

Práctica 2

Equipo 103

Francisco Cárdenas Gallardo
Candela Cabrera Sanz
Hiba Tahiri

ÍNDICE

ÍNDICE.....	2
1. Introducción.....	3
2. Listado preliminar de requisitos.....	4
3. Análisis de requisitos (IEEE 830 y métodos débiles).....	5
3.1. Criterios IEEE 830.....	5
3.2. Anotación de los requisitos.....	6
3.3. Checklist de análisis.....	7
3.4. Lista de errores.....	8
3.5. Matriz de interacción.....	9
4. Requisitos ajustados.....	10
5. Historias de usuario.....	13
6. Casos de uso.....	21
6.1. Casos de uso según UML.....	21
6.2. Diagrama de casos de uso.....	28
6.3. Matriz de trazabilidad: Requisitos VS Casos de Uso.....	28
7. Diagrama de clases.....	30
7.1. Especificación de clases.....	31
7.2. Matriz de Trazabilidad: Casos de uso VS Diagrama de Clases.....	36
8. Diagramas de secuencia.....	37
Enlace a Youtrack.....	39
9. Bibliografía.....	39

1. Introducción

Este proyecto tiene como objetivo desarrollar un sistema centralizado enfocado en la tutorización de los alumnos de primer curso de la Universidad de Córdoba, se pretende reemplazar el uso de excel, email y papel para esta actividad. Este sistema cuenta con un chat individual y privado entre los tutores y los alumnos a tutorizar, también cuenta con un sistema de registro de las reuniones tutor-alumno. Este sistema permite enviar alertas cuando el alumno tenga alguna emergencia o haya sucedido algo relevante. El sistema será utilizado por tres roles principales: *alumnos* (de primer curso y movilidad), *tutores* (profesores y/o alumnos de último curso) y *coordinadores* (con permisos de administrador).

La asignación de los tutores, en principio, se realizará de forma automática teniendo en cuenta la facultad y el grado que cursen. Aún así, será posible la asignación manual en el caso del descontento o cualquier otro problema que pueda suceder. Como ya se ha mencionado anteriormente, el sistema va a estar enfocado principalmente en los alumnos de primer curso debido a su posible desubicación en un entorno nuevo, pero también se consideran los alumnos de movilidad (Erasmus o SICUE).

2. Listado preliminar de requisitos

1. Implementación de un chat bidireccional | Funcional
2. Diseño de una interfaz sencilla y fácil de usar | No funcional
3. Almacenamiento de roles de usuario (alumnos o tutores) | Información
4. Asignación por orden alfabético de los apellidos de los alumnos a tutorizar | Funcional
5. Restricción de acceso (solo alumnos de 1º y movilidad) | No funcional
6. Gestión de 3 roles diferentes: alumno, tutor y administrador | Funcional
7. Almacenamiento de los datos de las sesiones de tutoría | Información
8. Desarrollo de un sistema centralizado | No funcional
9. Registro de las sesiones de tutoría (alumno-tutor) | Funcional
10. Restricción de no duplicar el trabajo manual existente | No funcional
11. Acceso al sistema con las mismas credenciales de Moodle | Funcional
12. Conectividad del sistema (web UCO y "desde la calle") | No funcional
13. Generación de informes | Funcional
14. Almacenamiento de estadísticas para informes | Información
15. Gestión de incentivos para tutores (Organización) | No funcional
16. Posibilidad de asignar manualmente un tutor, permitiendo solicitar el cambio de tutor | Funcional
17. El sistema debe tener una interfaz intuitiva para los usuarios | No funcional
18. Implementación de un sistema de valoración de tutores (mejora futura) | Funcional
19. Integración del sistema como un apartado dentro de la web de la UCO | No funcional
20. Recibo y gestión de alertas (académicas, convivencia, salud...) | Funcional
21. Almacenamiento de historial de alertas | Información
22. Duración limitada de los chats (1 año) | No funcional
23. Definición de los permisos de cada uno de los roles | Funcional
24. Evitar la complejidad del sistema | No funcional
25. Almacenamiento de las conversaciones del chat | Funcional
26. Almacenamiento de datos para la asignación (carrera, facultad, apellidos) | Información
27. Protección y privacidad del chat (un tutor solo ve a sus alumnos) | No funcional
28. Debe ser un sistema "conjunto" de la UCO (no un proyecto aislado) | No funcional
29. Garantizar la protección de los datos | No funcional
30. Garantizar la privacidad general de los datos del sistema | No funcional

3. Análisis de requisitos (IEEE 830 y métodos débiles)

3.1. Criterios IEEE 830

- **Corrección:** cada requisito debe reflejar las necesidades de una forma correcta. Así, el programa finalizado deberá ser igual al programa que se pidió.
- **No Ambigüedad:** los requisitos deben tener una única interpretación. El objetivo es reducir la ambigüedad natural del lenguaje cotidiano.
- **Completitud:** se deben cubrir todos los requisitos pertinentes del sistema.
- **Consistencia:** los requisitos deben ser coherentes y no entrar en conflicto. Un conjunto de requisitos contradictorios tiene difícil implementación.
- **Clasificación:** los requisitos se ordenan de acuerdo a su funcionalidad.
- **Verificabilidad:** cada requisito debe poder ser verificado a través de pruebas.
- **Modificabilidad:** los requisitos se podrán cambiar de forma sencilla sin que todo el sistema sufra cambios significativos.
- **Trazabilidad:** será posible rastrear desde el origen hasta la implementación de cada requisito.
- **Requisitos de Rendimiento:** deben quedar especificados los requisitos relacionados con el rendimiento, como la capacidad de soportar un número elevado de usuarios.
- **Restricciones de Diseño:** se debe especificar la accesibilidad desde la web de la UCO.
- **Atributos de Calidad del Sistema:** se considera un aspecto clave de calidad la interfaz intuitiva del sistema.

3.2. Anotación de los requisitos

Requisitos funcionales

- RF1 - Asignación por orden alfabético de los apellidos de los alumnos a tutorizar.
- RF2 - Posibilidad de asignar manualmente un tutor, permitiendo solicitar el cambio de tutor.
- RF3 - Registro de las sesiones de tutoría (alumno-tutor).
- RF4 - Recibo y gestión de alertas (académicas, convivencia, salud...).
- RF5 - Generación de informes.
- RF6 - Implementación de un chat bidireccional.
- RF7 - Almacenamiento de las conversaciones del chat.
- RF8 - Acceso al sistema con las mismas credenciales de Moodle.
- RF9 - Gestión de 3 roles diferentes: alumno, tutor y administrador.
- RF10 - Definición de los permisos de cada uno de los roles.
- RF11 - Implementación de un sistema de valoración de tutores (mejora futura).

