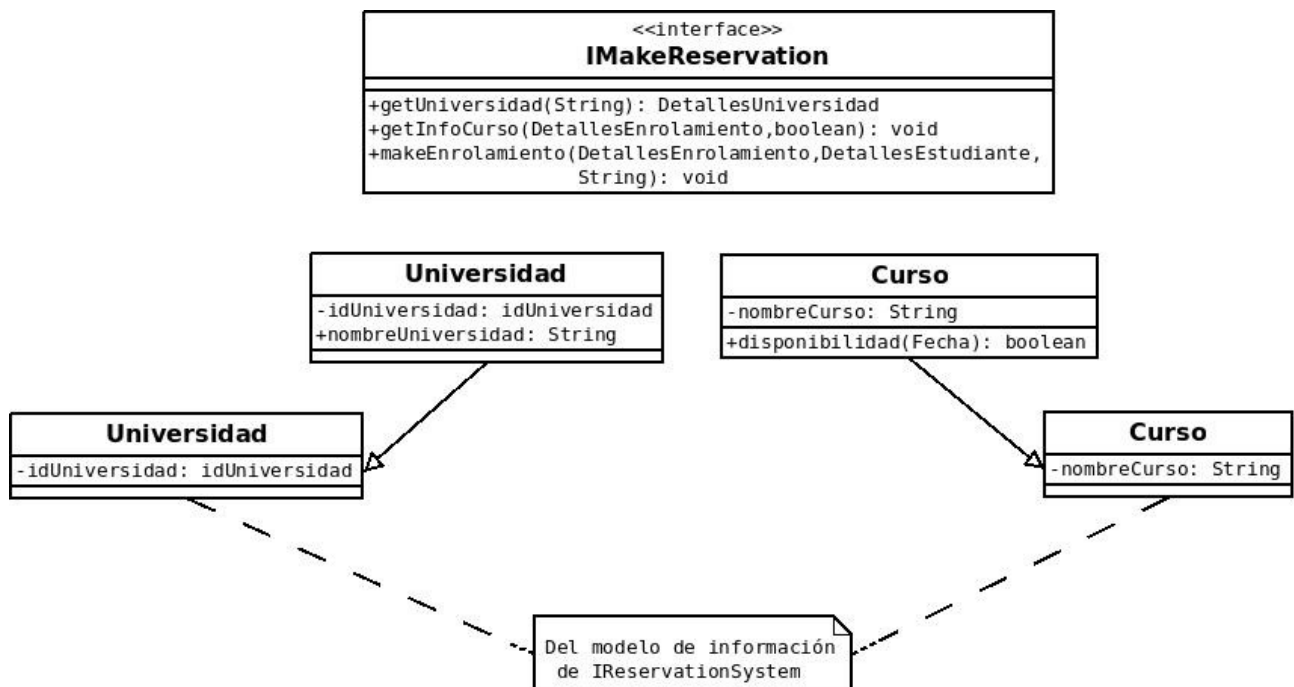


Diagrama 4