

PROYECTO DE ANÁLISIS Y DISEÑO DE SOFTWARE

-Práctica 2-

SAPERE AUDE

ÍNDICE

1.Diagrama de clases

2.Diagramas de estados

3.Diagrama de secuencia

4.Matriz de trazabilidad de requisitos

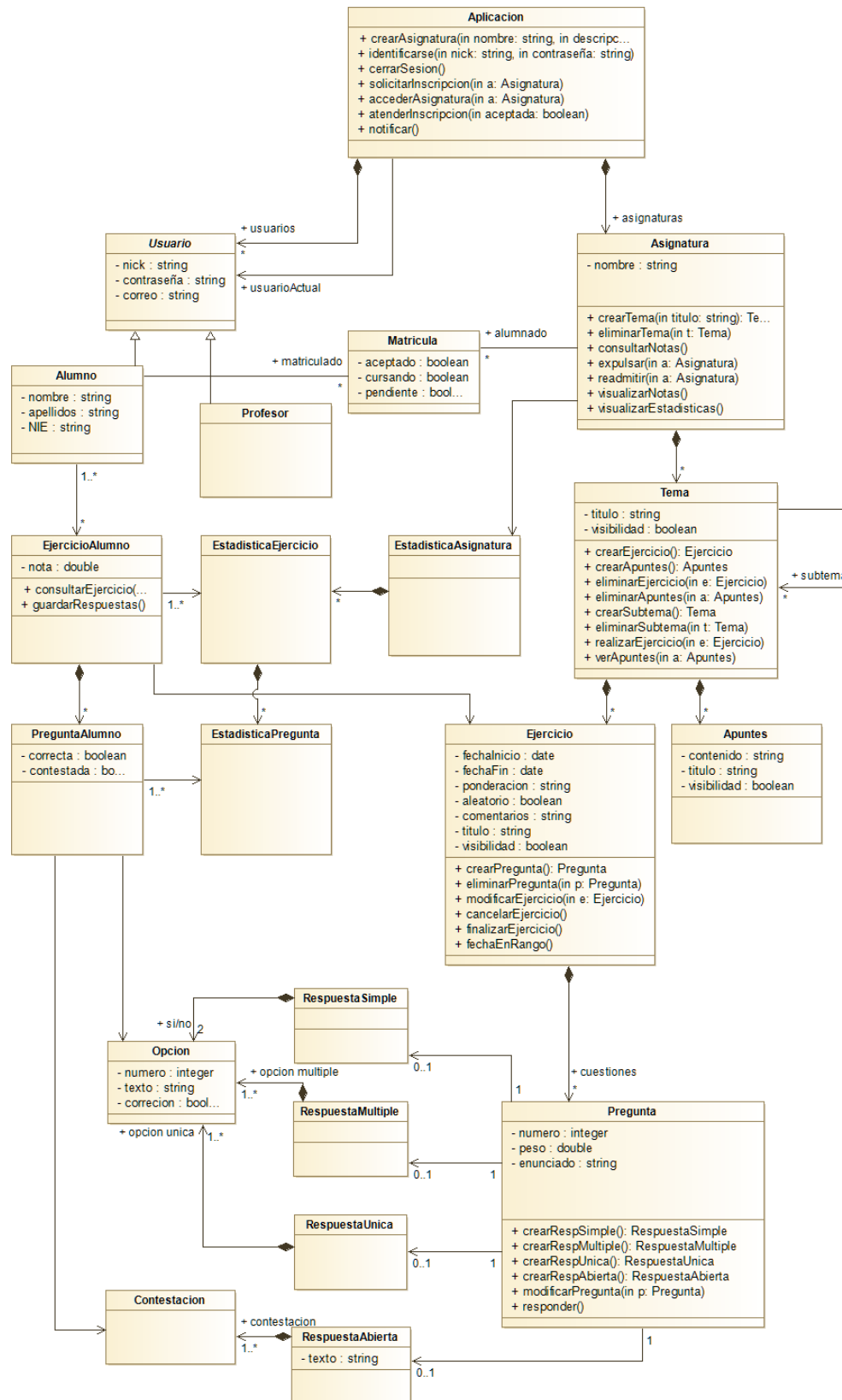
INTRODUCCIÓN

Esta segunda práctica de Proyecto de Análisis y Diseño de Software, con fecha de entrega el martes 7 de marzo, consiste en la culminación de la parte correspondiente al diseño en lo que sería todo nuestro proyecto. Concretamente el trabajo ha consistido en la realización de una serie de diagramas en los que hemos estructurado, tras un profundo análisis, la información capturada en la práctica anterior a fin de facilitar su posterior implementación en Java y tener una idea clara del funcionamiento de la aplicación exigida.