Requisitos de información

- RI1 - Almacenamiento de los datos de las sesiones de tutoría.
- RI2 - Almacenamiento de historial de alertas.
- RI3 - Almacenamiento de roles de usuario (alumnos o tutores).
- RI4 - Almacenamiento de datos para la asignación (carrera, facultad, apellidos).
- RI5 - Almacenamiento de estadísticas para informes.

Requisitos no funcionales

- RNF1 - El sistema debe tener una interfaz intuitiva para los usuarios.
- RNF2 - Desarrollo de un sistema centralizado
- RNF3 - Protección y privacidad del chat (un tutor solo ve a sus alumnos)
- RNF4 - Diseño de una interfaz sencilla y fácil de usar
- RNF5 - Conectividad del sistema (web UCO y "desde la calle")
- RNF6 - Duración limitada de los chats (1 año)
- RNF7 - Restricción de acceso (solo alumnos de 1º y movilidad)
- RNF8 - Gestión de incentivos para tutores (Organización)
- RNF9 - Garantizar la protección de los datos
- RNF10 - Evitar la complejidad del sistema
- RNF11 - Garantizar la privacidad general de los datos del sistema
- RNF12 - Restricción de no duplicar el trabajo manual existente
- RNF13 - Debe ser un sistema "conjunto" de la UCO (no un proyecto aislado)
- RNF14 - Integración del sistema como un apartado dentro de la web de la UCO

3.3. Checklist de análisis

Atributo de calidad a considerar	Pregunta
Corrección	¿El requisito describe con precisión la funcionalidad esperada?
No Ambigüedad	¿El requisito está redactado de forma que no permite interpretaciones múltiples?
Completitud	¿El requisito cubre todos los aspectos necesarios o le faltan detalles?
Consistencia	¿El requisito es consistente y no entra en conflicto con otros requisitos?
Concisión	¿El requisito está formulado de forma clara, simple y atómica?
Realizabilidad	¿Es realizable el requisito en la plataforma de implementación?
Verificabilidad	¿Es posible diseñar pruebas finitas y económicas para verificar el cumplimiento del requisito?
Modificabilidad	¿El requisito permite modificaciones sin afectar de forma impredecible al resto del sistema?
Trazabilidad	¿Es posible rastrear este requisito a través del diseño y desarrollo (identificar su origen)?
Independencia del diseño	¿El requisito describe qué debe hacer el sistema sin especificar cómo implementarlo?
Clasificación	¿El requisito está correctamente clasificado (RF, RI, RNF)?
Estabilidad	¿El requisito es estable o es probable que sufra cambios frecuentes?
Relevancia	¿El requisito cubre una necesidad real del usuario (alumno, tutor, admin) o es innecesario?

3.4. Lista de errores

ID de Requisito	Defectos identificados (Atributo de Calidad)	Acciones recomendadas
RF 2	Concisión	Estamos mezclando dos ideas en una. Una cosa es que el coordinador pueda asignar a mano, y otra muy distinta es que el alumno pueda pedir un cambio.
RF 5	Compleitud	Esto está incompleto. "Generación de informes" es demasiado escueto. Si le damos esto a un programador no sabrá exactamente qué debe contener el informe.
RF 9 y RF 10	Concisión (Redundancia)	Aquí hay redundancia. RF 9 habla de los roles y RF10 de sus permisos. En realidad, un rol es un conjunto de permisos.
RNF 1, RNF 4, RNF 10	Concisión (Redundancia) y Verificabilidad	"Intuitiva", "sencilla" y "evitar complejidad" vienen a decir lo mismo.
RNF 8	Corrección / Realizabilidad	La "Gestión de incentivos" no es algo que esté realmente relacionado con el software. Es un convenio de la universidad con algunas otras empresas.
RF 6, RF 7	Concisión (Redundancia)	El "almacenamiento" (RF 7) no es una función separada, es una característica imprescindible del "chat" (RF 6). No tendría sentido un chat que borra los mensajes al cerrar la ventana.
RNF 3, RNF 9, RNF 11	Concisión (Redundancia)	RNF 9 y RNF 11 son básicamente lo mismo (cumplir la RGPD) y RNF 3 no es un requisito nuevo, es solo el ejemplo más importante de cómo se aplica esa privacidad.

3.5. Matriz de interacción

Tras realizar un cruce exhaustivo de los 30 requisitos (cuya matriz completa se puede consultar aquí), hemos extraído la siguiente matriz resumen. En ella se destacan las interacciones más importantes que encontramos: los Solapamientos como 'S', que hay que limpiar. En este caso no se han encontrado conflictos, es por ello que no se indican en la matriz.

	RF 6	RF 8	RI 4	RI 5	RNF 8	RNF 9	RNF 10	RNF 12 RNF 13
RF 6	S							
RF 7	S							
RF 9		S						
RF 10		S						
RNF 1 / RNF 2				S	S			
RNF 3 / RNF 4 / RNF 5				S		S	S	S
RNF 9				S		S		
RNF 10			S	S				
RNF 11				S	S			
RNF 14				S				

4. Requisitos ajustados

Requisitos funcionales

- **RF 1. Asignación por orden alfabético de los apellidos de los alumnos a tutorizar:**
 - El sistema ejecutará el algoritmo de asignación automáticamente.
 - La lógica se basará en el orden alfabético de los apellidos de los alumnos.
- **RF 2. Asignación manual y solicitud de cambio:**
 - El Coordinador tendrá una herramienta para asignar un tutor a un alumno manualmente.
 - El Estudiante podrá iniciar una solicitud de cambio de tutor.
 - El sistema debe mostrar al Coordinador una notificación por cada solicitud de cambio pendiente.
- **RF 3. Registro de sesiones:**
 - El sistema permitirá al tutor registrar cada sesión de tutoría.
 - El tutor tendrá la opción de marcar la propia conversación del chat como registro oficial.
 - Alternativamente, el tutor podrá introducir la fecha y un resumen breve de la sesión.
- **RF 4. Recibo y gestión de alertas (académicas, convivencia, salud...):**
 - El sistema debe permitir a estudiantes y tutores crear una alerta.
 - La alerta debe incluir un campo obligatorio para el tipo de problema (académico, convivencia, salud).
 - El Coordinador debe recibir una notificación inmediata de las alertas registradas.
- **RF 5. Generación de informes:**
 - El sistema debe proporcionar al Coordinador la capacidad de generar informes para el seguimiento del programa.
 - Los informes deben incluir métricas clave como: número de alumnos por tutor o número de alertas por tipo, entre otras funciones.
 - Se debe permitir exportar los informes en formatos estándar.
- **RF 6. Implementación de un chat bidireccional con almacenamiento de conversaciones:**
 - El chat debe permitir una comunicación fluida entre el tutor y el alumno.
 - Las conversaciones deben almacenarse de forma segura para consultas futuras.
 - El tutor debe tener un listado de sus alumnos para iniciar conversaciones con facilidad.

