



Departamento de Informática
Universidad Técnica Federico Santa María



Pauta Entregable II (Informe + Modelos + Software)
Análisis y Diseño de Software / Fundamentos de Ingeniería de Software

Integrantes:

Nombre	Email	Teléfono
Giovanni Aravena	giovanni.aravena.13	+56949319142
Rodrigo Lagos	rodrigo.lagos.13	+56967188391
Fernando Llorens	fernando.llorens.13	+56952378157

Entregable 2

1- Listado de requerimientos

Requerimientos funcionales:

1. El sistema debe proveer al usuario un inicio de sesión (*estudiante, coordinador y profesor*).
2. El sistema debe proveer al usuario (*estudiante*) de una opción de registro.
3. El sistema debe proveer al usuario (*profesor*) de una opción de registro.
4. El sistema debe proveer al usuario (*estudiante*) de una encuesta inicial.
5. El sistema debe ser capaz de determinar el tipo de aprendizaje a raíz de la encuesta contestada por el usuario (*estudiante*).
6. El sistema debe ser capaz de proveer al usuario (*Coordinador*) opciones para gestionar usuarios.
7. El sistema debe ser capaz de proveer al usuario (*Coordinador*) opciones para gestionar cursos.
8. El sistema debe proveer al usuario (*Coordinador*) opciones para modificar la apariencia de un curso.
9. El sistema debe proveer al usuario (*profesor*) opciones para gestionar material a un curso.
10. El sistema debe proveer al usuario (*profesor*) un modelo de registro inicial.
11. El sistema debe proveer al usuario (*profesor*) opciones para validar a un nuevo profesor
12. El sistema debe ser capaz de cambiar la interfaz de un curso acorde al tipo de usuario (*estudiante*).
13. El sistema debe proveer al usuario (*estudiante*) opciones para gestionar sus cursos.
14. El sistema debe proveer al usuario (*estudiante*) opciones para realizar feedback.
15. El sistema debe proveer opciones para visualización de feedback para el usuario *Coordinador* y el usuario *alumno*.

Entregable 2

Id requerimiento	Requerimiento	Obligatoriedad
IN_USR	El sistema debe proveer al usuario un inicio de sesión (<i>estudiante, coordinador y profesor</i>).	Obligatorio
REG_1	El sistema debe proveer al usuario (<i>estudiante</i>) de una opción de registro.	Obligatorio
REG_2	El sistema debe proveer al usuario (<i>profesor</i>) de una opción de registro.	Deseable
QU	El sistema debe proveer al usuario (<i>estudiante</i>) de una encuesta inicial.	Obligatorio
ST_TYP	El sistema debe ser capaz de determinar el tipo de aprendizaje a raíz de la encuesta contestada por el usuario (<i>estudiante</i>).	Obligatorio
CRUD_USR	El sistema debe ser capaz de proveer al usuario (<i>Coordinador</i>) opciones para gestionar usuarios.	Obligatorio
CRUD_CUR	El sistema debe ser capaz de proveer al usuario (<i>Coordinador</i>) opciones para gestionar cursos.	Obligatorio
CUR_VIEW	El sistema debe proveer al usuario (<i>Coordinador</i>) opciones para modificar la apariencia de un curso.	Obligatorio
CRUD_MAT	El sistema debe proveer al usuario (<i>profesor</i>)	Obligatorio

Entregable 2

	opciones para gestionar material a un curso.	
REG_PROF	El sistema debe proveer al usuario (<i>profesor</i>) un modelo de registro inicial.	Deseable
CHECK_PROF	El sistema debe proveer al usuario (<i>profesor</i>) opciones para validar a un nuevo profesor.	Deseable
TYPE_VIEW	El sistema debe ser capaz de cambiar la interfaz de un curso acorde al tipo de usuario (<i>estudiante</i>).	Obligatorio
CUR_GEST_EST	El sistema debe proveer al usuario (<i>estudiante</i>) opciones para gestionar sus cursos.	Obligatorio
FEED_USR	El sistema debe proveer al usuario (<i>estudiante</i>) opciones para realizar feedback.	Obligatorio
VIEW_FEED	El sistema debe proveer opciones para visualización de feedback para el usuario <i>Coordinador</i> y el usuario <i>alumno</i> .	Obligatorio