Para la realización de los diagramas mencionados hemos utilizado las siguientes herramientas:

- Modelio: tanto para el diagrama de clases como los de estados.
- Cacoo: para desarrollar los diagramas de secuencia.

1. DIAGRAMA DE CLASES



Comentarios al diagrama

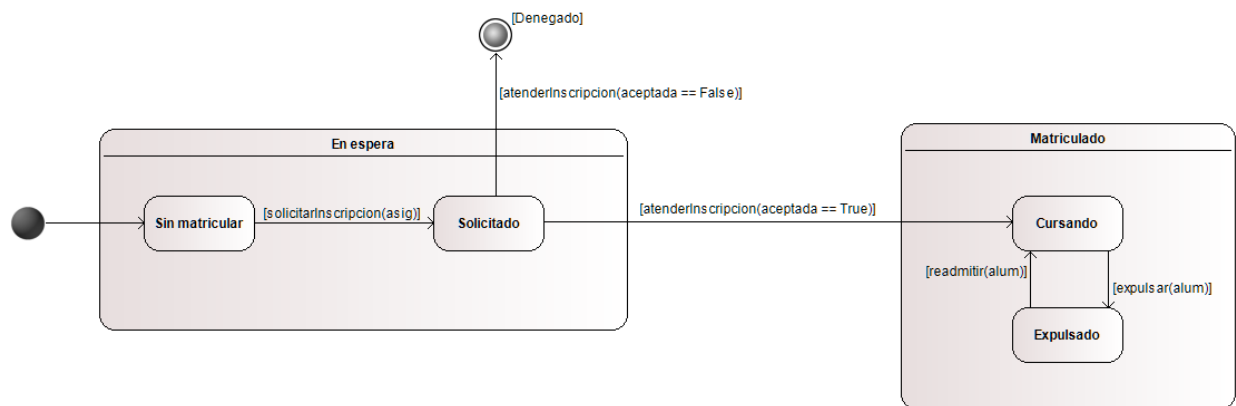
El diagrama mostrado en la página anterior se corresponde con el de clases y engloba a grandes rasgos cual va a ser la estructura general a seguir para implementar nuestra aplicación a la hora de utilizar un lenguaje de programación para ello.

Podemos observar que su clase principal es Aplicacion, de la cual parten el resto de ellas. Se encuentra formada por una serie de asignaturas y usuarios. Estos últimos están encapsulados en la única clase abstracta que hemos decidido emplear y se dividen en Alumno y Profesor. Contemplamos la posibilidad de crear otra clase abstracta llamada recursos que generalizase el comportamiento común de las clases Tema, Ejercicio y Apuntes; no obstante, debido a que los temas contienen a los otros dos tipos de recursos e incluso a sí mismos con subtemas que se pueden anidar decidimos emplear la clase Tema como una de jerarquía superior que pudiera estar formada por cualquier recurso. Por otro lado, cada ejercicio estará formado por una serie de preguntas, que estas a su vez tendrán un tipo de respuesta asociado.

Por último, y sin duda la parte más delicada del diseño, nos encontramos con una serie de clases que permiten la corrección, cálculo de notas y estadísticas acerca de los ejercicios. EjercicioAlumno y PreguntaAlumno (la primera clase se encuentra formada por elementos de la segunda) son una especie de registros que almacenan las contestaciones de cada alumno con el fin de poder compararlas con las respuestas correctas a la hora de corregir. Siguiendo, las clases cuyos nombres comienzan por Estadistica se encargan de recopilar la información y establecer las relaciones necesarias con el objetivo de presentar a los usuarios la información exigida en los requisitos.

Cabe mencionar que, debido a haber utilizado Modelio para la realización del diagrama, los parámetros de entrada de los métodos llevan un “in” delante el cual se puede obviar y además solo se puede escribir el tipo “String” con la primera letra minúscula. Los getters, setters y constructores de cada clase no están representados en el diagrama pero se asume su existencia.