- **RF 7. Acceso al sistema con las mismas credenciales de Moodle:**
 - El sistema utilizará la autenticación centralizada de la UCO para validar la identidad de todos los usuarios (Estudiante, Tutor, Coordinador).
- **RF 8. Gestión de Acceso por Roles:**
 - El sistema debe distinguir entre tres roles principales: Alumno, Tutor y Administrador (Coordinador).
 - Los permisos de acceso y las funcionalidades disponibles deben ser exclusivos de cada rol.
- **RF 9. Implementación de un sistema de valoración de tutores (mejora futura):**
 - La función de valoración solo estará activa durante un periodo definido.
 - Los estudiantes deben poder puntuar al tutor en distintos aspectos.
 - Las valoraciones deben guardarse de forma anónima para proteger la privacidad del alumno.

Requisitos de información

- **RI 1. Almacenamiento de los datos de las sesiones de tutoría:** Se debe almacenar el registro de sesiones (ya sea el chat o el resumen manual).
- **RI 2. Almacenamiento de historial de alertas:** Se deben guardar los datos de todas las alertas generadas (fecha, tipo, descripción, autor).
- **RI 3. Almacenamiento de roles de usuario (alumnos o tutores):** Se debe guardar la información del rol de cada usuario para la gestión de permisos.
- **RI 4. Almacenamiento de datos para la asignación (carrera, facultad, apellidos)**
- **RI 5. Almacenamiento de estadísticas para informes:** Se deben agregar y guardar las métricas necesarias para generar los informes .

Requisitos no funcionales

De organización

- **RNF 1. Restricción de acceso:** El acceso al sistema debe estar limitado únicamente a los estudiantes de primer año y a los alumnos de movilidad (Erasmus/SICUE).
- **RNF 2. Restricción de no duplicar el trabajo manual existente:** El diseño de la aplicación debe sustituir los procesos manuales actuales (Excel, email, etc.) para mejorar la eficiencia del Coordinador y los Tutores.
- **RNF 3. Debe ser un sistema "conjunto" de la UCO:** Debe percibirse como un sistema oficial y no como un proyecto ajeno a la estructura universitaria.

De producto

- **RNF 4. Usabilidad del sistema:** La interfaz será clara y sencilla.
- **RNF 5. El sistema debe garantizar la privacidad (RGPD):** El sistema debe cumplir con la RGPD, limitando el acceso de los tutores únicamente a los alumnos que tienen asignados.
- **RNF 6. Conectividad del sistema (web UCO y "desde la calle"):** El sistema debe ser accesible desde cualquier dispositivo y ubicación fuera de la red de la UCO.
- **RNF 7. Duración limitada de los chats (1 año):** El sistema debe eliminar automáticamente el historial de chat después de que transcurra un año desde la última interacción.

Externo

- **RNF 8. Integración en la web de la UCO:** La integración en la web UCO será un hipervínculo para asegurar la independencia del sistema y cumplir con la disponibilidad 24/7.

5. Historias de usuario

Título	Asignación manual de tutores
Identificador	HU-01
Descripción	Como coordinador, quiero asignar un tutor a un alumnos de forma manual para gestionar casos especiales como el idioma u otros problemas.
Responsable	Candela Cabrera Sanz
Criterios de validación	<ul style="list-style-type: none">El coordinador podrá realizar una búsqueda en el alumno y visualizar su tutor actual.En el perfil del alumno, el coordinador tendrá una opción para cambiar el tutor y elegir uno distinto.Una vez confirmado el cambio, el alumno debe quedar eliminado del listado de alumnos del tutor antiguo y aparecer en el listado de alumnos del tutor nuevo asignado.
Prioridad	Alta
Estimación	2 días
Sprint	1

Título	Solicitud de cambio de tutor
Identificador	HU-02
Descripción	Como alumno, quiero poder solicitar un cambio de tutor si no estoy satisfecho para tener una mejor experiencia.
Responsable	Candela Cabrera Sanz
Criterios de validación	<ul style="list-style-type: none"> El alumno debe disponer en su perfil de una opción “Solicitar cambio de tutor”. El sistema debe permitir registrar el motivo de la solicitud y enviarla al coordinador el cual será notificado. Mientras que el coordinador tendrá una notificación pendiente, el alumno recibirá un mensaje de confirmación de que su solicitud ha sido enviada correctamente.
Prioridad	Media
Estimación	1 día
Sprint	2

Título	Registro de sesiones de tutoría
Identificador	HU-03
Descripción	Como tutor, quiero registrar una sesión de forma rápida para mantener un historial de todos mis alumnos.
Responsable	Hiba Tahiri
Criterios de validación	<ul style="list-style-type: none"> El tutor debe tener un apartado que le permita seleccionar el alumno y la función que quiere realizar: ver el historial de sesiones o añadir una nueva sesión. En el historial de sesiones se mostrará el alumno seleccionado y todas las sesiones ordenadas por fecha. En el caso de seleccionar la opción de añadir una nueva sesión, se pedirá la fecha y un resumen de lo que se ha tratado en la sesión de tutoría. Al guardar una nueva sesión debe aparecer en el historial privado del alumno, visible para el tutor.
Prioridad	Alta
Estimación	2 días
Sprint	2

Título	Envío de alertas
Identificador	HU-04
Descripción	Como tutor o alumno, quiero poder enviar alertas sobre el estado del alumno para informar al tutor y al coordinador y se tenga constancia de ello.
Responsable	Francisco Cárdenas Gallardo
Criterios de validación	<ul style="list-style-type: none"> • Tanto el tutor como el alumno deben poder seleccionar la opción “Enviar alerta”. • Una vez seleccionada esta opción, se seleccionará el tipo de alerta (académica, salud, convivencia) y se proporcionará una descripción obligatoria del aviso. • Al enviar la alerta se deberá registrar y notificar a todos los usuarios relacionados con ese alumno.
Prioridad	Alta
Estimación	2 días
Sprint	2

Título	Generación de informes
Identificador	HU-05
Descripción	Como coordinador, quiero generar informes para poder supervisar el funcionamiento y estado general del sistema.
Responsable	Hiba Tahiri
Criterios de validación	<ul style="list-style-type: none"> ● El coordinador debe disponer de una sección “Informes”. ● Debe poder seleccionar entre distintas opciones: número de alumnos por tutor, número de alertas (pudiendo ordenarlas por tipo, por tutor y por alumno), número de sesiones medio por tutores y alumnos. ● Los informes deben reflejar métricas clave del sistema y ser accesibles desde la web UCO.
Prioridad	Media
Estimación	3 días
Sprint	3