Entregable 2

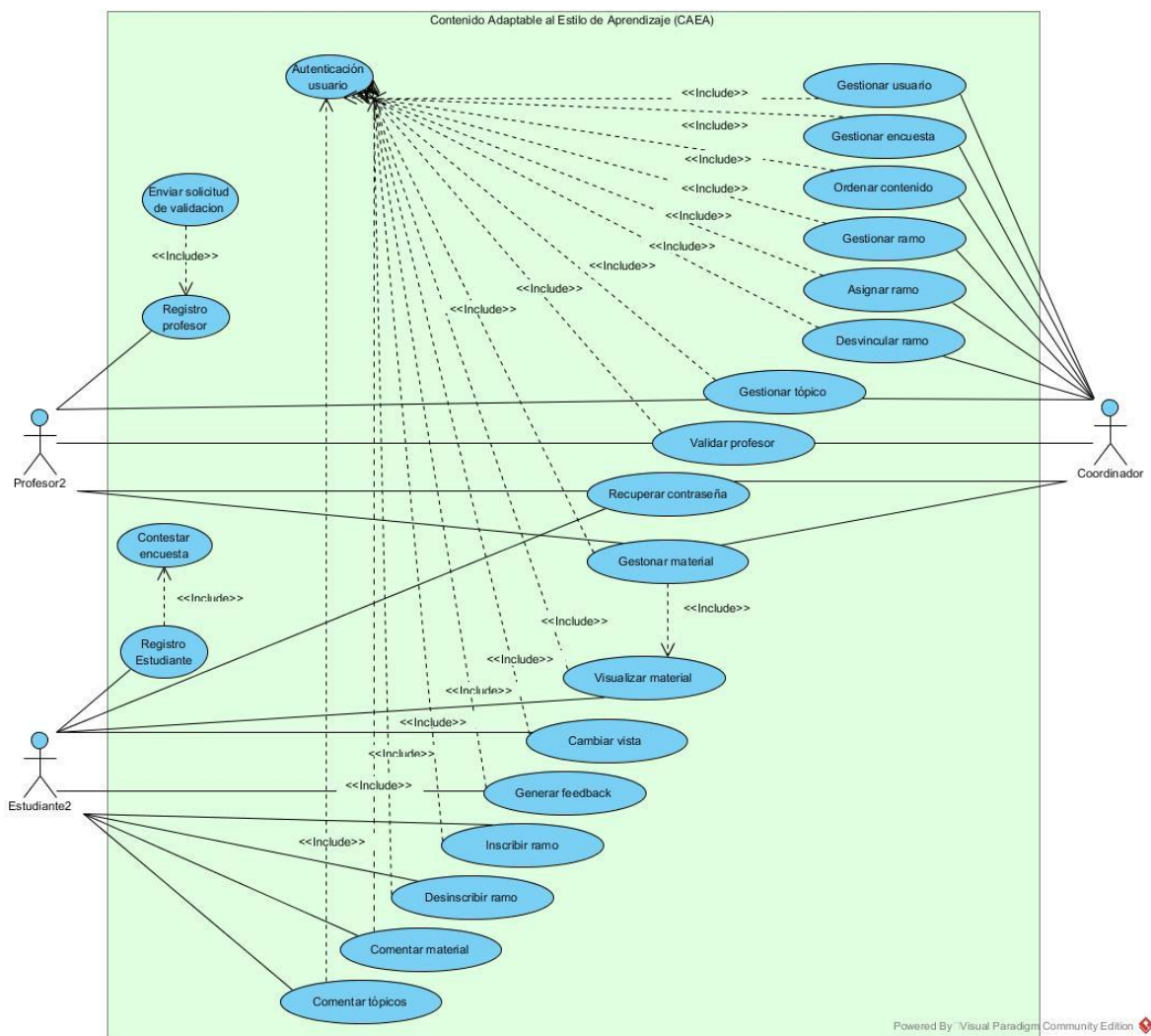
Requerimientos no funcionales:

1. El usuario (*estudiante*) no debe saber cuál es su tipo de aprendizaje.
2. No hay diferenciación en el contenido del material, pero sí en la visualización de este (vista).
3. El usuario (*estudiante*) tiene la posibilidad de ver los otros tres estilos de aprendizaje (vista).
4. Profesor y Coordinador comparten la misma vista sobre el sistema, no así las funciones.