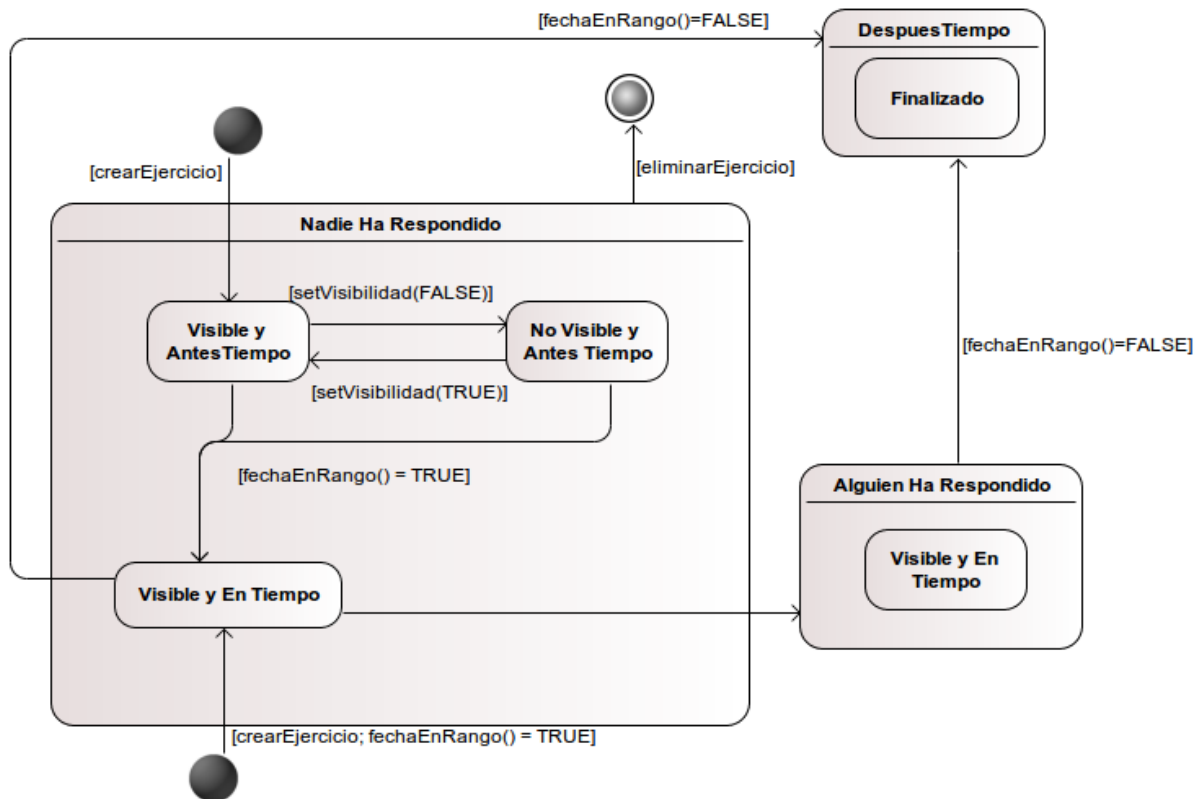
2. DIAGRAMAS DE ESTADOS



Comentarios sobre el diagrama de estados de alumno

Este primer diagrama de estados muestra las posibles situaciones en la que un alumno perteneciente a nuestra aplicación puede encontrarse con respecto a su pertenencia a una determinada asignatura. El estado inicial (Sin matricular) se trata de aquel en el que el estudiante nos ha solicitado en ningún momento el acceso a dicha asignatura. Una vez este realiza la solicitud, pasa el estado 'Solicitado', en el cual permanecerá hasta que un profesor atienda este proceso de inscripción. En caso de que sea denegado, llegamos al estado final; mientras que si lo aceptan entonces el estudiante pasa al estado de matriculado. En concreto siempre comenzará cursando el curso; no obstante, un profesor puede expulsarle de la asignatura, dejando al alumno en estado de 'Expulsado' hasta que no vuelva a ser readmitido.