Título	Chat bidireccional
Identificador	HU-06
Descripción	Como alumno o tutor, quiero enviar y recibir mensajes del chat para tener una comunicación fluida, directa y privada.
Responsable	Francisco Cárdenas Gallardo
Criterios de validación	<ul style="list-style-type: none"> ● El sistema debe tener una sección titulada “Chat”. ● El alumno solo puede ver y enviar mensajes con su tutor asignado, al igual que el tutor únicamente con sus alumnos asignados. ● El tutor tendrá un listado de todos sus alumnos tutorizados y así poder iniciar un chat con cualquiera de ellos. ● Los mensajes enviados deben guardarse en el historial de la conversación.
Prioridad	Alta
Estimación	3 días
Sprint	1

Título	Acceso al sistema
Identificador	HU-07
Descripción	Como usuario, quiero acceder al sistema usando mis credenciales de Moodle para simplificar el proceso de autenticación.
Responsable	Francisco Cárdenas Gallardo
Criterios de validación	<ul style="list-style-type: none"> • La pantalla de inicio debe mostrar una opción que sea “Acceder con Moodle”, esto hará que se redirija al portal de login oficial de la Universidad. • Si el login es correcto, el usuario será redirigido al sistema donde podrá acceder al panel de control, dependiendo de su rol asignado.
Prioridad	Alta
Estimación	2 días
Sprint	1

Título	Valoración de tutores
Identificador	HU-08
Descripción	Como alumno, quiero valorar a mi tutor al finalizar el curso para dar mi opinión sobre mi experiencia y posibles mejoras.
Responsable	Candela Cabrera Sanz
Criterios de validación	<ul style="list-style-type: none"> • Esta función solo estará activa durante un periodo definido. • El alumno podrá puntuar al tutor y añadir un comentario escrito. • El sistema guardará las valoraciones.
Prioridad	Baja
Estimación	2 días
Sprint	4

6. Casos de uso

6.1. Casos de uso según UML

ID de Caso de Uso	CU-01
Nombre	Acceder al sistema.
Objetivo	Permitir al usuario entrar al sistema y ser autenticado, cargando la interfaz que se corresponde con su rol ya asignado.
Contexto	El sistema no tiene un registro propio de las credenciales de los usuarios, es decir, depende en su totalidad de la autenticación de Moodle para validar su identidad.
Actores	Usuarios: coordinador, tutor y alumno.
Escenario principal	<ol style="list-style-type: none">1. El usuario navega a la dirección web del sistema.2. El sistema mostrará en la pantalla de bienvenida una única opción que es “Acceder con Moodle”.3. Una vez seleccionada esta opción, el usuario será redirigido al portal de acceso de la UCO (Moodle).4. El usuario introduce sus credenciales (usuario y contraseña).5. El servicio de moodle valida al usuario y lo dirige de vuelta al sistema de tutorías.6. El sistema identifica el rol del usuario y le muestra su panel principal.
Extensiones	<p>4a. El usuario introduce credenciales incorrectas.</p> <p>4a.1. El sistema de Moodle muestra un error y se finaliza.</p> <p>4b. El usuario no recuerda su contraseña.</p> <p>4b.1. El usuario utilizará la opción “Olvidé mi contraseña” de Moodle.</p> <p>6a. Moodle valida al usuario pero no cumple los requisitos de acceso (no es de primero, ni tutor, ni de movilidad).</p> <p>6a.1. El sistema le muestra un mensaje de “Acceso no autorizado” y la sesión será cerrada.</p>

ID de Caso de Uso	CU-02
Nombre	Gestionar asignaciones de tutores.
Objetivo	El sistema normalmente realizará una asignación automática de los tutores pero también se debe permitir al coordinador intervenir en esta asignación si existiese algún problema, resultando esto en la opción de la asignación manual.
Contexto	El sistema debe asignar un tutor a cada alumno de primer ingreso y de movilidad basándose en sus características (carrera, facultad, apellido, idiomas...). El proceso principal se realiza de forma automática, pero el coordinador tendrá el poder de intervenir manualmente para corregir o gestionar casos especiales.
Actores	Coordinador, sistema.
Escenario principal	<ol style="list-style-type: none"> 1. El coordinador tendrá la opción “Iniciar asignación automática”. 2. El sistema accede a la lista de alumnos y de tutores, para cada alumno el sistema aplica las reglas de asignación: carrera y facultad. 3. En el caso de que haya varios tutores y/o alumnos se asignan por orden alfabético. 4. El sistema crea el vínculo alumno-tutor. 5. El sistema finaliza y notifica al coordinador los resultados.
Extensiones	<p>1a. El coordinador interviene para asignar de forma manual.</p> <p>1a.1. El coordinador accede a “Gestión de Alumnos”.</p> <p>1a.2. Se realiza una búsqueda del alumno.</p> <p>1a.3. Usa la opción “Asignar tutor” y selecciona uno.</p> <p>1a.4. El sistema guarda el vínculo.</p> <p>1b. El coordinador interviene como respuesta a una solicitud del alumno.</p> <p>1b.1. El alumno envía “Solicitud de cambio”.</p> <p>1b.2. Una vez recibe el coordinador esta solicitud, se revisa el caso y sigue los pasos de la asignación manual.</p>

ID de Caso de Uso	CU-03
Nombre	Registrar sesión de tutoría.
Objetivo	Permitir al tutor registrar las sesiones de tutoría y llevar un registro digital y privado de estas sesiones.
Contexto	Esta funcionalidad sustituye el registro de reuniones en papel y es una herramienta de seguimiento de los alumnos. Los alumnos no tienen acceso a esta sección.
Actores	Tutor.
Escenario principal	<ol style="list-style-type: none"> 1. El tutor accede desde el panel a una sección llamada “Sesiones”. 2. El tutor selecciona a un alumno de su listado. 3. El sistema muestra el historial de sesiones de ese alumno ordenadas por fecha. 4. El tutor consulta los registros para recordar acuerdos previos.
Extensiones	<p>4a. El tutor decide añadir un nuevo registro.</p> <p>4a.1. El tutor selecciona la opción “Registrar nueva sesión”.</p> <p>4a.2. El sistema muestra un formulario con los datos necesarios: fecha, resumen y acuerdos.</p> <p>4a.3. Una vez rellenados los campos, se guarda la sesión.</p> <p>4a.4. El sistema almacena el nuevo registro y actualiza la vista del historial de sesiones añadiendo la nueva.</p>