Id requerimiento	Requerimiento no funcional	Lista de FRs asociados
NOT_TYPE	El usuario (<i>estudiante</i>) no debe saber cuál es su tipo de aprendizaje.	ST_TYP, TYPE_VIEW.
SAME_MAT	No hay diferenciación en el contenido del material, pero sí en la visualización de este (vista).	TYPE_VIEW, CRUD_MAT.
OTHER_VIEW	El usuario (<i>estudiante</i>) tiene la posibilidad de ver los otros tres estilos de aprendizaje (vista).	TYPE_VIEW, ST_TYP
DIFF_US_SAME_VIEW	Profesor y Coordinador comparten la misma vista sobre el sistema, no así las funciones.	TYPE_VIEW.

Entregable 2

2- Casos de uso



Entregable 2

Casos de uso breves:

1. *Generar feedback (FEED_USR)*: el **estudiante** desea dar su opinión sobre la visibilidad de contenidos. El **estudiante** mueve los bloques de información y salva los datos. Los datos son recibidos por el **sistema**. El **sistema** envía esta información al **Coordinador**.
2. *Ordenar contenido (CUR_VIEW)*: el **Coordinador** desea cambiar bloques de información. El **Coordinador** se sitúa sobre el perfil a modificar. El **Coordinador** mueve los bloques de información y salva los datos. El **sistema** recibe estos datos y los guarda.
3. *Subir material (CRUD_MAT)*: el **profesor** desea subir un nuevo material a la plataforma. El **profesor** selecciona el curso y tópico buscado. El **profesor** selecciona subir material. El **profesor** selecciona la visibilidad de este material y guarda los datos. El **sistema** guarda los datos y actualiza el curso y tópico en cuestión (*esto es una parte del caso de uso Gestionar material - CRUD material*).
4. *Asignar ramo (CRUD_USR)*: el **Coordinador** desea asignar un curso a un **profesor**. El **Coordinador** busca el curso en el **sistema** y tipea el nombre del **profesor** a asignar. El **Coordinador** guarda los cambios. El **sistema** recibe los datos correctamente.
5. *Actualizar usuario (CRUD_USR)*: el **Coordinador** desea cambiar cierta información de un usuario. El **Coordinador** busca a través del **sistema** al usuario en cuestión. Selecciona al usuario y hace los cambios necesarios. Guarda los datos y el **sistema** acepta los cambios (*esto es una parte del caso de uso Gestionar usuario - CRUD usuario*).
6. *Cambiar vista (TYPE_VIEW)*: el **estudiante** desea probar otra visualización de contenido. El **estudiante** selecciona la opción para cambiar de vista. El **sistema** muestra al **estudiante** una nueva visualización. El **estudiante** decide cuál usar y guarda los datos. El **sistema** acepta la solicitud.
7. *Crear tópico (CRUD_CUR)*: el **Coordinador** desea añadir temas a un curso. El **Coordinador** selecciona el curso en cuestión y elige crear tópico. El **sistema** recibe la información necesaria. El **sistema** actualiza el curso (*esto es una parte del caso de uso Gestionar tópico - CRUD tópico*).
8. *Contestar encuesta (ST_TYP)*: el **estudiante** debe ser clasificado según su estilo de aprendizaje por el **sistema**. El **estudiante** contesta la encuesta inicial de registro y envía los datos. El **sistema** recibe los datos y clasifica al **estudiante**.

Entregable 2

Casos de uso extendidos:

Caso de uso: Contestar encuesta.
Propósito: Definir la categoría del estudiante.
Actor(es): Estudiante
Precondiciones: 1.- Estar logueado en el sistema. 2.- Estar identificado como estudiante.
Postcondiciones: Estudiante es clasificado según el estilo de aprendizaje.

Secuencia contestar encuesta

Estudiante	Sistema
1. Estudiante completa los campos necesarios de la encuesta y presiona enviar.	
	2. El sistema recibe los datos y verifica si son correctos.
	3. En caso de ser correctos, el sistema redirige al estudiante a la página principal de CAEA.

Secuencia alternativa contestar encuesta

	3.1 En caso de ser incorrectos, el sistema envía un mensaje de error y el proceso regresa a 1.
--	--

Entregable 2

Caso de uso: Generar feedback.
Propósito: Definir la categoría del estudiante.
Actor(es): Estudiante
Precondiciones: 1.- Estar logueado en el sistema. 2.- Estar identificado como estudiante.
Postcondiciones: Ninguna.

Secuencia generar feedback.