Los atributos de la clase Matricula están diseñados como unas variables booleanas las cuales se encargan de definir en cuál de los estados presentados en el diagrama se encuentra un estudiante en relación a la asignatura en la cual se encuentra matriculado. Debemos mencionar que en el diagrama tan solo se muestran las funciones de alto nivel; sin embargo estas deben llamar a los setters de la clase Matricula para realizar su labor.



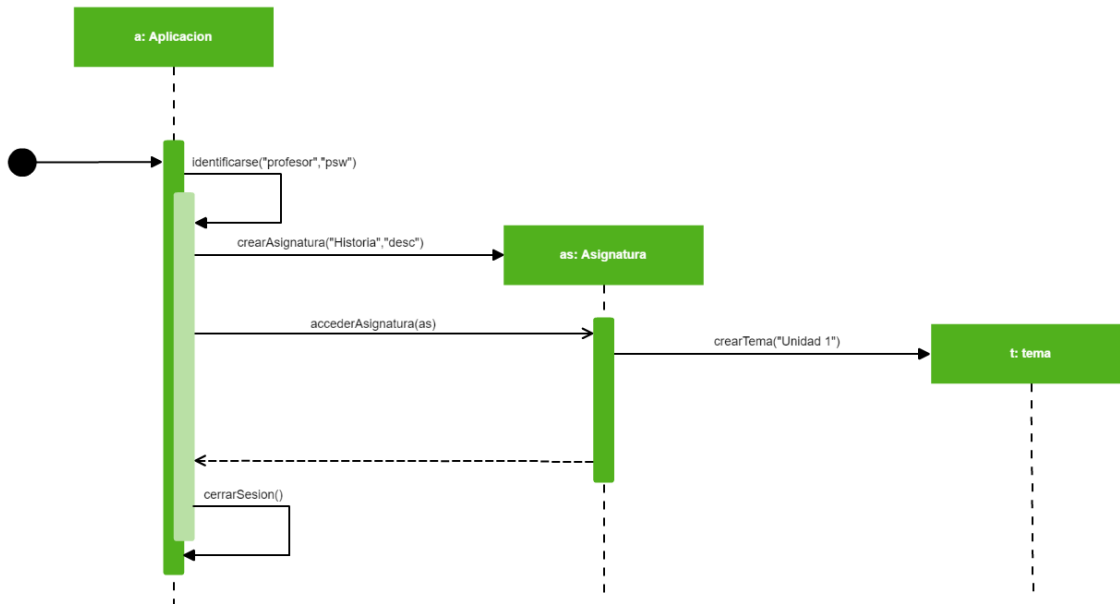
Comentarios sobre el diagrama de estados de ejercicio

En este segundo diagrama representamos aquellos estados en los que un ejercicio es capaz de encontrarse. Su situación depende de tres factores fundamentales: el primero se trata del momento en el que nos encontramos, ya que podemos tener un ejercicio previo a su fecha de inicio, puede encontrarse abierto, o ya puede haber caducado su período activo con lo que estaríamos en una fecha posterior a su final.

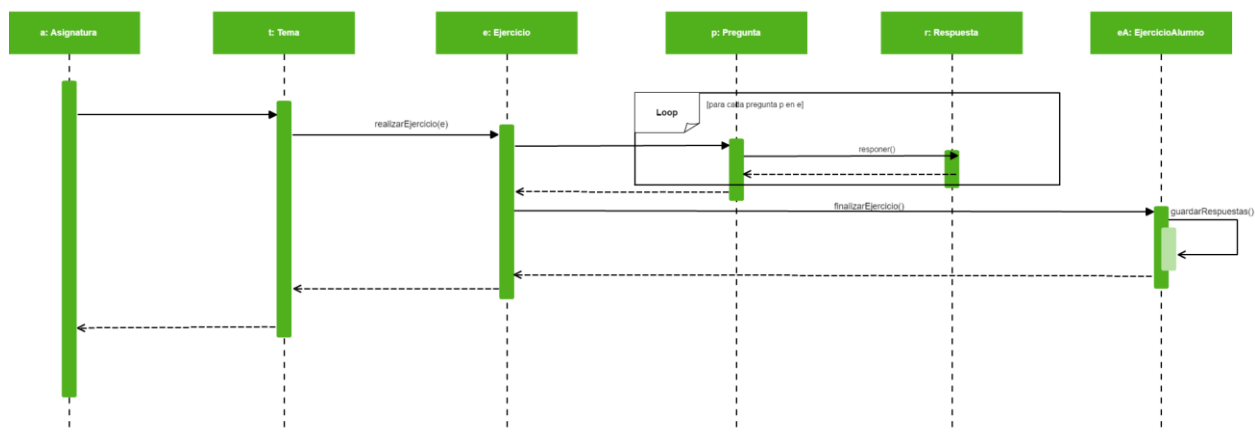
Por otro lado, los profesores pueden modificar la visibilidad de cada ejercicio de forma que se encuentren en el estado de no visible o visible. Cabe destacar aquí que asumimos que siempre que se cree un nuevo ejercicio, por defecto este estará visible. Además, si el ejercicio está en o después de tiempo no puede ocultarse.