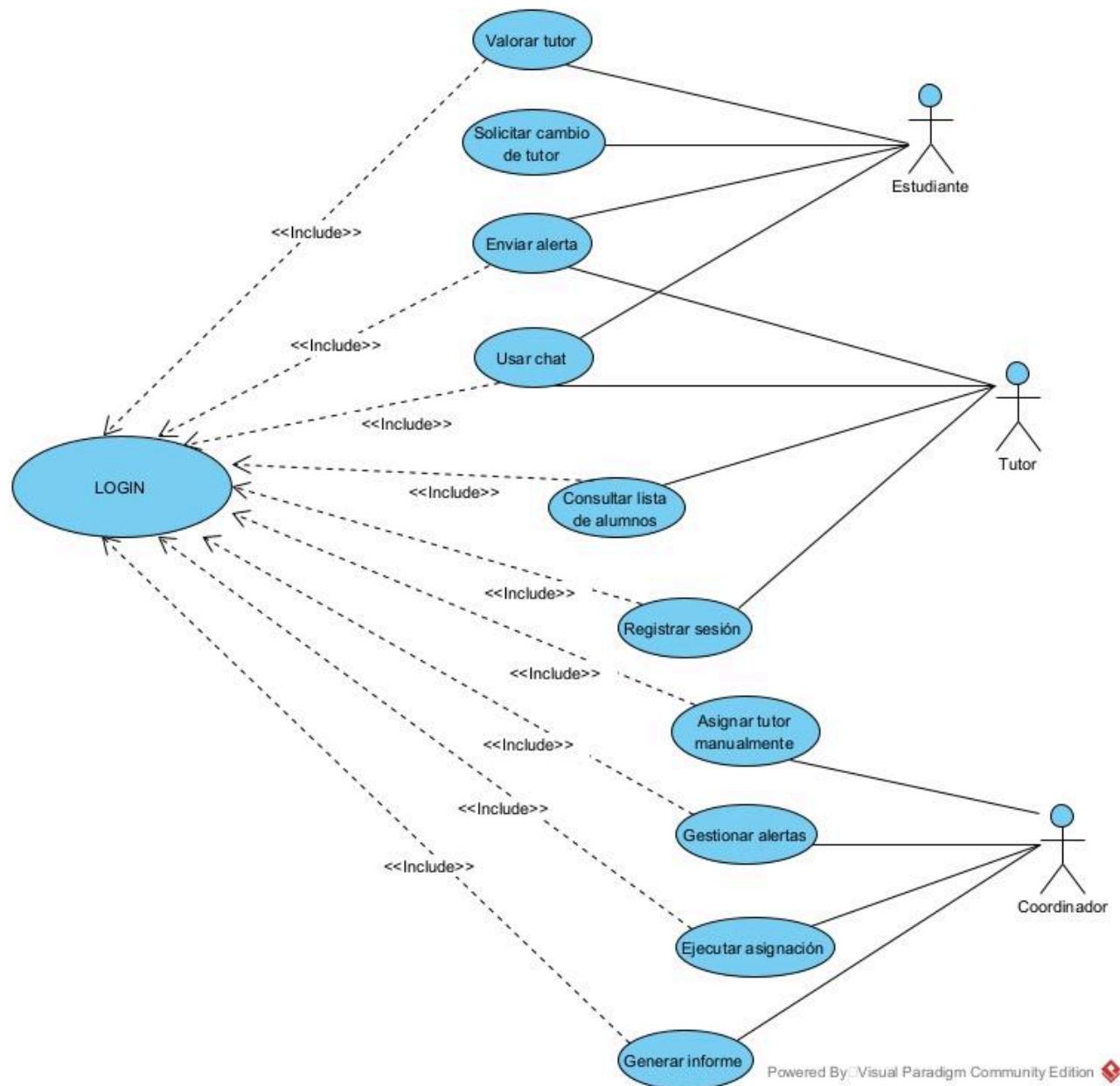
ID de Caso de Uso	CU-04
Nombre	Gestionar alertas.
Objetivo	Permitir a tutores y alumnos enviar alertas sobre problemas o advertencias, y permitir al coordinador gestionarlas para su seguimiento.
Contexto	Las alertas son primordiales para la detección temprana de problemas de los alumnos. Se utilizará para comunicar problemas académicos, de convivencia o de salud, entre otros.
Actores	Alumno, tutor, coordinador.
Escenario principal	<ol style="list-style-type: none"> 1. El usuario accede a la sección “avisos”, seguidamente se mostrará una opción de “enviar nueva alerta”. 2. El sistema muestra un formulario pidiendo: tipo de alerta y una breve descripción. 3. Una vez llenado el formulario, se envía. 4. El sistema guarda la alerta y notifica al coordinador y al alumno o tutor (dependiendo de quien la haya enviado). 5. El sistema mostrará, finalmente, un mensaje de confirmación.
Extensiones	<p>3a. El usuario presiona “enviar” pero no se ha llenado.</p> <p>3a.1. El sistema muestra un error y no se envía la alerta.</p> <p>4a. El coordinador es notificado (también lo es el tutor en el caso de que la alerta haya sido enviada por el alumno, o al alumno si la alerta ha sido enviada por el tutor).</p> <p>4a.1. El usuario accede a su panel “Gestión de alertas”.</p> <p>4a.2. El sistema muestra una lista de alertas pendientes.</p> <p>4a.3. Al seleccionar una alerta se mostrarán los detalles: quién la envió, el tipo y la descripción.</p> <p>4a.4. Una vez gestionado el problema, se marcará la alerta como revisada o resuelta.</p>

ID de Caso de Uso	CU-05
Nombre	Consultar panel de control.
Objetivo	Permitir al coordinador visualizar datos ya procesados sobre el funcionamiento del programa para supervisar el estado general y facilitar la detección temprana de problemas.
Contexto	En la pantalla principal del coordinador se mostrará una opción que permite ver los informes teniendo así un resumen visual con los datos más importantes.
Actores	Coordinador.
Escenario principal	<ol style="list-style-type: none"> 1. El coordinador accede al sistema y se muestra el panel de control, en él aparece la sección “Informes”. 2. Una vez dentro de informes se nos muestran distintas opciones de informes disponibles como por ejemplo número de alumnos por tutor, número de alertas por tipo, sesiones medias, entre otros. 3. El coordinador selecciona un informe y selecciona generar. 4. El sistema consulta los datos y muestra el informe en pantalla con gráficos incluidos.
Extensiones	<p>2a. También se permitirá filtrar los datos en las tablas en función de en lo que se quiera enfocar la búsqueda.</p> <p>4a. El coordinador quiere el informe en un archivo.</p> <p>4a.1. Se dispondrá de la opción “descargar como PDF”.</p> <p>4a.2. El sistema genera el archivo y lo descarga.</p>

ID de Caso de Uso	CU-06
Nombre	Mantener comunicación (chat).
Objetivo	Permitir una comunicación fluida, directa y privada entre el alumno y su tutor asignado.
Contexto	En este canal de comunicación bidireccional se pretenden solventar dudas de una forma rápida, sustituyendo los correos que suelen demorarse más en responder. Debe ser privado y seguro, y las interfaces de los alumnos y los tutores son distintas. Las conversaciones serán borradas al año de su creación.
Actores	Tutor, alumno.
Escenario principal	<ol style="list-style-type: none"> 1. El usuario accede a la sección “Chat”. 2. El usuario selecciona la conversación con el destinatario. 3. Una vez abierto el chat, se escribirá el mensaje y se enviará. 4. El sistema guarda el mensaje en la conversación y lo muestra en la pantalla de ambos usuarios. 5. El destinatario recibe una notificación al instante.
Extensiones	<p>2a. Si el usuario es tutor:</p> <p>2a.1. El sistema le muestra una lista con todos sus alumnos asignados.</p> <p>2a.2. El tutor selecciona a un alumno para iniciar la conversación.</p> <p>2b. Si el usuario es alumno:</p> <p>2b.1. El sistema le muestra únicamente la conversación con su tutor asignado.</p>