Estudiante	Sistema
1. Estudiante selecciona un tópico y elige la opción "proponer diseño".	
	2. El sistema muestra los bloques de información modificables (posición).
3. El estudiante permuta los bloques de información.	
4. El estudiante selecciona la opción "enviar sugerencia".	
	5. El sistema recibe los datos, en caso de ser correctos, redirige al estudiante a la página del curso sin modificaciones.

Secuencia alternativa contestar feedback

	5.1 El sistema recibe los datos, pero no puede enviar el feedback. El sistema lanza un mensaje de error y el proceso regresa al punto 2.
--	--

Entregable 2

Caso de uso: Ordenar contenido.
Propósito: Cambiar orden de bloques de información.
Actor(es): Coordinador
Precondiciones: 1.- Estar logueado en el sistema. 2.- Estar identificado como Coordinador.
Postcondiciones: Nueva visualización de datos para uno o más tipos de estudiantes.

Secuencia ordenar contenido.

Coordinador	Sistema
1. Coordinador selecciona un curso y tópico.	
	2. El sistema muestra la vista actual de los 4 tipos sobre el curso.
3. El Coordinador elige un tipo y selecciona la opción "cambiar orden".	
	4. El sistema muestra los bloques de información modificables (posición).
5. El Coordinador permuta los bloques de información.	
6. El Coordinador selecciona la opción "guardar cambios".	
	7. El sistema recibe los datos. Si todo está correcto, mostrará la nueva visualización de bloques.
	7.1. Si existe error, el sistema lanzará un mensaje de fracaso y el proceso regresa a 5.

Entregable 2

Caso de uso: Actualizar usuario
Propósito: Coordinador desea cambiar algún dato sobre algún usuario
Actor(es): Coordinador
Precondiciones: 1.- Estar logueado en el sistema. 2.- Estar identificado como Coordinador.
Postcondiciones: Actualización correcta de datos del usuario

Secuencia ver usuario

Coordinador	Sistema
1. El Coordinador selecciona la pestaña de gestión de usuarios.	
	2. El sistema redirige al Coordinador a la página correspondiente.
	3. El sistema despliega un cuadro de búsqueda.
4. Coordinador ingresa el nombre del usuario y selecciona buscar.	
	5. El sistema verifica la búsqueda
	6. Si existe coincidencia, el sistema muestra los usuarios posibles.
7. Coordinador selecciona al usuario deseado.	
	8. El sistema despliega un sub-menú de opciones.
9. El Coordinador selecciona "Actualizar usuario".	
	10. El sistema despliega los detalles sobre el usuario.
11. El Coordinador selecciona los campos a	

Entregable 2

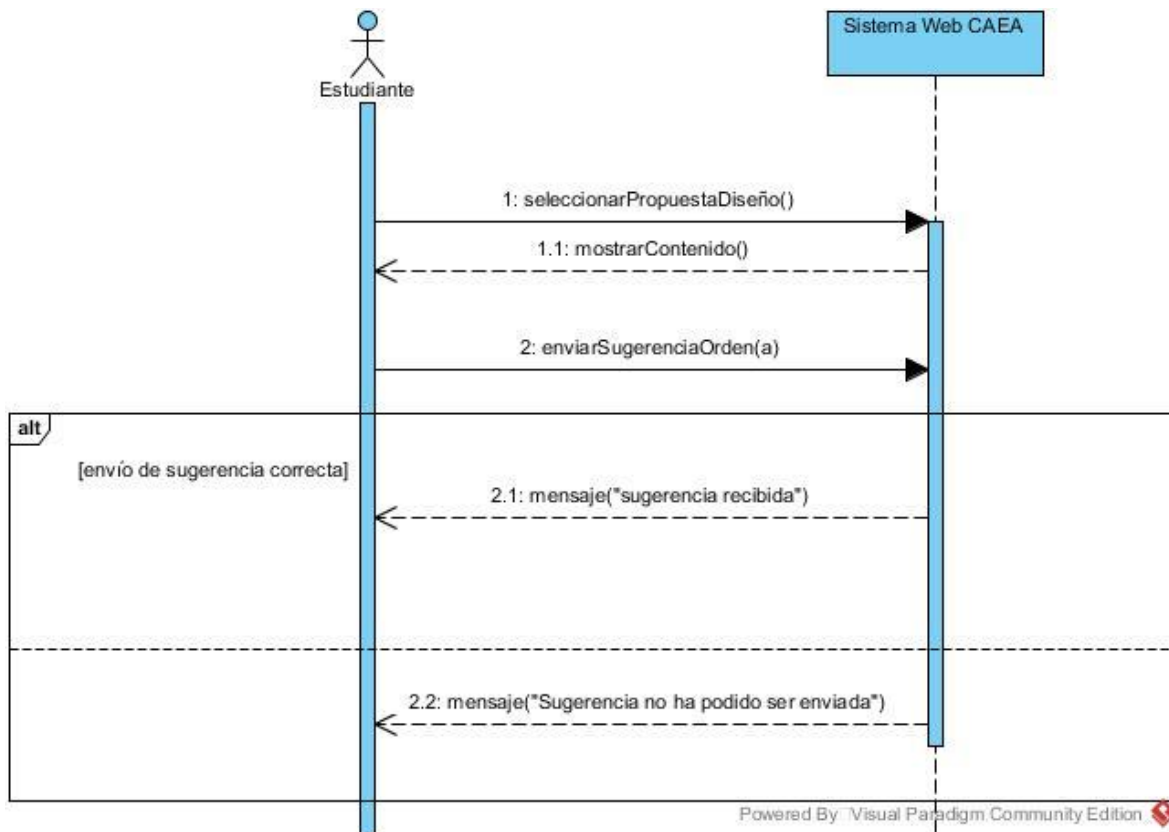
editar.	
	12. El sistema despliega campos de texto editables.
13. El Coordinador ingresa los datos nuevos y presiona “enviar”.	
	14. El sistema recibe los nuevos datos y envía un mensaje de éxito.