En último lugar, otro factor que determina el estado propio de un ejercicio es si ha sido respondido por algún alumno o no, pues esto es un criterio para decidir si es modificable o no. A modo de resumen y en consecuencia de todo lo anterior nos encontramos con los cinco estados posibles mostrados en el diagrama.

3. DIAGRAMAS DE SECUENCIA



La imagen superior representa el diagrama de secuencia correspondiente con la acción “crear asignatura y un primer tema” por parte de un profesor. Aquí mostramos el proceso entero, desde que este se identifica hasta que cierra su sesión. Simplemente se tendrá que llamar a las funciones de creación de asignatura para añadir una nueva clase Asignatura al sistema, de acceso a asignatura y finalmente, en ella, de creación de tema para que la asignatura no se encuentre vacía.



En contraste, en este segundo diagrama no se aprecian creaciones de nuevas clases pues suponemos que ya tenemos el sistema construido con anterioridad y no incluimos la clase Aplicación, donde debería iniciarse y finalizar el diagrama, por simplificar. Aquí se representa el caso de uso asociado a la “realización de un ejercicio” por parte de un alumno. A grandes rasgos, este se encuentra en una determinada asignatura, selecciona un ejercicio perteneciente a cierto tema y comienza a responder las preguntas. Cuando termina con todas, pulsa finalizar por lo que se guardan sus respuestas en el sistema.

4. DIAGRAMAS DE SECUENCIA

Requisitos	Elementos Funcionales						
	Aplicacion	Asignatura	Tema	Ejercicio	Apuntes	Pregunta	EjercicioAlumno
Requisito 2.1.1							
Requisito 2.1.1.1	identificarse						
Requisito 2.1.1.2	solicitarInscripcion						
Requisito 2.1.1.3			verApuntes				
Requisito 2.1.1.4			realizarEjercicio				
Requisito 2.1.1.5			realizarEjercicio	cancelarEjercicio finalizarEjercicio			
Requisito 2.1.1.6							consultarEjercicio
Requisito 2.1.1.7							consultarEjercicio
Requisito 2.1.1.8		consultarNotas					
Requisito 2.1.2							
Requisito 2.1.2.1	identificarse						
Requisito 2.1.2.2	crearAsignatura						
Requisito 2.1.2.3	atenderInscripcion						
Requisito 2.1.2.4		expulsar					
Requisito 2.1.2.5		readmitir					
Requisito 2.1.2.6		crearTema	crearSubtema crearApuntes crearSubtema				
Requisito 2.1.2.7			setVisibilidad	setVisibilidad	setVisibilidad		
Requisito 2.1.2.8				setPonderacion			
Requisito 2.1.2.9			crearEjercicio	crearPregunta		crearRespSimple crearRespMultiple crearRespAbierta crearRespUnica	
Requisito 2.1.2.10				setAleatorio			
Requisito 2.1.2.11				modificarEjercicio			
Requisito 2.1.2.12				modificarEjercicio		modificarPregunta	
Requisito 2.1.2.13		visualizarNotas					
Requisito 2.1.2.14		visualizarEstadisticas					
Requisito 2.1.3							
Requisito 2.1.3.1	notificar						

En esta matriz de trazabilidad se listan todos los requisitos de la primera entrega ya corregidos, para una mayor facilidad en la corrección en la hoja siguiente se adjunta un listado de los mismos.