ID de Caso de Uso	CU-07
Nombre	Valorar tutor.
Objetivo	Permitir al alumno valorar a su tutor de forma anónima al final del curso, para ayudar a mejorar el servicio.
Contexto	Esta funcionalidad se considera una mejora futura, por ello su baja prioridad. Es un proceso que se activa durante un periodo de tiempo concreto y su valoración será anónima para proteger al alumno.
Actores	Alumno.
Escenario principal	<ol style="list-style-type: none"> 1. El sistema activa automáticamente el periodo de valoración. 2. Al entrar, el sistema notifica al alumno que el periodo de valoración está abierto. 3. El alumno accede a la sección “Valorar a mi tutor”. 4. El sistema muestra un formulario. 5. El alumno rellena la valoración y lo envía. 6. El sistema guarda los datos de forma anónima y muestra un mensaje de que se ha guardado con éxito.
Extensiones	<p>2a. El alumno ignora la notificación y decide no valorar al tutor.</p> <p>2a.1. El alumno cierra la notificación.</p>

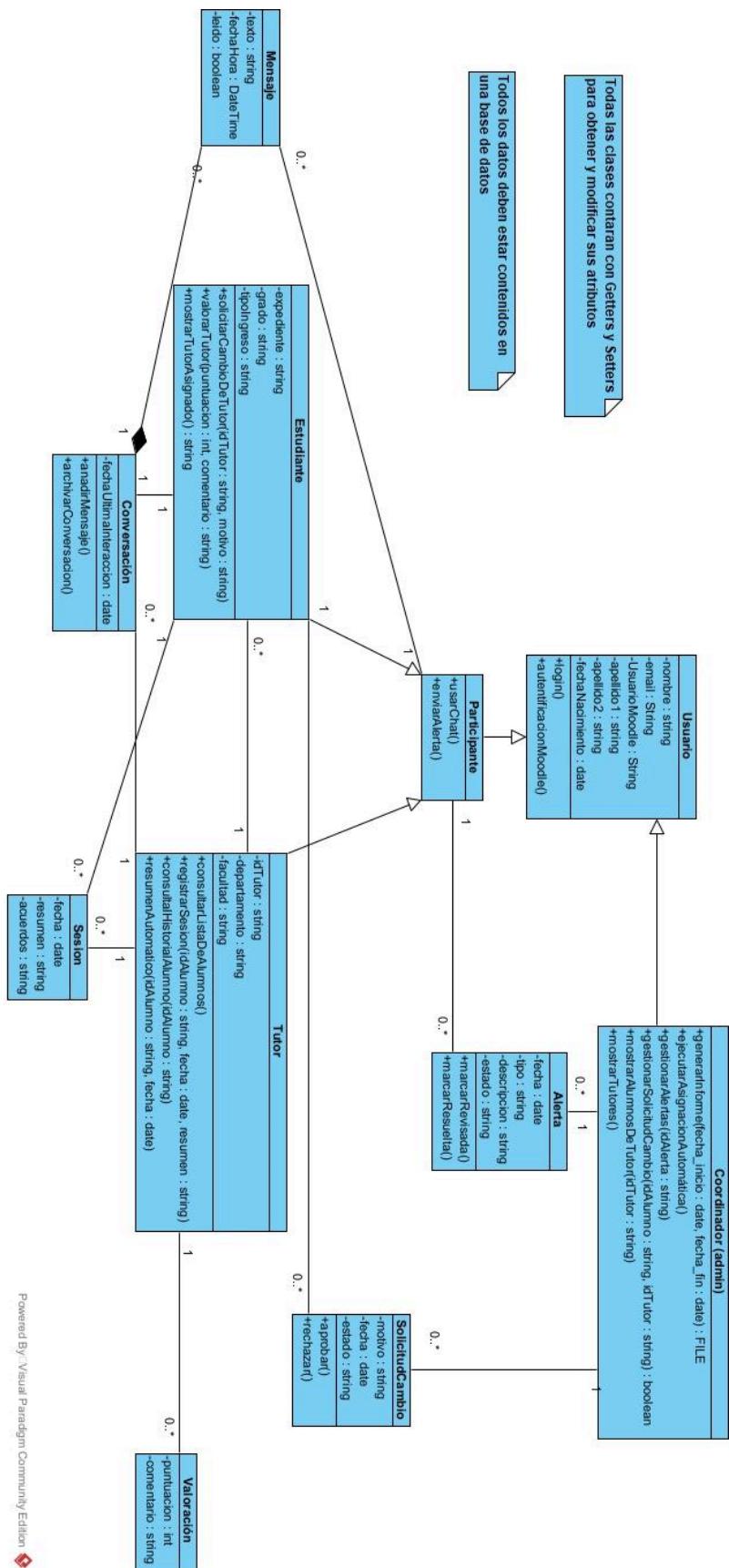
6.2. Diagrama de casos de uso



6.3. Matriz de trazabilidad: Requisitos VS Casos de Uso

	CU 1	CU 2	CU 3	CU 4	CU 5	CU 6	CU 7
RF 1							
RF 2							
RF 3							
RF 4							
RF 5							
RF 6							
RF 7							
RF 8							
RF 9							

7. Diagrama de clases



7.1. Especificación de clases

Clase: Usuario			
Clase base para todos los participantes del sistema			
Atributos			
-	Nombre	string	Nombre del usuario
-	email	string	Email del usuario
-	Usuario moodle	string	Usuario moodle del usuario
-	Apellido 1	string	El primer apellido del usuario
-	Apellido 2	string	El segundo apellido del usuario si tiene
-	Fecha nacimiento	date	Fecha nacimiento del usuario
Operaciones			
+	Login()	void	Usuario puede acceder a moodle
+	Autentificación moodle()	void	Verifica la identidad del usuario a través de Moodle para acceder al sistema.

Clase: Estudiante			
Usuario que recibe tutorías y puede interactuar con tutores y coordinadores			
Atributos			
-	Expediente	string	El expediente del estudiante
-	Grado	string	El grado del estudiante
-	Tipo Ingreso	string	El tipo de ingreso del estudiante
Operaciones			
+	Solicitar cambio de tutor ()	void	Solicita el cambio de tu tutor actual por otro disponible
+	Valorar tutor ()	void	Evaluar el trabajo del tutor
+	Mostrar tutor asignado()	string	Muestra el tutor que tiene el estudiante actualmente

Clase: tutor			
Usuario tutor que gestiona su lista de alumnos y el registro de sesiones.			
Atributos			
-	idTutor	string	Identificador único del tutor.
-	departamento	string	Departamento al que pertenece el tutor.
-	facultad	string	Facultad a la que pertenece el tutor.
Operaciones			
+	consultarListaDeAlumnos	void	Muestra la lista de alumnos que tiene asignados.
+	registrarSesion	void	Crea un nuevo registro de sesión para un alumno.
+	consultarHistorialAlumno	void	Muestra el historial de sesiones registradas para un alumno específico.
+	resumenAutomatico	void	Genera un resumen de la interacción con un alumno.