Secuencia alternativa actualizar usuario

	6.1 Si no existe coincidencia, el sistema muestra mensaje de error y el proceso regresa a 5.
	14.1 El sistema no recibe correctamente los datos y envía un mensaje de error, el proceso regresa a 11.

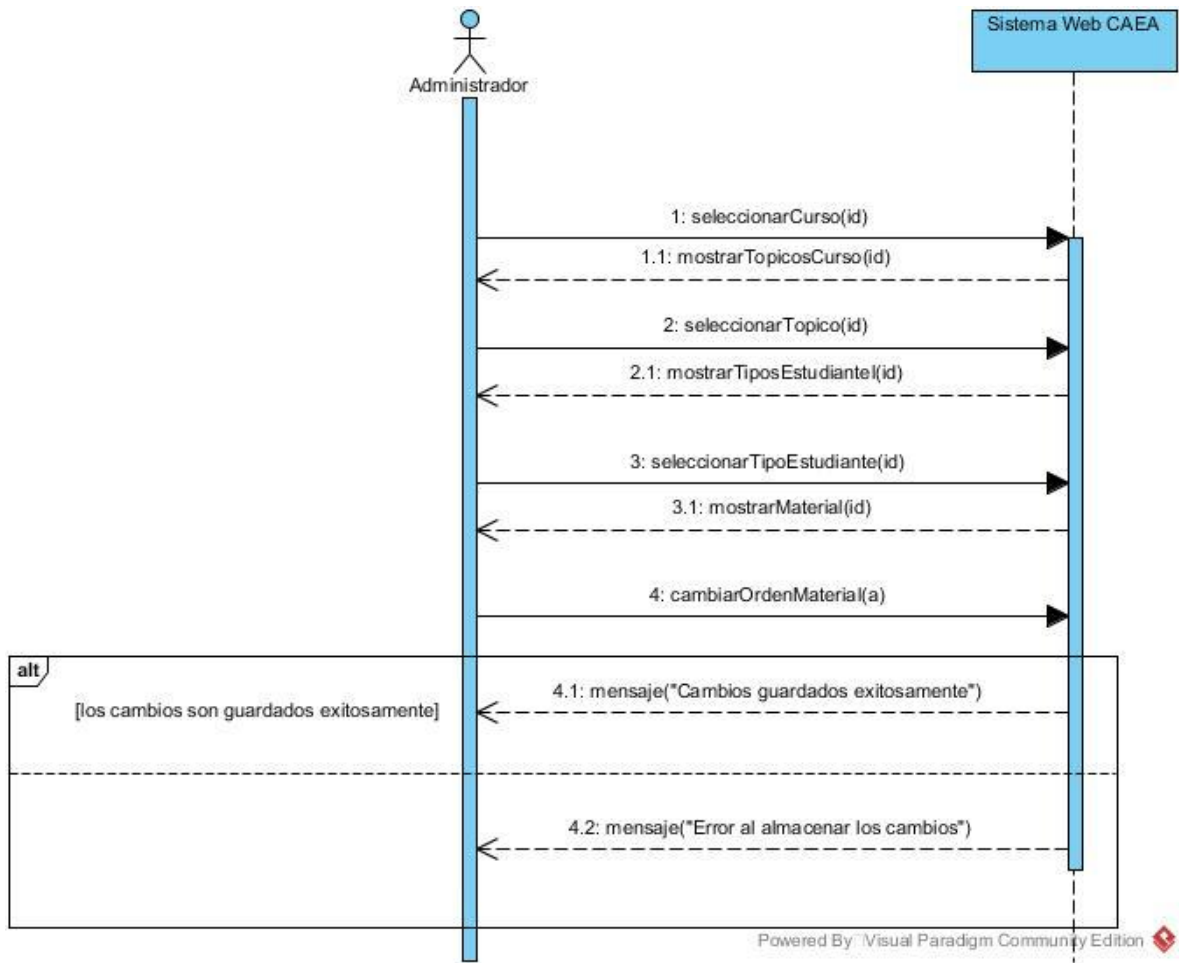
3- Diagramas de secuencia del sistema

1. Generar Feedback:



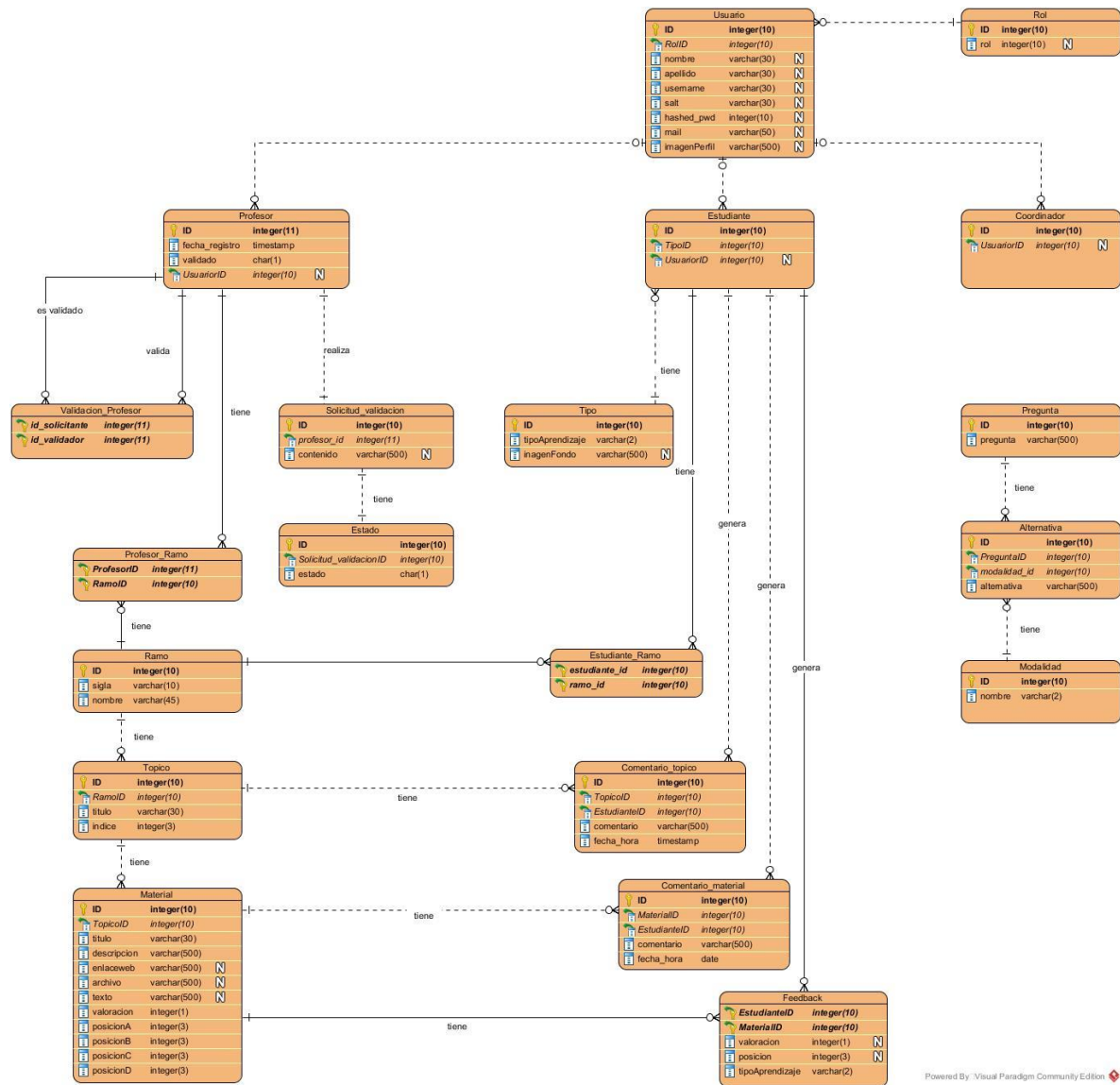
Entregable 2

2. Ordenar contenido:



Entregable 2

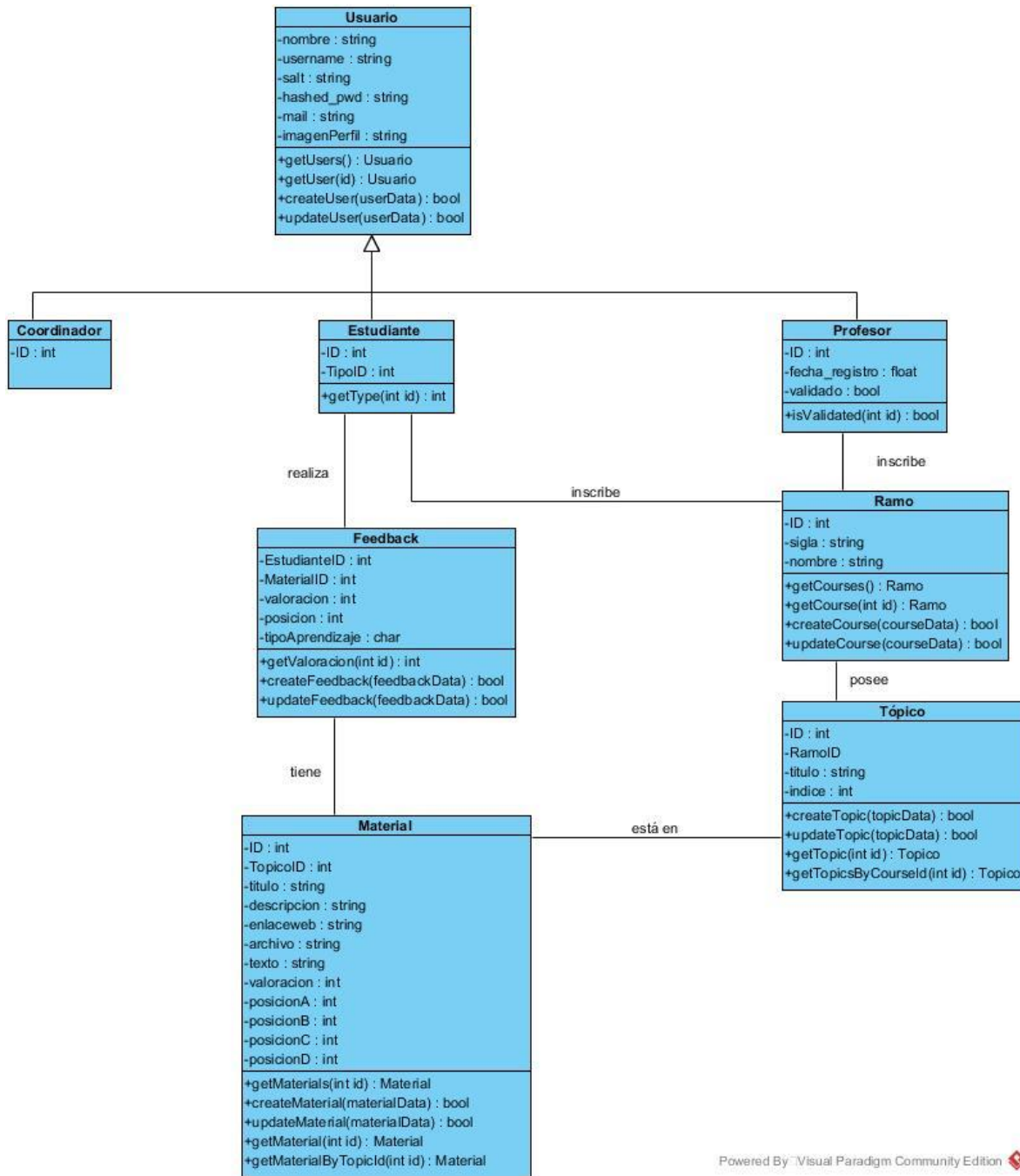
4- Modelo relacional de la base de datos