En la matriz, hemos decidido por cuestiones de espacio, modificar un poco la plantilla dada. En esta matriz al igual que en la dada se listan todos los requisitos a la izquierda, marcándose con un color más oscuro aquellos que simplemente expresan qué tipo de usuario interactúa con la aplicación (profesor, alumno, aplicación). Por otro lado, arriba hemos listado las clases necesarias para cumplimentar todos los requisitos y sus métodos los hemos ido escribiendo en los espacios de su correspondiente clase (en la que se implementa el método) y requisito que usa dicha función para ser completado.

A continuación, mostramos una tabla que almacena los requisitos capturados en la práctica anterior para poder consultarlos a la hora de analizar la matriz de trazabilidad:

<u>Numeracion</u>	<u>Contenido</u>
Requisito 2.1.1	ESTUDIANTE
Requisito 2.1.1.1	Acceder a la aplicacion mediante su nombre de usuario y contraseña únicos para cada alumno
Requisito 2.1.1.2	Solicitar admisión en una asignatura
Requisito 2.1.1.3	Acceder a los contenidos visibles de las asignaturas que está cursando.
Requisito 2.1.1.4	Acceder a un ejercicio antes de la fecha de inicio estipulada. Se observará el título del ejercicio y sus fechas de inicio y fin.
Requisito 2.1.1.5	Acceder a un ejercicio dentro de la fecha estipulada para realizarlo permitiendo al alumno enviarlo su intento entero o cancelarlo de forma que pueda repetirlo más adelante.
Requisito 2.1.1.6	Acceder a su ejercicio una vez realizado el mismo pero sin que haya terminado su fecha de fin para permitir ver cuáles fueron sus respuestas.
Requisito 2.1.1.7	Acceder al ejercicio tras haberse cumplido la fecha final del mismo, permitiendo que el alumno ahora observe la solución correcta del ejercicio, su respuesta enviada y la nota total sobre 10 del ejercicio.
Requisito 2.1.1.8	Acceder a la nota que tengan acumulada en cada asignatura con la ponderación indicada.
Requisito 2.1.2	PROFESOR
Requisito 2.1.2.1	Acceder a la aplicacion mediante el login con un mismo usuario y contraseña para todos los profesores.
Requisito 2.1.2.2	Crear nombre de las asignaturas (no borrarlas)
Requisito 2.1.2.3	Aceptar o denegar acceso a alumnos en una determinada asignatura.
Requisito 2.1.2.4	Expulsar a un alumno de alguna asignatura.
Requisito 2.1.2.5	Readmitir al alumno en una asignatura de la que fue expulsado.
Requisito 2.1.2.6	Subir apuntes así como organizarlos en temas dentro de una asignatura
Requisito 2.1.2.7	Hacer visibles u ocultar ciertos contenidos de las asignaturas.
Requisito 2.1.2.8	Crear y modificar las ponderaciones de cada uno de los ejercicios respecto a la nota final de la asignatura.
Requisito 2.1.2.9	Crear un ejercicio con la elección de opciones propias (titulo y orden de preguntas fijo o aleatorio por ejemplo), alteración de los diversos tipos de pregunta (RS,RM,RU,RA) y de sus puntuaciones existiendo una posibilidad de puntuacion por defecto para todas las preguntas (solo se indicarán las puntuaciones de aquellas que ponderen distinto)
Requisito 2.1.2.10	Configurar orden de las respuestas posibles a una pregunta(fijo o aleatorio).
Requisito 2.1.2.11	Modificar ejercicios subidos siempre que no se hayan iniciado o no los haya realizado ningún alumno.
Requisito 2.1.2.12	Modificar la fecha de fin de un ejercicio cuando ya haya sido realizado por al menos un alumno (no podrá modificar otros campos del ejercicio como fecha de inicio o las preguntas de las que se compone). La nueva fecha de fin será posterior a la fecha de la última respuesta.
Requisito 2.1.2.13	Consultar notas por estudiante (si hubiese algún ejercicio sin contestar pero dentro de plazo el profesor vería la media sin contar ese ejercicio).
Requisito 2.1.2.14	Ver porcentaje de aciertos, fallos y no contestado por pregunta de cada uno de los ejercicios.
Requisito 2.1.3	APLICACION
Requisito 2.1.3.1	Notificar vía email a los alumnos de un curso cuando se hagan visibles apuntes y/o ejercicios, la fecha de inicio o fin de los ejercicios cambien, se produzca una respuesta a su solicitud de inscripción, sea expulsado o readmitido en la asignatura.