Clase: coordinador			
Es el administrador con permisos para ejecutar asignaciones, gestionar alertas y generar informes del sistema.			
Atributos			
Los hereda de usuario.			
Operaciones			
+ generarResultado	FILE	Genera un archivo con los datos seleccionados.	
+ ejecutarAsignacionAutomática	void	Inicia el proceso automático de asignaciones de tutores.	
+ gestionarAlertas	void	Permite al coordinador ver, filtrar y marcar alertas como resueltas.	
+ gestionarSolicitudCambio	boolean	Procesa la solicitud del alumno asignándole un nuevo tutor manualmente.	
+ mostrarAlumnosDeTutor	void	Muestra la lista de alumnos asignados a un tutor específico.	
+ mostrarTutores	void	Muestra la lista de todos los tutores registrados en el sistema.	

Clase: Sesión			
Registro de una sesión de tutoría entre tutor y estudiante			
Atributos			
- fecha	date	Fecha de la sesión	
- resumen	string	Resumen de la sesión	
- acuerdos	string	Acuerdos de la sesión	

Clase: valoración			
Almacena la puntuación y el comentario que un alumno otorga a un tutor.			
Atributos			
- puntuación	int	Valoración numérica.	
- comentario	string	Comentario sobre la tutoría.	

Clase: Conversación			
Canal de comunicación entre estudiante y tutor			
Atributos			
-	Fecha última interacción	date	Fecha última interacción de la conversación
Operaciones			
+	Añadir mensaje ()	void	Escribe un nuevo mensaje
+	Archivar conversación ()	void	Guarda conversación en el archivo

Clase: Solicitud Cambio			
Solicitud realizada por el estudiante para cambiar de tutor			
Atributos			
-	fecha	date	Fecha de solicitud cambio
-	motivo	string	Motivo de solicitud cambio
-	estado	string	Estado de solicitud cambio
Operaciones			
+	Aprobar ()	void	La solicitud de cambio ha sido aprobada
+	Rechazar ()	void	La solicitud de cambio ha sido rechazada

Clase: participante			
Clase que agrupa las funciones comunes del chat y alertas.			
Atributos			
Los hereda de usuario.			
Operaciones			
+	usarChat	void	Permite el acceso a la función de chat.
+	enviarAlerta	void	Permite el envío de una alerta al sistema.

Clase: Alerta			
Notificación generada por participantes para informar sucesos o eventos			
Atributos			
-	fecha	date	Fecha de la alerta
-	tipo	string	Tipo de la alerta
-	descripción	string	Descripción de la alerta
-	estado	string	Estado de la alerta
Operaciones			
+	Marcar Revisada ()	void	Verificar y revisar la alerta
+	Marcar resultado ()	void	Registrar los resultados de la alerta

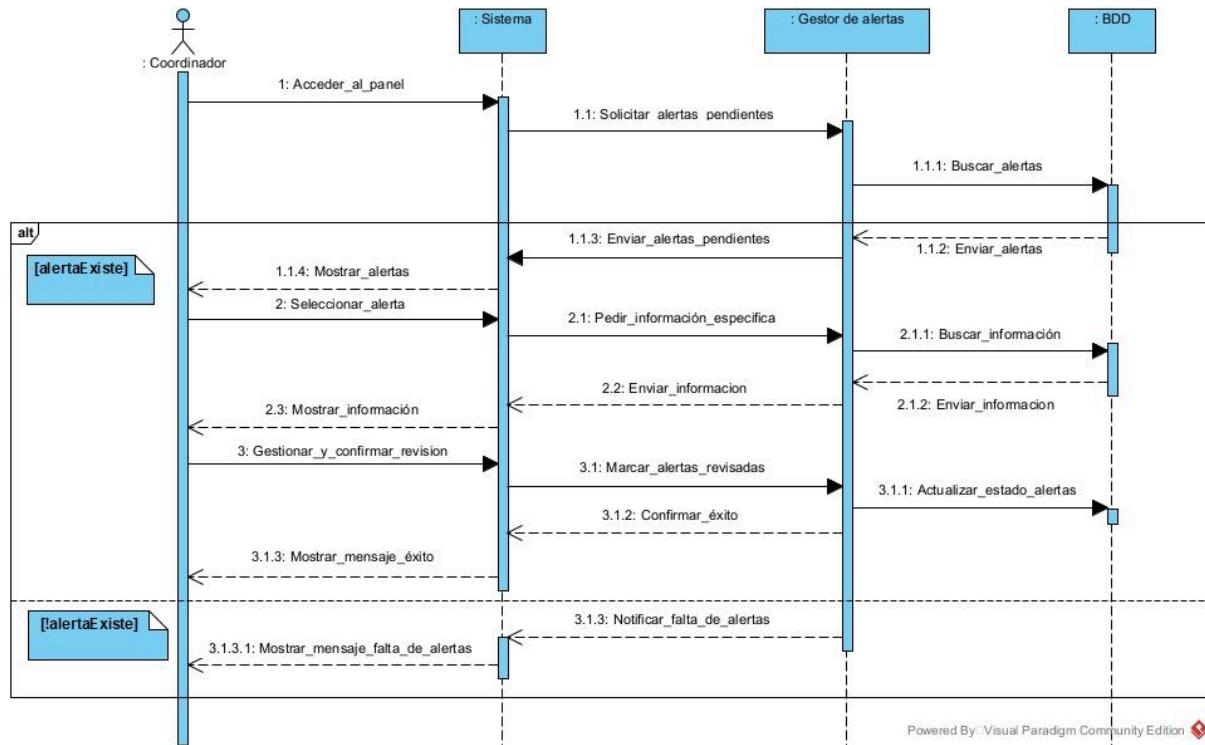
Clase: mensaje			
Representa una línea de texto con fecha y hora dentro de un chat.			
Atributos			
-	texto	string	Contenido del mensaje.
-	fechaHora	DateTime	Fecha y hora exactas del envío.
-	leido	Boolean	Indica si el destinatario ha leído el mensaje.