Entregable 2

Entregable 2

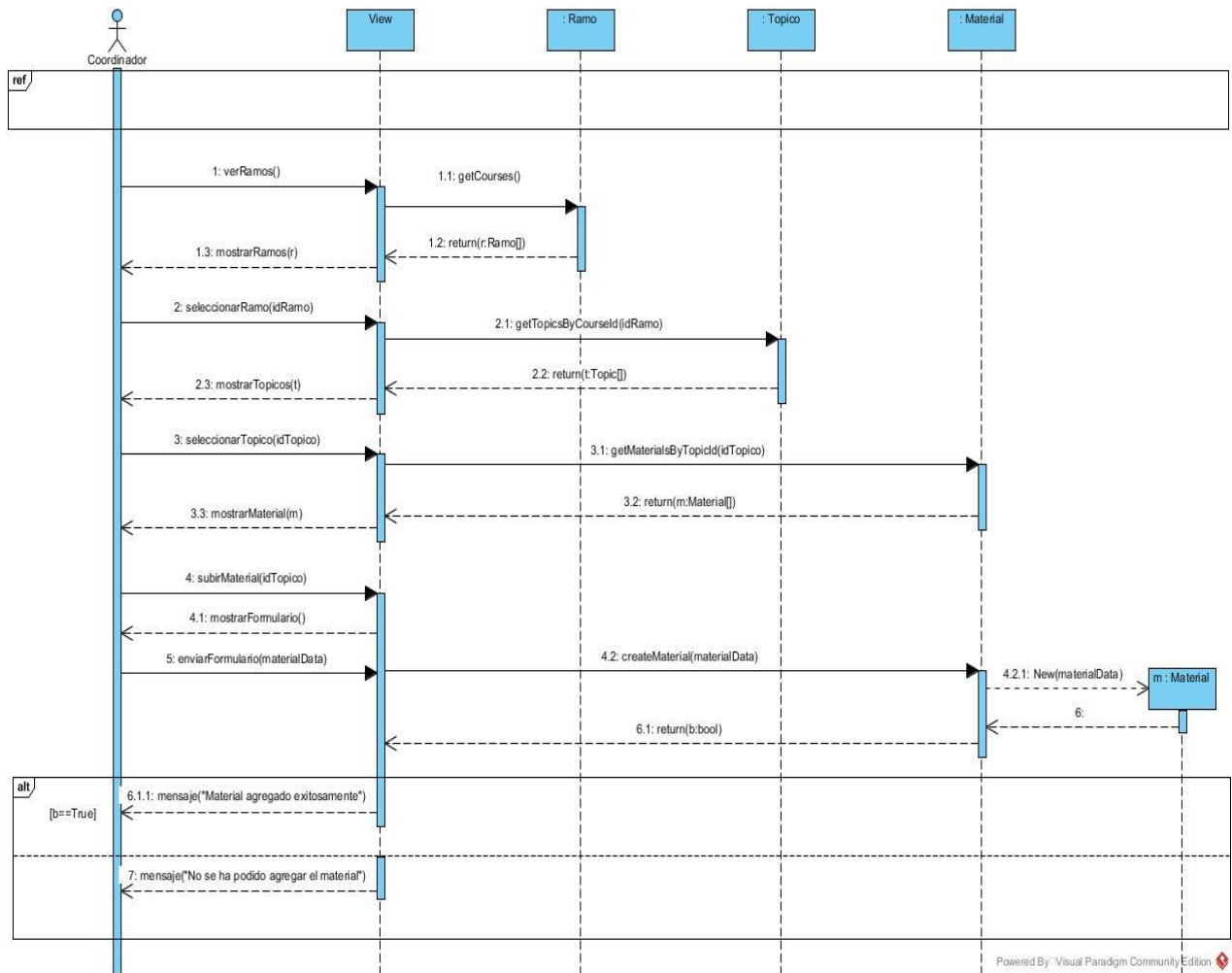
5- Modelo de clases



Powered By Visual Paradigm Community Edition

Entregable 2

6- Diagrama de secuencia de componentes del sistema



Entregable 2

7- Bosquejo MVC