7.2. Matriz de Trazabilidad: Casos de uso VS Diagrama de Clases

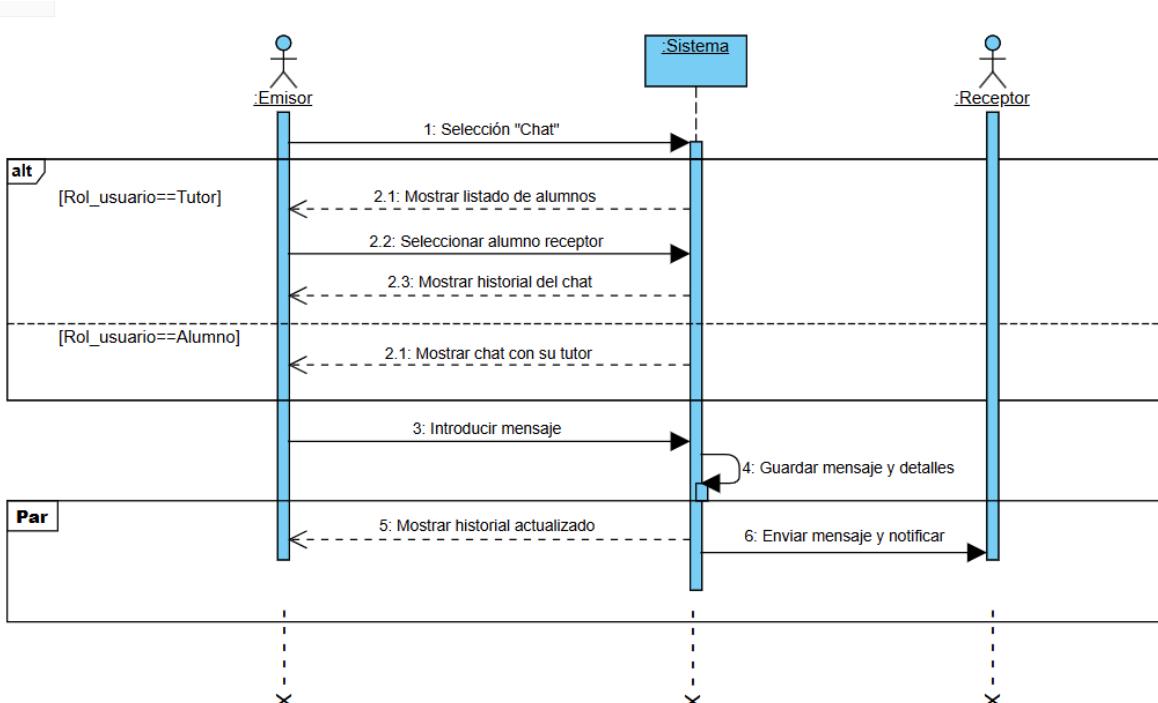
	CU1	CU2	CU3	CU4	CU5	CU6	CU7
Usuario							
Tutor							
Estudiante							
Coordinador							
Alerta							
Valoración							
Sesión							
Participante							
Conversación							
Mensaje							
Solicitud Cambio							

8. Diagramas de secuencia

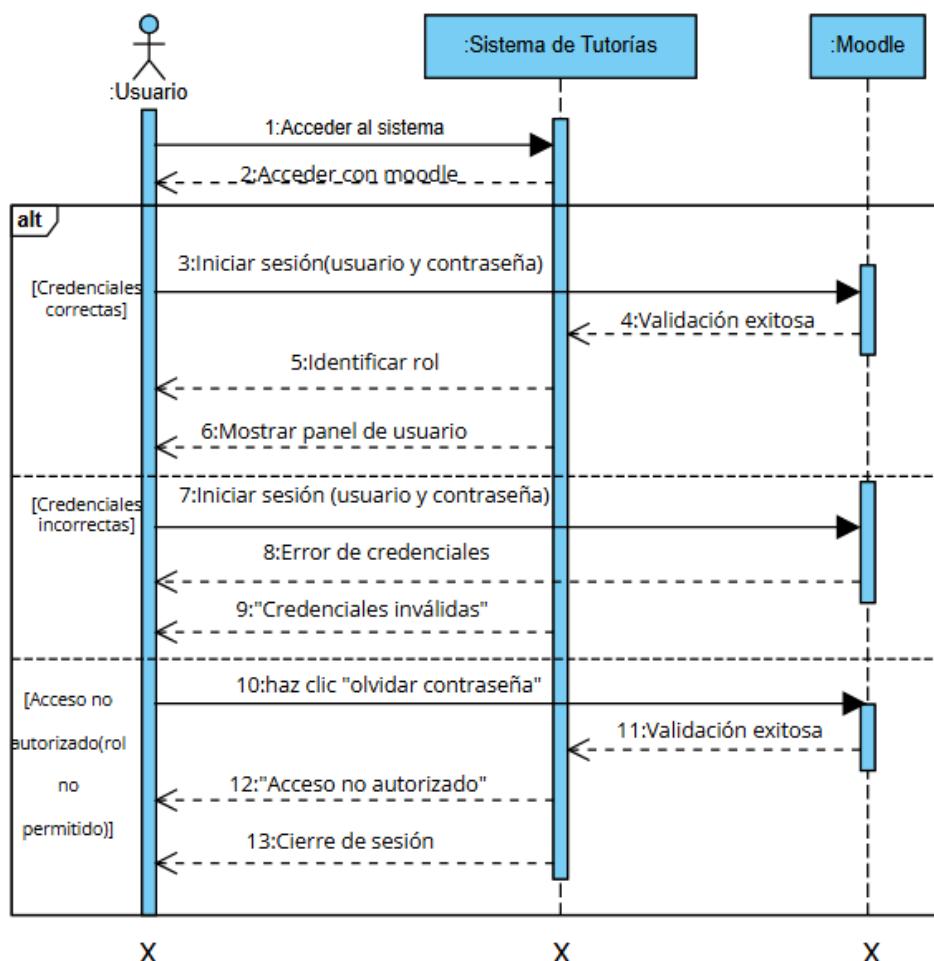
Gestión de alertas



Chat bidireccional



Acceso al sistema



Enlace a Youtrack

Se adjunta el link para acceder a la plataforma YouTrack: [aquí](#).

9. Bibliografía

Para la realización de la práctica hemos usado los archivos siguientes que aparecen en Moodle:

Práctica 1.1 - Organización y apuntes
Práctica 1.1 - Diapositivas
Práctica 1.2 - Organización y apuntes
Práctica 1.2 - Diapositivas
Especificación Requisitos IEEE 830
Métodos débiles de análisis
Scrum Historias de Usuario
Buenas prácticas de casos de uso
Ejemplo de caso de uso
Visual Paradigm: Diagrama de casos de uso
Práctica 2 - Organización y apuntes
Práctica 2 - Diapositivas

También hemos utilizado otras herramientas como la Inteligencia Artificial: gemini Pro. Para preguntar por posibles correcciones principalmente en Historias de Usuario y Casos de Uso, también como apoyo para utilizar Visual Paradigm.

Para el diagrama de casos de uso se ha utilizado Visual Paradigm.